EDM 2014 Extended Proceedings

7th International Conference on Educational Data Mining (EDM 2014)

London, UK - July 4-7, 2014

Edited by:

Sergio Gutiérrez-Santos
Birkbeck College, UK

Olga C. Santos
aDeNu Research Group, UNED, Spain

Published in CEUR workshop proceedings
ISSN 1613-0073

EDM 2014 Extended Proceedings

Preface

This volume compiles the papers accepted for publication in the four workshops that take place in the 7th International Conference on Educational Data Mining (EDM 2014). For the first time in this conference series a call for workshops was published, resulting in a very positive response from the EDM community, as these proceedings show.

The purpose of the EDM workshops is to provide an opportunity for participants from academia, industry, government, and other related parties to present and discuss novel ideas on current and emerging topics relevant to Educational Data Mining. EDM requires adapting existing approaches or developing new approaches that build upon techniques from a combination of areas, including but not limited to statistics, psychometrics, machine learning, information retrieval, recommender systems, and scientific computing.

The workshops held in conjunction with EDM 2014 are the following:

- **Graph-based Educational Data Mining (G-EDM).** Organizers: *Collin F. Lynch, Tiffany Barnes*
- **Non--Cognitive Factors & Personalization for Adaptive Learning (NCFPAL).** Organizers: *Steven Ritter, Stephen E. Fancsali*
- **Approaching Twenty Years of Knowledge Tracing: Lessons Learned, Open Challenges, and Promising Developments (BKT20y).** Organizers: *Michael Yudelson, José P. González-Brenes, Michael Mozer*
- **Feedback from Multimodal Interaction in Learning Management Systems (FFMI).** Organizers: *Lars Schmidt-Thieme, Ruth Janning*

We would like to thank all workshop organizers for their involvement and cooperation along the process, as well as their efforts in attracting contributions and participants to their workshops.

Sergio & Olga
June, 2014

Table of Contents

Workshop on Graph-Based Educational Data Mining (G-EDM)

Graph data has become increasingly prevalent in data-mining and data analysis generally. Many types of data can be represented naturally as graphs including social network data, log traversal, and online discussions. Moreover recent work on the importance of social relationships, peer tutoring, collaboration, and argumentation has highlighted the importance of relational information in education including:

- Graphical solution representations such as argument diagrams and concept maps;
- Graph-based models of problem-solving strategies;
- User-system interaction data in online courses and open-ended tutors;
- Sub-communities of learners, peer-tutors and project teams within larger courses; and
- Class assignments within a larger knowledge space.

Our goal in this workshop was to highlight the importance of graph data and its relevance to the wider EDM community. We also sought to foster the development of an interested community of inquiry to share common problems, tools, and techniques. We solicited papers from academic and industry professionals focusing on: common problems, analytical tools, and established research. We also particularly welcomed new researchers and students seeking collaboration and guidance on future directions. It is our hope that the papers published here will serve as a foundation for ongoing research in this area and as a basis for future discussions.

The papers included here cover a range of topics. Kovanovic, Joksimovic, Gasevic & Hatala focus on evaluating social networks, and specifically on the development of social capital and high-status individuals in a course context while Catete, Hicks, Barnes, & Lynch describe an online tool designed to promote social network formation in new students. Similar work is also described by Jiang, Fitzhugh & Warschauer who focus on the identification of high-connection users in MOOCs.

Other authors turned to the extraction of plan and hint information from course materials and user logs. Belacel, Durand, & Laplante define a graph-based algorithm for identifying the best path through a set of learning objects. Kumar describes an algorithm for the automatic construction of behavior graphs for example-tracing tutors based upon expert solutions and Dekel & Gal in turn consider plan identification to support automatic guidance. Two further papers by Vaculík, Nezvalová & Popelínský, and by Mostafavi & Barnes, apply graph analysis techniques to the specific domain of logic tutoring and, in particular, on the classification of student solutions and to the evaluation of problem quality.

And finally several authors chose to present general tools for the evaluation of graphical data. Lynch describes Augmented Graph Grammars, a formal rule representation for the analysis of rich graph data such as argument diagrams and interconnected student assignments, and details an implementation of it. Sheshadri, Lynch, & Barnes present InVis a visualization and analysis platform for student interaction data designed to support the types of research described above. And

McTavish describes a general technique to support graph analysis and visualization particularly for student materials through the use of interactive hierarchical edges. We thank the included authors for their contributions to the discussion and look forward to continued research.

The G-EDM workshop organizers
Collin F. Lynch
Tiffany M. Barnes

Table of Contents G-EDM

A Binary Integer Programming Model for Global Optimization of Learning Path Discovery

Nabil Belacel
National Research Council Canada
100, des Aboiteaux St.
Moncton, E1A 7R1,Canada
+1.506.861.0963
nabil.belacel@NRC.gc.ca

Guillaume Durand
National Research Council Canada
100, des Aboiteaux St.
Moncton, E1A 7R1,Canada
+1.506.861.0961
guillaume.durand@NRC.gc.ca

François Laplante
Université de Moncton
60, Notre-Dame-du-Sacré-Cœur St.
Moncton, E1A 3E9,Canada
+1.506.381.6220
francois.laplante@umoncton.ca

ABSTRACT

This paper introduces a method based on graph theory and operations research techniques to optimize learning path discovery. In this method, learning objects are considered as nodes and competencies as vertices of a learning graph. A first step consists in reducing the solution space by obtaining an induced subgraph H. In a second step, the search of an optimal learning path in H is considered as a binary integer programming problem which we propose to solve using an exact method based on the well-known branch-and-bound algorithm. The method detailed in the paper takes into account the prerequisite and gained competencies as constraints of the optimization problem by minimizing the total competencies needed to reach the learning objective.

Keywords

Learning path, learning object recommendation, graph theory, clique, mathematical programming, binary integer programming, branch-and-bound algorithm.

1. INTRODUCTION

Global Positioning System (GPS) is a Global Navigation Satellite System (GNSS) that is massively used by car drivers. This large acceptance is easily understandable by the benefits that such a system can offer. Car navigation systems can dynamically calculate an itinerary between two points taking into account, depending on the system, several constraints like duration, distance, closed roads, traffic jams, etc....Drivers can focus exclusively on their driving limiting risks of accidents, stress, and losing their way.

To some extent, the learning path followed by a student could be seen as an itinerary between several learning objects [9]. In this context, constraints on learning objects are not distance or time duration to go from one learning object to the other but rather prerequisite and gained competencies. As a result the itinerary or path between learning objects is regulated by competency dependencies that lead a learner from an initial to a targeted competency state. For example, a learner with solid grounds in integer arithmetic (starting location) willing to learn the solving of systems with multiple variables (destination) should be advised to previously learn to solve one variable linear equations (next step of the itinerary).

Over the years, educational data mining and recommendation technologies have proposed significant contributions to provide learners with adequate learning material by recommending educational papers [18] or internet links [10], using collaborative and/or content-based filtering. These approaches usually aim at recommending learning material satisfying an immediate interest rather than fitting in the learner's sequential learning process.

Sequential pattern [28] and process mining [19] technologies have also been investigated. However, these technologies have been used to understand the learner's interaction with content to discover general patterns and trends rather than to recommend adapted learning paths to learners.

Other approaches, in the course generation research community, address the need for recommending not only the learning objects themselves, but sequences of learning objects. Sicilia et al. [17] or Ulrich and Melis [20] addressed learning design concepts and requirements through Course Generation. Though numerous solutions have been proposed, using statistical methods [13], decision rules [23], production rules [11], Markov processes [8] and Hierarchical Task Network Planning [17, 21, 22], most of them do not take into account eventual competency dependencies among learning objects and/or are not designed for large repositories of interdependent learning objects[1].

Therefore, we detailed in [7] a dynamic graph based model and a heuristic approach tailored to find a learning path in a graph containing millions of learning object nodes.

This paper is an extension of this previous work and summarizes the model, the heuristic presented in [7], and proposes a major optimization to calculate a global optimum learning path. In the previous work [7], we applied a greedy heuristic algorithm to obtain a pseudo-optimal learning path from a set of cliques. Greedy heuristics are efficient, but they sometimes get stuck in a local solution and fail to find a global optimum [26]. They are based on an intimate knowledge of the problem structure and have no scope of incremental improvement.

[1] A more complete discussion can be found in [7].

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Therefore, in this work we slightly reformulate our model in order to fit as an integer programming problem and we propose an exact method based on the branch-and-bound algorithm.

2. PROBLEM CONSIDERED

In order to facilitate the understanding of the presented model, several key elements and assumptions need to be clearly defined.

A competency can be seen as a knowledge component being part of a "model that decomposes learning into individual knowledge components (KCs)" [16]. In this paper, a competency is "an observable or measurable ability of an actor to perform a necessary action(s) in given context(s) to achieve a specific outcome(s)" [12]. A competency in our situation can be a prerequisite to the efficient completion of a learning object. According to Wiley [25], a learning object is "any digital resource that can be reused to support learning". In the rest of the paper we define the learning object as any digital resource that can be reused to provide a competency gain.

A learner is a dynamic user interacting with learning objects in order to increase his/her competencies from an initial set to a targeted set of competencies. We assume that a learner completing a learning object will gain the competencies targeted to be transmitted by the interaction with the learning object. We also assume that a learner who would not possess the prerequisite set of competencies required by a learning object should not attempt this learning object since this would result in a limited competency gain.

Last but not least, we assume that the number of learning objects available is very large (millions to billions of learning objects) and that each learning object cannot provide the gain of a competency that is a pre-requisite to itself.

2.1 Graph Theory Contribution

Graph theory aims at studying mathematical structures composed of elements having relationships or connection between them. The use of directed graphs is not a novelty in e-learning systems [1, 3, 24, 25]; however, we were unable to find a formal model for discussing learning path problems based on graph theory, especially one taking into account the dynamic nature of a learning environment.

A directed graph, or digraph, $G = (V, E)$ consists of:

- A non-empty finite set V of elements called vertices or nodes,
- A finite set E of distinct ordered pairs of vertices called arcs, directed edges, or arrows.

Let $G = (V, E)$ be a directed graph for a personalized learning path. Each vertex or node in G corresponds to a learning object. Two vertices are connected if there exists a dependency relation, such that one vertex satisfies the prerequisites of the other. So, each edge between two vertices $Arc\{u, v\}$ means that the learning object v is accessible from u. The accessibility property required to define edges between vertices relies on post and pre-requisite competencies associated to each learning object. Considering $Arc\{u, v\}$, this edge means that after having completed the learning object u, the learner should have the required competencies to undertake resource v. By extension, each vertex v is represented by a pair (C_{pre}, C_{post}) where:

- C_{pre} is a set of the competencies required by vertex v
- C_{post} is a set of competencies offered by vertex v

The relationship between learning objects and competencies is multidimensional [6]: a learning object can require several competencies and transmit more than one competency to the learner as well. The existence of an edge between two learning objects u and v can be formalized by the following formula:

$$C_{pre}(v) \subseteq C_{post}(u) \Rightarrow Arc\{u, v\}$$

$$(Condition\ 1)$$

where $C_{pre}(v) \subseteq C_{post}(u)$ means that the competencies required by v are provided by learning object u. *Condition 1* is sufficient but not necessary. For example, before having completed u, the learner might already have some or the totality of the competencies required by v. This means that we may have an arc between u and v even though none the competencies required by v are provided by u. In other words, edge set E also depends on the learner's competency set at time t: $E = E(\mathrm{C}_{\mathrm{learner}}(t))$ and $\mathrm{C}_{\mathrm{learner}}(\mathrm{t}) = \{c_1, \ldots, c_n\}$ where $c_1 \ldots c_n$ are competencies which the learner possesses. As a result, graph G is a dynamic directed graph and *condition 1* can be strengthened by the necessary and sufficient *condition 2*:

$$Arc\{u, v\} \Leftrightarrow C_{pre}(v) \subseteq C_{post}(u) \cup C_{learner}(t)$$

$$(Condition\ 2)$$

2.2 Model Dynamicity

The dynamicity of our model is due to the fact that a learning object can bring competencies that could be among the prerequisites of future learning objects.

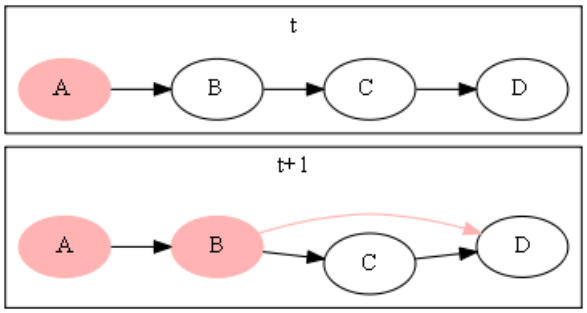


Figure 1. Edge dynamicity.

For example, as shown in Figure 1, a learning object D could be accessible to a learner if he has acquired the competencies c1 and c2. Assuming that competency c1 is provided by learning objects A and C and competency c2 is provided by learning objects B and C; D is reachable if learning objects A and B are completed or if learning object C is completed. If a learner completes learning object A at time t and learning object B at time t+1, the learner will have the competencies required to reach D and according to the condition 2, a new edge between B and D will be created (red edge on Figure 1).

3. INVESTIGATED SOLUTION

3.1 Reducing the solution space

Eliminating irrelevant learning objects is generally the first step of a course generation tool [1, 15]. In our case, as the learning object repository is supposed to be very large, the learning objects

cannot all be checked individually. The approach we chose consists in reducing the considered solution space by obtaining an induced subgraph H which consists of all the vertices and edges between the vertices in G that could be used in the learning path.

The algorithm can be seen as a loop generating complete sub-graphs, or cliques, until one such clique is generated whose prerequisites are a subset of the learner's competencies. Cliques are generated in a top-down fashion where we begin with the target clique, which is composed of a single learning object (we create a fictitious learning object, β, whose prerequisite competencies correspond to the list of the learner's target competencies). Cliques are then generated by finding every vertex where at least one output competency is found in the prerequisite competencies of the clique (the union of all prerequisite competencies of every learning object within the clique) to which it is prerequisite. As such, cliques contain the largest possible subset of vertices which satisfies the condition "if every learning object in the clique is completed, then every learning object in the following clique is accessible". We simplify the stopping condition by adding a second fictitious object, α, into the dataset with no prerequisite competencies and with the learner's current competencies as its output competencies. If a clique contains this object, the stopping condition is true.

	β_6	
v_1	$A^6_5 \quad E^6_{3,5}$	↑ **6**
v_2	$T^{3,2,4}_7 \; U^5_0$	↑ **3,5**
v_3	$L^{0,7}_{8,9} \; I^7_9 \; K^0_8$	↑ **0, 7**
	$A^{8,9}$	↑ **8, 9**

α: Fictitious LO with initial learner competency state

β: Fictitious LO with targeted learner competency state

$LO^{\text{list of gained competencies}}\ LO_{\text{list of prerequisite competencies}}$

Figure 2. Induced sub-graph generation.

Considering the target competency β as shown in Figure 2, all the vertices leading to those competencies (competency 6 in Figure 2) are selected in a set v_1, then the learning objects leading to the prerequisites of set v_1 (competencies 3 and 5) are selected from graph G to create the set v_2. This mechanism continues until the prerequisite competencies of the set v_n are all competencies which the learner has already acquired.

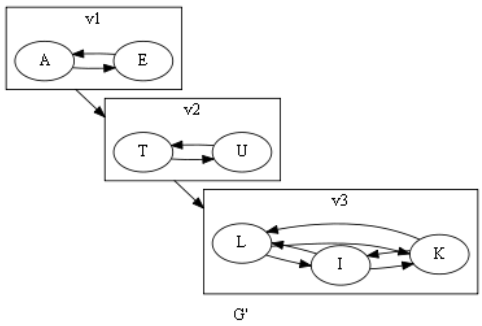


Figure 3. G' consists of connected cliques.

As shown in Figure 3, G', consisting of the vertices E of sets $v_1,\ldots,v_n$, is an induced sub-graph of G. If the learner has completed all the vertices of v_i, he/she will have access to all the vertices of v_{i+1}, thus all subsets of vertices of v_i can be considered to be a clique.

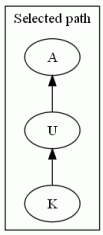
In addition to reducing the solution space, clique generation is also an efficient way to check whether a solution learning path exists between α and β. If the algorithm is not able to generate cliques linking α and β, there is no need to proceed forward with an algorithm aiming at finding one of the possible solutions.

3.2 Greedy Algorithm

Once the induced sub-graph is obtained, we use a greedy algorithm that searches for a local optimum within each clique. The definition of such a local optimum, depending on the dataset and the results pursued, has to follow a specific heuristic or strategy.

The shortest path strategy seems to be widely accepted in the literature [1, 27]. This strategy is not necessarily the best to adopt in any situation since the proposed learning path might lead to the learning of non–essential competencies and potentially cognitive overloads. For example a learning object could lead to competency gains that would not be required to reach the targeted learner competency state; there is no need to understand the proof of the Landau Damping to learn about the history of theoretical physics. Considering a learning object presenting an introduction to the perturbation theory and a second one introducing the theory and the proof of the Landau Dumping, it might make sense to choose the first one in order to minimize the cognitive load to the learner. Some might argue that using such "straight to the point" heuristic might limit too drastically the natural curiosity of the learner. As any heuristic, we agree that it is discussable but this is not the purpose of this paper.

The greedy algorithm considered attempts to find a path by considering each clique one after the other and reducing it to a minimal subset of itself which still verifies the condition "if every learning object in the clique is completed, then every learning object in the following clique is accessible".



	β_6	
v_1	$\boldsymbol{A^6_5}$ $E^6_{3,5}$	$\uparrow$ **6**
v_2	$T^{3,2,4}_7$ $\boldsymbol{U^5_0}$	$\uparrow$ **3,5**
v_3	$L^{0,7}_{8,9}$ I^7_9 $\boldsymbol{K^0_8}$	$\uparrow$ **0,7**
	$A^{8,9}$	$\uparrow$ **8, 9**

α: Fictitious LO with initial learner competency state

β: Fictitious LO with targeted learner competency state

$LO^{\text{list of gained competencies}}\ LO_{\text{list of prerequisite competencies}}$

Figure 4. Illustration of the greedy algorithm execution

The first clique considered will be the one leading to the targeted competencies (the clique satisfying the prerequisites of β). In the case of the three cliques v_1 to v_3 as illustrated by Figure 3, v_1 will be considered first followed by v_2 then by v_3.

For each clique, the local optimum is considered obtained when the minimum subset of vertices with a minimum "degree", being the sum of the number of prerequisite competencies and output competencies of the vertex, are found. In other words, the greedy algorithm select in each clique a set of learning objects minimizing the number of competencies required and gained in order to locally limit the cognitive load of the selected material. The greedy algorithm locally optimizes a function called "deg" (for degree) detailed in the following section.

For clique v_1, the selected learning object is A since its number of prerequisites is smaller than that of E while they share the same competency gain. As A has been chosen in v_1, only the objects in v_2 respecting the new v_1's prerequisites is chosen. As a result, the algorithm chooses U in v_2. In v_3, K and L lead to v_2's prerequisite but K requires fewer prerequisites than L, therefore K is selected and the proposed learning path is $K \rightarrow U \rightarrow A$.

4. OPTIMIZATION

In this section we present our mathematical model for learning path discovery and then we introduce the algorithm for solving our mathematical model.

After eliminating irrelevant learning objects in the first step, we generate the optimal solution from the obtained induced sub-graph as presented in Figure 4. For this purpose, we applied in [7] a greedy algorithm to obtain an optimal or pseudo-optimal learning path from a set of cliques. Greedy heuristics are computationally efficient, but they sometimes fail to find a global optimum as we explain in the following section.

4.1 Notation and limits of the Greedy heuristic

Let $Q_{n,m}$, $G_{n,m}$, $C_{n,v}$ the matrices representing the distribution of the m competencies that are prerequisite to the n items contained in the v cliques, the m competencies that are gained when the n items of the v cliques are performed, and the clique distribution of the n items. Note that the matrix $Q_{n,m}$ could be considered as a Q-Matrix [5].

Considering our example (*Example 1*):

- $N = \{A, E, T, U, L, I, K\}$,
- $M = \{0,2,3,4,5,6,7,8,9\}$,
- $V = \{v_1, v_2, v_3\}$.

$$Q_{n=7,m=9} = \begin{pmatrix} & 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ T & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ U & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ K & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \end{pmatrix}$$

$$G_{n=7,m=9} = \begin{pmatrix} & 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ T & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ U & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ L & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ K & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C_{n=7,v=3} = \begin{pmatrix} & \mathrm{v}_1 & \mathrm{v}_2 & \mathrm{v}_3 \\ \mathrm{A} & \mathbf{1} & 0 & 0 \\ \mathrm{E} & 1 & 0 & 0 \\ \mathrm{T} & 0 & 1 & 0 \\ \mathrm{U} & 0 & \mathbf{1} & 0 \\ \mathrm{L} & 0 & 0 & 1 \\ \mathrm{I} & 0 & 0 & 1 \\ \mathrm{K} & 0 & 0 & \mathbf{1} \end{pmatrix}$$

From this example the solution sequence using the greedy algorithm is $K \rightarrow U \rightarrow A$.

To check if we get an optimal solution or not, we have to calculate the objective function called deg. The objective function deg returns the total number of prerequisite and gained competencies of a set of learning objects.

We can draw from the previous example the following conditions to check if we have an optimal solution or not.

Let $S = \{s_0, s_1, \dots, s_v, s_{v+1}\}$ a solution set (s_i contains at least one learning object as in example 3).

$$\forall s_{i=1..v} \in S, \quad s_0 = \alpha,\ s_{v+1} = \beta,\ Q_{s_i} \subseteq G_{s_{i-1}} \quad (i)$$

$$\forall\, j = 1 \dots v \neq i = 1 \dots v,\ C_{s_i} \cap C_{s_j} = \emptyset \quad (ii)$$

$$\deg(S = \{s_0, s_1, \dots, s_v, s_{v+1}\}) = \sum_{i=0}^{v+1} \sum_{j=1}^{m} (Q_{s_i,j} + G_{s_i,j}) \quad (iii)$$

$$\forall s_{i=1..v+1} \in S;\ \exists\, s^*_{i=1..v+1} \in S^*$$

$$\deg(S^* = \{s^*_0, s^*_1, \dots, s^*_v, s^*_{v+1}\}) \leq \deg(S = \{s_0, s_1, \dots, s_v, s_{v+1}\})$$

$$(iv)$$

Condition (i) and (ii) mean that the competencies required by a clique set have to be covered by the gains of the previous clique set and two different clique sets cannot share the same clique. While condition (iii) defines the deg function, condition (iv) introduces the optimality condition. A learning path is optimal if

no other path with a lower degree exists. However this doesn't apply at the clique level since the optimal s_i^* is not necessary the set of clique i having the lowest degree. The global optimum is not the sum of the local optima calculated by the greedy algorithm.

The following example highlights this case where local optima obtained by the greedy algorithm lead to non-optimal solution.

Example 2:

	β_6	
v_1	$M^6{}_5$ $N^{6,7}{}_4$	↑ **6**
v_2	$O^5{}_{3,4}$ $P^4{}_8$	↑ **4,5**
v_3	$T^8{}_7$ $R^{3.4}{}_7$	↑ **3, 4, 8**
	α^7	↑ *7*

$$\deg(\alpha, R, O, M, \beta) = 1 + 3 + 3 + 2 + 1 = 10$$
$$\deg(\alpha, Q, P, N, \beta) = 1 + 2 + 2 + 3 + 1 = 9$$

The solution obtained by the greedy algorithm is $S_a = \alpha \to R \to O \to M \to \beta$ and the associated value of the objective function deg (S_a) is equal to 10. As the algorithm starts from β, it chooses in each clique the learning object with the lowest degree which is M and keeps going until it reaches α.

The path $S_b = \alpha \to T \to P \to N \to \beta$ is an alternative that the algorithm did not find. It's even a better alternative since $\deg(S_b) = 9 \leq \deg(S_a) = 10$ and the optimal solution.

The following example highlights another case where local optima obtained by the greedy algorithm lead to a non-optimum solution. In this example, two learning objects are selected in one of the generated cliques.

Example 3:

	β_6	
v_1	$M^6{}_5$ $N^{6,7}{}_4$	↑ **6**
v_2	$O^5{}_{3,9}$ $P^4{}_8$	↑ **4,5**
v_3	$T^8{}_7$ $Y^9{}_7$, $Z^3{}_7$	↑ **3, 9, 8**
	α^7	↑ *7*

$$\deg(\alpha, Y, Z, O, M, \beta) = 1 + 2 + 2 + 3 + 2 + 1 = 11$$

The objective function of the path $(\alpha \to T \to P \to N \to \beta)$ is 9, which means that the path $(\alpha \to T \to P \to N \to \beta)$ is the optimal solution.

In the following section, we use the notation introduced here to propose a mathematical formulation of our learning path optimization problem as an integer programming problem.

4.2 Formulating the integer programming problem

Let us consider n items or learning objects and m competencies; $Q_{n,m}$ is the matrix representing m prerequisite competencies for the n items and $G_{n,m}$ is the matrix representing the m competencies that are gained when the n items are performed. In other words, if $Q_{ij} = 1$ means that the item i has competency j as one of its prerequisite competencies; and $G_{i,j} = 1$, means that the competency j is gained when the item i is performed. The personalized learning path may then be formulated as a binary integer programming (BIP) as follows:

Minimize:

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{m}\left(Q_{i,j} + G_{i,j}\right)x_i\right) = \deg(X) \qquad (1)$$

Subject to:

$$Q_{i,j}x_i - \left(\sum_{k=1}^{i-1} G_{k,j}x_k\right) \times Q_{i,j} \leq 0 \qquad (2)$$

$$for \quad i = 2, \dots, n-1; \quad for \quad j = 1, \dots, m; \quad x_i \in \{0,1\};$$

X = $\{x_i, i=1,\dots,n\}$, are the decision variables such that:

$$x_i = \begin{cases} 1 & if\ the\ item\ i\ is\ selected; \\ 0 & otherwise, \end{cases} \qquad (3)$$

We suppose that $x_1 = 1$ and $x_n = 1$, knowing that:

$$x_1 = 1\ presenting\ the\ initial\ item\ \alpha$$
$$and\ x_n = 1\ presenting\ the\ resulting\ item\ \beta$$

The function (1) represents the total number of prerequisite and gained competencies to be minimized. The constraints (2) states that if the item i has competency j as one of its prerequisite competencies; the competency j should be gained from the items on the learning path (1,…, i-1). Our problem is to minimize the objective function (1) subject to (2) and (3).

To find the optimal learning path we have to solve the BIP problem with $(n+m)$ constraints and n decision variables $x_{i=1,\dots n} \in \{0,1\}$.

Considering example 3, the prerequisite and gain matrices Q and G can be written as follows:

The competencies that are required by the items are represented by the matrix Q (9x7).

$$Q = \begin{pmatrix} \frac{Comp.}{LO} & C1 \equiv 7 & C2 \equiv 8 & C3 \equiv 9 & C4 \equiv 3 & C5 \equiv 5 & C6 \equiv 4 & C7 \equiv 6 \\ \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ Y & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ Z & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ O & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ P & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ M & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ N & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \beta & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The competencies that are gained by the items are represented by the matrix G (9x7).

$$G = \begin{pmatrix} \frac{Comp.}{LO} & C1 \equiv 7 & C2 \equiv 8 & C3 \equiv 9 & C4 \equiv 3 & C5 \equiv 5 & C6 \equiv 4 & C7 \equiv 6 \\ \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ T & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ Y & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ Z & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ O & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ P & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ M & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ N & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The BIP formulation of example 3 is given as follows:

Minimize :

$$\deg(X) = x_1+2x_2+2x_3+2x_4+3x_5+2x_6+2x_7+3x_8+x_9$$

Subject to:

$$x_2 - x_1 \leq 0$$
$$x_3 - x_1 \leq 0$$
$$x_4 - x_1 \leq 0$$
$$x_5 - x_3 \leq 0$$
$$x_5 - x_4 \leq 0$$
$$x_6 - x_2 \leq 0$$
$$x_7 - x_5 \leq 0$$
$$x_8 - x_6 \leq 0$$
$$x_9 - x_7 - x_8 \leq 0$$
$$x_i \in \{0,1\}, i = 1, \dots, 9$$

x_1 *is* the fictitious learning object α with initial learner competency state.

x_9 is the fictitious learning object β with targeted learner competency state.

Since $x_1 = x_9 = 1$, then our BIP becomes:

Minimize :

$$\deg(X) = 2x_2+2x_3+2x_4+3x_5+2x_6+2x_7+3x_8$$

Subject to:

$$x_5 - x_3 \leq 0$$
$$x_5 - x_4 \leq 0$$
$$x_6 - x_2 \leq 0$$
$$x_7 - x_5 \leq 0$$
$$x_8 - x_6 \leq 0$$
$$-x_7 - x_8 \leq -1$$
$$x_i \in \{0,1\}, i = 2, \dots, 8$$

4.3 The Branch-and-Bound (B&B) method for solving the BIP problem

Since the BIP problem is bounded, it has only a finite number of feasible solutions. It is then natural to consider using an enumeration procedure to find an optimal solution. However, in the case of large learning object repositories (millions of items), an enumeration procedure might be ill adapted (even after reducing the solution space); therefore, it is imperative to cleverly structure the enumeration procedure so that only a tiny fraction of feasible solutions need to be explored.

A well-known approach called branch-and-bound technique (B&B) provides such a procedure. B&B traces back to the 1960s' and the work of Land and Doig [14]. Since then, B&B algorithms have been applied with success to a variety of operations research problems. B&B is a divide and conquer method. It divides a large problem into a few smaller ones (This is the "Branch" part). The conquering part estimates the goodness of the solution that is obtained from each of the sub-problems; the problem is divided until solvable sub-problems are obtained (this is the "bound" part).

For the bounding part we use a linear programming relaxation to estimate the optimal solution [26]. For an integer programming model P; the linear programming model obtained by dropping the requirement that "*all variables must be integers*" is called the linear programming relaxation of P.

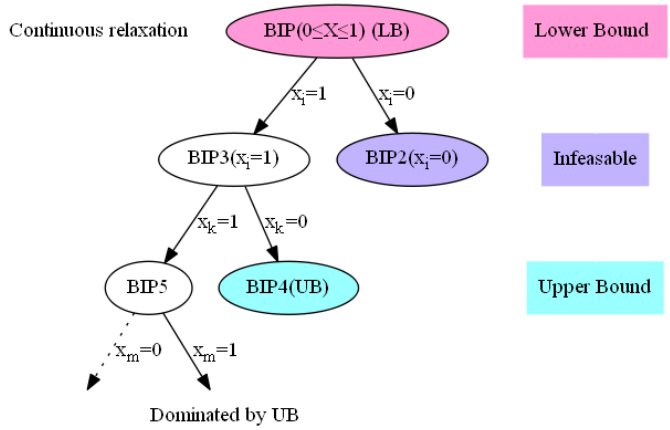


Figure 5. Branch and bound algorithm that traverses the tree by solving BIPs at every node of the tree.

The general approach of a BIP B&B algorithm [26] is presented in the following steps (see also Figure 5):

Initialization: Set deg* = $+\infty$.

The initial step represents the root node of the B&B search tree. The root node corresponds to the continuous relaxation of the BIP($0 \leq X \leq 1$), the solution value provides lower bound.

Apply the bounding step, fathoming step, and optimality test described below. If not fathomed, classify this problem as the one remaining "subproblems" for performing the first full iteration below.

Steps for each iteration:

1. Branching: Among the remaining (unfathomed) subproblems, select the one that was created most recently (break ties by selecting the subproblem with the larger bound). Branch from the node for this subproblem to create two new subproblems by fixing the next variable (the branching variable) at either 0 or 1 (see Figure 5).
2. Bounding For each new subproblem, obtain its bound by applying the simplex method to its LP-relaxation and rounding down the value of deg for the resulting optimal solution.
3. Fathoming (Pruning rules): The pruning rules for B&B BIP are based on optimality and feasibility of BIP. For

each new sub-problem, apply the fathoming tests and discard those sub-problems that are fathomed by any of the tests.

Optimality test: Stop when there are no remaining sub-problems:

- The current incumbent is optimal,
- Otherwise, return to perform another iteration.

A sub-problem is fathomed (dismissed from further consideration) if it verifies one of the following tests:

1. The relaxation of the sub-problem has an optimal solution with deg < deg∗ where deg* is the current best solution (The solution is dominated by upper bound);
2. The relaxation of the sub-problem (LP-relaxation) has no feasible solution;
3. The relaxation of the sub-problem has an optimal solution that has all binary values. (If this solution is better than the incumbent, it becomes the new incumbent, and test1 is reapplied to all unfathomed sub-problems with the new larger deg*).

For example, the *example 3* solved by B&B produces an optimal solution with deg* = 9 and x_2=1, x_6=1, x_8=1 where the number of nodes explored is 1 because the first LP-relaxation at node 1 gives an integer optimal solution with deg*=9 and the 3rd fathomed test is true, so we do not need to branch anymore.

Decision Variables	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
LO	α	T	Y	Z	O	P	M	N	β
X*	1	1	0	0	0	1	0	1	1

Figure 6. Solution of example 3in the B&B algorithm.

As illustrated in Figure 6, the optimal solution of the B&B algorithm is X*={1, 1, 0, 0, 0, 1, 0, 1, 1} and the optimal path is: $\alpha \rightarrow T \rightarrow P \rightarrow N \rightarrow \beta$.

5. CONCLUSION

The clique based approach is an asset since it offers an efficient way to reduce the solution space and check the existence of a solution. However, a greedy search within the cliques to find a leaning path does not lead, in many cases, to the best learning path according to the criteria considered.

Binary integer programming is a well-known mathematical optimization approach. While reformulating the conditions an optimal learning path should meet, we realised how we could benefit from expressing the constraints as a binary programming problem.

Our preliminary implementation of the proposed optimization using the *bintprog* function (*MATLAB*), a function based on the branch- and-bound (B&B) algorithm, shows the accuracy of the proposed integer program model.

In future work, we will apply the proposed binary integer model in order to build a learning design recommendation system in the case where learning objects are stored in very large repositories. Even though the B&B algorithm is highly accurate and somehow computationally efficient, it is not efficient enough to deal with very large size problem instances. In some cases, the bounding step of B&B is not invoked, and the branch and bound algorithm can then generate a huge number of sub-problems.

Moreover, as mentioned in [7], the efficiency of reducing the solution space with the cliques' mechanism is highly dependent on the dataset topology (average number of gain and prerequisite competencies per learning object). The solution space may remain large after the reduction

Therefore, to deal with very large problems, we will implement a variant of the B&B algorithm such as Branch & Cut [2] or Branch & Price [4]. Applegate et al. [2] showed how Branch & Cut could get a global optimal for extremely large binary optimization problems. It will be then interesting to measure both in terms of computational time and accuracy how the greedy search compares to the B&B-like approach.

6. ACKNOWLEDGMENTS

This work is part of the National Research Council of Canada's Learning and Performance Support Systems (NRC LPSS) program. The LPSS program addresses training, development and performance support in all industry sectors, including education, oil and gas, policing, military and medical devices.

7. REFERENCES

[1] Alian, M. Jabri, R. 2009. A shortest adaptive learning path in e-learning systems: Mathematical view, *Journal of American Science* 5(6) (2009) 32-42.

[2] Applegate, D., Bixby, R., Chvatal, V. and Cook, W. 1998. On The solution of traveling salesman problems, *in: Proc. Int. Congress of Mathematicians, Doc. Math. J. DMV*, Vol. 645.

[3] Atif, Y., Benlarmi, R., and Berri, J. 2003. Learning Objects Based Framework for Self-Adaptive Learning, *Education and Information Technologies, IFIP Journal, Kluwer Academic Publishers* 8(4) (2003) 345-368.

[4] Bamhart, C, Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and Vance, P. H. 1998. Branch-and-price: column generation for huge integer programs, *Operations Research* 46:316.

[5] Barnes, T. 2005. The Q-matrix Method: Mining Student Response Data for Knowledge. *Proceedings of the Workshop on Educational Data Mining at the Annual Meeting of the American Association for Artificial Intelligence*.

[6] Carchiolo, V., Longheu, A., and Malgeri, M. 2010. Reliable peers and useful resources: Searching for the best personalised learning path in a trust- and recommendation-aware environment, *Information Sciences* 180(10) (2010) 1893-1907.

[7] Durand, G., Belacel, N., and Laplante, F. 2013. Graph theory based model for learning path recommendation. *Information Sciences.* 251(10) (2013) 10-21.

[8] Durand, G., Laplante, F. and Kop, R. 2011. A learning Design Recommendation System Based On Markov Decision Processes, Proc. *17th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD) Workshop on Knowledge Discovery in Educational Data*, San Diego, CA.

[9] Durand, G., Downes, S. 2009. Toward Simple Learning Design 2.0. *In: 4th Int. Conf. on Computer Science &Education 2009*, Nanning, China, 894-897.

[10] Godoy, D., Amandi, A. 2010. Link Recommendation in E-learning Systems based on Content-based Student Profiles, In: Romero C., Ventura S., Pechenizkiy, M., Baker, R. (Eds.), *Handbook of Educational Data Mining, Data Mining*

and Knowledge Discovery Series, Chapman & Hall/CRC Press, 273-286.

[11] Huang, Y.M., Chen, J.N., Huang, T.C., Jeng, Y.L., and Kuo, Y.H. 2008. Standardized course generation process using Dynamic Fuzzy Petri Nets, *Expert Systems with Applications*, 34 (2008) 72-86.

[12] ISO 24763/final version: Conceptual Reference Model for Competencies and Related Objects, 2011.

[13] Karampiperis, P., Sampson, D. 2005.Adaptive learning resources sequencing in educational hypermedia systems. *Educational Technology & Society* 8 (4) (2005) 128-147.

[14] Land, A. H., Doig, A. G. 1960. An automatic method of solving discrete programming problems. *Econometrica* 28(3), 497–520.

[15] Liu, J., Greer J. 2004. Individualized Selection of Learning Object, *In: Workshop on Applications of Semantic Web Technologies for e-Learning*, Maceió, Brazil.

[16] Pavlik, P. I. Jr., Presson, N., and Koedinger K. R. 2007. Optimizing knowledge component learning using a dynamic structural model of practice, *Proc. 8th International Conference on Cognitive Modeling.* Ann Arbor, MI.

[17] Sicilia, M.-A., Sánchez-Alonso, S. and García-Barriocanal, E. 2006. On supporting the process of learning design through planners, Proc. *Virtual Campus Post-Selected and Extended*, 81–89.

[18] Tang, T.Y., Mccalla, G.G. 2010. Data Mining for Contextual Educational Recommendation and Evaluation Strategies, In: Romero C., Ventura S., Pechenizkiy, M., Baker, R. (Eds.), *Handbook of Educational Data Mining, Data Mining and Knowledge Discovery Series*, Chapman & Hall/CRC Press, Chapter 18,257-271.

[19] Trcka, N., Pechenizkiy, M. and Van-Deraalst, W. 2010. Process Mining from Educational Data, In: Romero C., Ventura S., Pechenizkiy, M., Baker, R. (Eds.), *Handbook of Educational Data Mining, Data Mining and Knowledge Discovery Series*, Chapman & Hall/CRC Press, Chapter 9, 123-141.

[20] Ullrich, C., Melis, E. 2010. Complex Course Generation Adapted to Pedagogical Scenarios and its Evaluation, *Educational Technology & Society*, 13 (2) (2010) 102–115.

[21] Ullrich, C., Melis, E. 2009. Pedagogically founded courseware generation based on HTN-planning, *Expert Systems with Applications* 36(5) (2009) 9319-9332.

[22] Ullrich C. 2005. Course Generation Based on HTN Planning, Proc. *13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, Saarbrucken, Germany,74-79.

[23] Vassileva, J., Deters, R. 1998, Dynamic courseware generation on the www, *British Journal of Educational Technology*, 29(1) (1998) 5–14.

[24] Viet, A., Si, D.H. 2006. ACGs: Adaptive Course Generation System - An efficient approach to Build E-learning, *Proc. 6th IEEE International Conference on Computer and Information Technology*, Jeju Island, Korea, 259-265.

[25] Wiley, D.A. 2002. Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy, *In: The Instructional Use of Learning Objects, D. A. WILEY (Ed.)*, 3-23.

[26] Winston, W.L., Venkataramanan, M. 2003. *Operations Research: Introduction to Mathematical Programming*. Thompson, 4th Edition.

[27] Zhao, C., Wan, L. 2006. A Shortest Learning Path Selection Algorithm in E-learning, *Proc. 6th IEEE International Conference on Advanced Learning Technologies*, Kerkrade, The Netherlands, 94-95.

[28] Zhou, M., Xu, Y., Nesbit, J.C. and Winne, P.H. 2010. Sequential pattern analysis of learning logs: Methodology and applications, In: Romero C., Ventura S., Pechenizkiy, M., Baker, R. (Eds.), *Handbook of Educational Data Mining, Data Mining and Knowledge Discovery Series*, Chapman & Hall/CRC Press, Chapter 8, 107-120.

On-Line Plan Recognition in Exploratory Learning Environments

Reuth Dekel and Ya'akov (Kobi) Gal
Dept. of Information Systems Engineering
Ben-Gurion University
Beer-Sheva 84105, Israel

ABSTRACT

Exploratory Learning Environments (ELE) are open-ended and flexible software, supporting interaction styles by students that include exogenous actions and trial-and-error. ELEs provide a rich educational environment for students and are becoming increasingly prevalent in schools and colleges, but challenge conventional plan recognition algorithms for inferring students' activities with the software. This paper presents a new algorithm for recognizing students' activities in ELEs that works on-line during the student's interaction with the software. Our approach, called CRADLE, reduces the amount of explanations that is maintained by the plan recognition in a way that is informed by how people execute plans. We provide an extensive empirical analysis of our approach using an ELE for chemistry education that is used in hundreds of colleges worldwide. Our empirical results show that CRADLE was able to output plans exponentially more quickly than the state-of-the-art without compromising correctness. This result was confirmed in a user study that included a domain expert who preferred the plans outputted by CRADLE to those outputted by the state-of-the-art approach for the majority of the logs presented.

1. INTRODUCTION

This paper focuses on inferring students' activities in educational environments in which students engage widely in exploratory behavior, and present new approaches for plan recognition in such settings that can outperform the state-of-the-art.

Our empirical analysis is based on students' interactions with an Exploratory Learning Environment (ELE) in which students build scientific models and examine properties of the models by running them and analyzing the results[1, 6]. Such software is open-ended and flexible and is generally used in classes too large for teachers to monitor all students and provide assistance when needed. The open-ended nature of ELEs affords a rich spectrum of interaction for students: they can solve problems in many different ways, engage in exploratory activities involving trial-and-error, they can repeat activities indefinitely, and they can interleave between activities.

These aspects significantly hinder the possibilities of making sense of students' activities without some sort of support. This paper presents a new algorithm for recognizing students' interactions with ELEs in real time, which can support both teachers and students. For teachers, this support takes the form of visualizing students' activities during their interaction in a way that faciliteates their understanding of students' learning. For students, this support can take the form of machine generated intervention that guides their learning and adapts to individual students' needs based on their inferred behavior.

The focus of this paper is on-line recognition that occurs during the students' actual interaction with the ELE, and outputs a hierarchy of interdependent activities that best describe the student's work at a given point in time. Recognizing students' activities this way is challenging because the algorithm needs to reason about and maintain possible explanations for future (yet unseen) student activities. The number of possible explanations grows exponentially with the number of observations. As we show in the empirical section of this paper, this significantly hinders the performance of the state-of-the-art, even for very short interaction sequences.

Our algorithm, called CRADLE (Cumulative Recognition of Activities and Decreasing Load of Explanations) builds on an existing approach for on-line plan recognition, but filters the space of possible explanations in a way that reflects the style of students' interactions in ELEs. The filtering aim is to produce complete, parsimonious and coherent explanations of students' interactions that can be easily understood by teachers and education researchers.

Our empirical evaluations were based on comparing CRADLE to the state-of-the-art approach for recognizing logs of students' interactions with a widely used ELE for chemistry education. We evaluated both of the approahes in terms of computation speed and correctness of the outputted explanation, as determined by a domain expert. Succeeding in both of these measures is criticial for an on-line plan recognition approach to work successfully.

Our empirical results show that CRADLE was able to outperform the state-of-the-art without compromising correctness. Specifically, although the state of the art approach is (in theory) complete, it was not able to terminate within an allocated time frame on many logs. In contrast, CRADLE was able to produce correct explanations for such logs. In addition, CRADLE significantly outperformed the state-of-the-art both in terms of correctness and speed of recognition.

These results demonstrate the benefit of applying novel plan recog-

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

nition technologies towards intelligent analysis of students' interactions in open-ended and flexible software. Such technologies can potentially support teachers in their understanding of student behavior as well as students in their problem solving, and lead to advances in automatic recognition in other exploratory domains.

2. RELATED WORK

Our work relates to two strands of research, inferring students' activities in educational software, and on-line planning algorithms in artificial intelligence. We relate to each of these in turn.

2.1 Inferring Students' Activities in ELEs and ITS systems

We first describe works that infer students' plans from their interactions with pedagogical software that assume the complete interaction sequence is known in advance. Gal et al. [11] and Reddy et al. [10] used plan recognition to infer students' plans from their interactions with TinkerPlots, an exploratory learning environment for statistics. Both of these approaches take as input a complete interaction sequence of a student as well as recipes for ideal solutions to TinkerPlots problems, and infer the plan used by the student retrospectively. Reddy et al. [10] proposed a complete algorithm which modeled the plan recognition task as a Constraint Satisfaction Problem (CSP). The complexity of the CSP algorithm is exponential in the size of both the interaction sequence and the data set containing the recipes. This approach requires that all possible plans can be explicitly represented, and therefore does not support recursive grammars which are needed to understand students' activities in VirtualLabs.

Other works have implemented plan recognition techniques to model students' activities in Intelligent Tutoring Systems (ITS) during their interactions. In contrast to exploratory learning environments, in intelligent tutoring systems the system takes an active role in students' interactions, as it tutors the student by providing feedback and hints. As an example, in the Andes physics tutor wrong steps are marked by the tutor and the students may ask for a "what's wrong here?" hint from the tutor. In addition, students can ask for a "what next?" hint to receive instruction when uncertain about how to proceed [20]. These systems are typically more closed-ended and less exploratory than ELEs. In the Andes physics tutor a probabilistic algorithm was used to infer the solutions plan followed by the student. For each Andes problem, a solution graph representing the possible correct solutions to the problem was automatically generated and were modeled using a dynamic Bayesian network. The algorithm observes students' actions and updates the probabilities of the different possible plans. The inferred plans were used to generate hints and to update students' cognitive models.

The tutors developed by the ACT-R group for teaching LISP, geometry and algebra, performed plan recognition using a model-tracing algorithm that tracked students' solution plans [2, 9]. These tutors maintained a list of production rules that can be triggered to accomplish the goal and sub-goals for solving a problem. The algorithm infers students' plans by identifying the production rules that were triggered according to the actions students had taken. After each observed action, the algorithm commits to a production rule that it infers the student triggered to perform the action. The system constrained students to remain on "correct paths" throughout their session by providing feedback after each action taken by the student. Moreover, ambiguities regarding the production rules being used by students were resolved by querying the student. By committing to one production rule at a time and enforcing students to remain on correct solution paths, the complexity of the plan recognition task in intelligent tutoring systems is substantially reduced.

Lastly, we mention works that use recognition techniques to model students' activities in Intelligent Tutoring Systems [20, 7, 21]. Our work is distinct from works on plan recognition in intelligent tutoring systems in several ways. First, ITS are more closed-ended from ELEs. Thus, students' activities with such software more constrained and less exploratory, and are easier to model and recognize. In addition, the tutoring systems described above provided constant feedback to students which helped them remain on correct solution paths that are recognizable by the model used. Second, the tutoring systems described above explicitly modeled all possible solution plans for solving a specific problem. This is not possible in the VirtualLabs domain, as there may be an infinite number of possible plans for solving a problem.

2.2 On-line Plan Recognition in Artificial Intelligence

We now discuss general work from Artificial Intelligence that is concerned with plan recognition in general, rather than recognizing students' activities in pedagogical software. On-line plan recognition is a significantly more difficult task than its off-line variant. The fact that the interaction sequence is not observed ahead of time raises additional challenges to on-line plan recognition. Blaylock et al. [4] developed an algorithm to infer the goal of users from their actions in a Linux shell environment. Pynadath [19] proposes a probabilistic inference of plan, but requires the observations to be fully ordered. The approach by Bui [5] used particle filtering to provide approximate solutions to on-line plan recognition problems. Avrahami and Kaminka [3] presented a symbolic on-line plan recognition algorithm which keeps history of observations and commits to the set of possible plans only when it is explicitly needed for querying. Geib and Goldman presented PHATT [14], a probabilistic on-line plan recognition algorithm that builds all possible plans incrementally with each new observation. This algorithm was applied to recognizing users' strategies in real-time video games [17].

All of these works have been evaluated on simulated, synthesized problems [19, 3, 14] or on toy problems [4, 17]. These approaches do not scale to the complexities of real-world domains. An exception is the work of Conati et al. [8, 18] who used on-line plan recognition algorithms to infer students' plans to solve a problem in an educational software for teaching physics, by comparing their actions to a set of predefined possible plans. Unfortunately, the number of possible plans grow exponentially in the types of domains we consider, making it unfeasible to apply this approach.

3. PLANS AND EXPLANATIONS

In this section we provide the basic definitions that are required for formalizing the on-line plan recognition problems in ELEs. Throughout the paper we will use an existing ELE for chemical education called VirtualLabs to demonstrate our approach which is actively used by students worldwide as part of their introductory chemistry courses. VirtualLabs allows students to design and carry out their own experiments for investigating chemical processes by simulating the conditions and effects that characterize scientific inquiry in the physical laboratory [22]. We use the following problem called "Oracle", which is given to students:

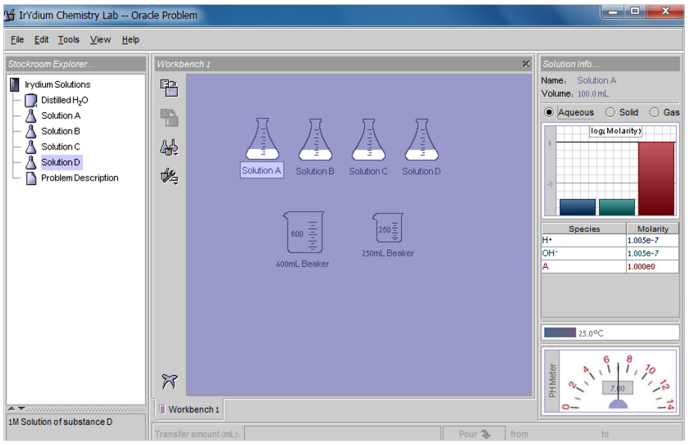


Figure 1: Snapshot of VirtualLabs

(a) $\underline{\mathsf{MSD}}[s_1 + s_2, d] \rightarrow \underline{\mathsf{MSD}}[s_1, d], \underline{\mathsf{MSD}}[s_2, d]$
(b) $\underline{\mathsf{MIF}}[s_1, d_2] \rightarrow \underline{\mathsf{MSD}}[s_1, d_1], \underline{\mathsf{MSD}}[d_1, d_2]$
(c) $\underline{\mathsf{MSD}}[s, d] \rightarrow \underline{\mathsf{MIF}}[s, d]$
(d) $\underline{\mathsf{MSD}}[s, d] \rightarrow \mathsf{MS}[s, d]$

Figure 2: Recipes for VirtualLabs

> Given four substances A, B, C, and D that react in a way that is unknown, design and perform virtual lab experiments to determine the correct reaction between these substances.

The flexibility of VirtualLabs affords two classes of solution strategies to this problem (and many variations within each). In the first strategy, a student mixes all four solutions together, and infers the reactants by inspecting the resulting solution. In the second strategy, a student mixes pairs of solutions until a reaction is obtained. A snapshot of a student's interaction with VirtualLabs when solving the Oracle problem is shown in Figure 1.

3.1 Definitions

We make the following definitions taken from the classical planning literature [16]. We use the term *basic actions* to define rudimentary operations that cannot be decomposed. These serve as the input to our plan recognition algorithm. For example, the basic "Mix Solution" action ($\mathsf{MS}_1[s = 1, d = 3]$) describes a pour from flask ID 1 to flask ID 3. A *log* is the output of a student's interaction. It is a sequence of basic level actions representing students' activities'. This is also the input to the plan recognition algorithm described in the next section.

Complex actions describe higher-level, more abstract activities that can be decomposed into sub-actions, which can be basic actions or complex actions themselves. For example, the complex action $\underline{\mathsf{MSD}}[s = 1 + 5, d = 3]$ (as shown in Figure 3) represents separate pours from flask ID 1 and 5 to flask ID 3.

A *recipe* for a complex action specifies the sequence of actions required for fulfilling the complex action. Figure 2 presents a set of basic recipes for VirtualLabs . In our notation, complex actions are underlined, while basic actions are not. Actions are associated with parameters that bind to recipe parameters. Recipe (a) in the figure, called Mix to Same Destination ($\underline{\mathsf{MSD}}$), represents the activity of pouring from two source flasks (s_1 and s_2) to the same destination flask (d). Recipe (b), called Mix via Intermediate Flask ($\underline{\mathsf{MIF}}$), represents the activity of pouring from one source flask (s_1) to a destination flask (d_2) via an intermediate flask (d_1).

Recipes can be recursive, capturing activities that students can repeat indefinitely. Indeed, this is a main characteristic of students' use of ELEs. For example, the constituent actions of the complex action $\underline{\mathsf{MSD}}$ in recipe (a) decompose into two separate $\underline{\mathsf{MSD}}$ actions. In turn each of these actions can itself represent a Mix to Same-Destination action, an intermediate-flask pour (by applying recipe (c)) or a basic action mix which is the base-case recipe for the recursion (recipe (d)). Recipe parameters also specify the type and volume of the chemicals in the mix, as well as temporal constraints between constituents, which we omit for brevity.

More generally, the four basic recipes in the figure can be permuted to create new recipes, by replacing $\underline{\mathsf{MSD}}$ on the right side of the first two recipes with $\underline{\mathsf{MIF}}$ or MS. An example of a derivation is the following recipe for creating an intermediate flask out of a complex Mix to Same Destination action and basic Mix Solution action.

$$\underline{\mathsf{MIF}}[s_1, d_2] \rightarrow \mathsf{MSD}[s_1, d_1], \mathsf{MS}[d_1, d_2] \quad (1)$$

These recipes may be combined to describe the different solution strategies by which students solve problems in VirtualLabs (e.g., capturing students mixing all possible substance pairs versus mixing all four pairs together).

A set of nodes N *fulfills* a recipe R if there exists a one-to-one matching between the constituent actions in R and their parameters to nodes in N. For example, the nodes $\mathsf{MS}_3[s = 5, d = 4]$ and $\mathsf{MS}_5[s = 4, d = 3]$ fulfill the Mixing via an Intermediate Flask recipe shown in Equation 1.

3.2 Planning

Planning is the process by which students use recipes to compose basic and complex actions towards completing tasks using VirtualLabs . Formally, a *plan* is an ordered set of basic and complex actions, such that each complex action is decomposed into sub-actions that fulfill a recipe for the complex action. Each time a recipe for a complex action is fulfilled in a plan, there is an edge from the complex action to its sub-actions, representing the recipe constituents.

Figure 3 shows part of a plan describing part of a student's interaction when solving the Oracle problem. The leaves of the trees are the actions from the student's log, and are labeled by their order of appearance in the log. For example, the node labeled with the complex action $\underline{\mathsf{MSD}}[s = 1 + 5, d = 3]$ includes the activities for pouring two solutions from flask ID 1 and ID 5 to flask ID 3. The pour from flask ID 5 to 3 is an intermediate flask pour ($\underline{\mathsf{MIF}}[s = 5, d = 3]$) from flask ID 5 to ID 3 via flask ID 4. The root of the plan represents te complex action of pouring three substances from flasks ID 1, 5 and 6 to flask ID 3.

In a plan, the constituent sub-actions of complex actions may interleave with other actions. This way, the plan combines the free-order nature of VirtualLabs recipes with the exploratory nature of students' learning strategies. Formally, we say that two ordered complex actions *interleave* if at least one of the sub-actions of the first action occurs after some sub-action of the second action. For

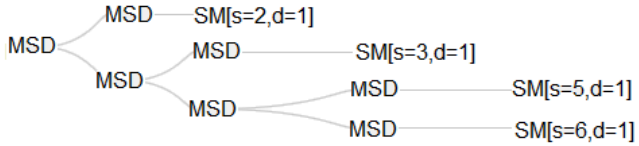


Figure 4: Example of an Explanation containing a Single Plan

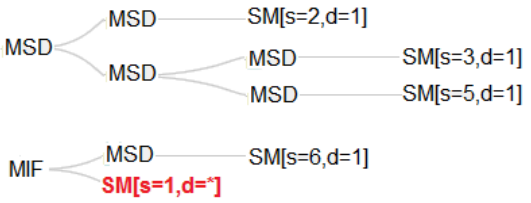


Figure 5: Example of an Explanation containing Two Plans, one of which has an open frontier

example, the nodes $\mathsf{MS}_3[s=5,d=4]$ and $\mathsf{MS}_5[s=4,d=3]$ and $\mathsf{MS}_2[s=6,d=8]$ and $\mathsf{MS}_4[s=8,d=3]$ both fulfill the Mixing via an Intermediate Flask recipe shown in Equation 1, but they are interleaved in the log. This interleaving quality makes the plan recognition task even more challenging.

4. ONLINE PLAN RECOGNITION

In this section we address the problem of on-line recognition in which agents' plans need to be inferred in real-time during execution. On-line recognition is essential for settings in which it is necessary to generate interventions to users. In ELEs, such intervention can provide feedback to students about their progress, alerting them to recurring mistakes or giving them hints about next steps during their exploration.

The fact that the interaction sequence is not known in advance requires to maintain the set of all plans that can explain the observations, including leaving place-holders for actions in the plan that relate to unseen future activities. Following Geib and Goldman [12], we define an *explanation* of actions O at time t a set of plans, such that there is an injective mapping from each action in O to a leaf in one of the plan instances. Each plan in an explanation describes a non-overlapping subset of the actions O. Some leaves in an explanation may not be included in O, and describe actions that are expected to appear in the future. These leaves are called the *open frontier* of the plan.

To illustrate, consider the recipes for VirtualLabs and the following explanations: Figure 4 shows a possible explanation for the observation sequence $SM[s=2,d=1], SM[s=3,d=1], SM[s=5,d=1], SM[s=6,d=1]$ in which all of the actions are constituents of the complex action MSD.[1] The explanation consists of a single plan.

Figure 5 shows a possible explanation for the same observation sequence, but in this case, the explanation consists of two plans. Here, the bold action $SM[s=1,d=*]$ represents a future (unseen) observation and is in the plan frontier. If the fifth observation turns out to be an SM action with $s=1$ (the parameter d does not hold any constraints), then the algorithm will incrementally combine this observation into the explanation. Otherwise, a third plan instance will be added to the explanation that matches the new observation, leaving $SM[s=1,d=*]$ in (and possibly adding new actions to) the plan frontier. We note that the plan frontier may also include complex actions, allowing to reason about future higher-level activities for which none of the constituents have been observed. The fact that the algorithm needs to maintain explanations for unseen observations is a significant computational challenge, as the possible number of explanations grows exponentially with the number of observations.

[1]For expository purposes we have omitted the parameters from nodes above the leaves.

5. CRADLE AND PHATT

The purpose of this section is to describe the state-of-the art in online plan recognition approach called PHATT, and our proposed extension to this approach for recognizing students' activities in ELEs.

We define the on-line plan recognition as follows: Given a set of observation at time t, output a set of explanations such that each explanation in the set can be used to derive the observations. PHATT is a top-down probabilistic algorithm that incrementally builds the set of possible explanations for explaining an observation sequence. PHATT works as follows: For each observation o^{t+1}, it takes the set of the possible explanations for the previous observations O^t, and tries to incorporate the new observation into each of the explanations in the set. This can be done either by integrating the new observation into one of the existing plans of the explanation, or by adding the observation as the first step of a new plan that will be added to the forest of plans in the explanation.

5.1 Using Filters

We now describe the basis for our proposed extension to PHATT, which is constraining the space of possible explanations in a way that reflect students' use of educational software. Our approach is called CRADLE (Cumulative Recognition of Activities and Decreasing Load of Explanations).[2]

Cradle extents the PHATT algorithm by constraining the space of possible explanations. We designed several "filters" that reduce the size of the explanation set in a way that reflects the intended use of plan recognition in ELEs. Specifically, the filters aim to produce complete, parsimonious and coherent explanations of students' interactions that can be easily understood by teachers and education researchers. We detail these filters below:

Explanation size This filter prefers explanations with smaller number of plans. Specifically, we discard explanations in which the number of plans is larger than a pre-computed threshold (the average number of plans per explanation).

Aging This filter prefers explanations in which successive observations extend existing sub-plans in the explanation rather than generate new plans. We discard explanations in which observations have not extended an existing plan for a given number of iterations.

[2]Also, cradle is the name of the mechanical contrivance used in placer mining, consisting of a box on rockers and moved by hand, used for washing out the gold-bearing soil, leaving only nuggets of gold.

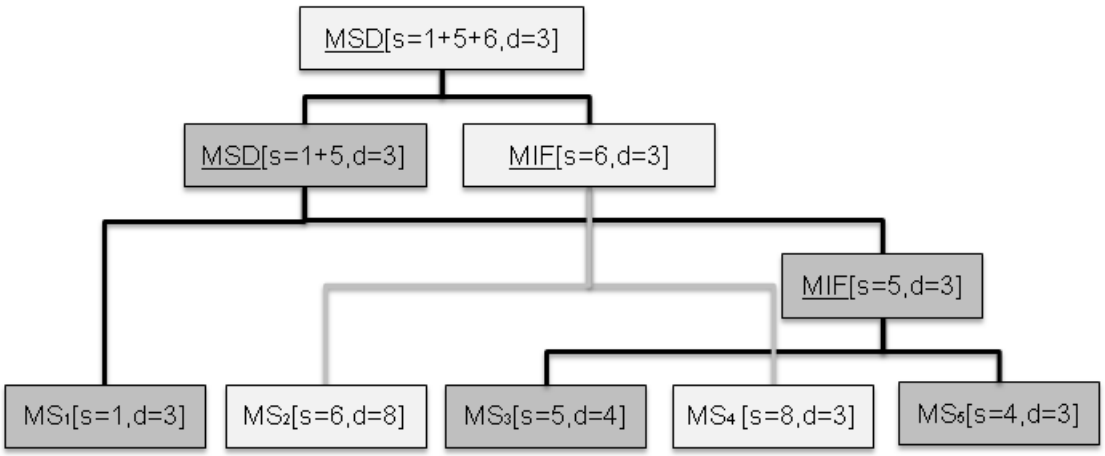


Figure 3: A partial plan for a student's log

Frontier size This filter prefers explanations which makes fewer commitments about future observations. It measure the amount of actions in the frontier that exist in each explanation, and discard explanations where this amount is above the average.

Probability This filter prefers explanations with a higher likelihood. It discards explanations whose probability of generating the observation sequence is lower than the average probability of the other explanations.

5.2 Augmenting PHATT

Figure 6 describes how CRADLE extends PHATT using the following methods, which we outline in some level of abstraction.

- Expand. Given a set of explanations that derive O^t, it is given a new observation o^{t+1}, this method creates all possible subplans in which o^{t+1} is a leaf, and tries to combine each of these subplans in all possible ways to each explanation. Each such subplan can be combined in two ways: (1) combineInExistingTrees - if the root of the subplan matches one of the plan frontier items, it replaces the frontier item with the subplan (replacing the placeholder with a concrete observation) or (2) extendWithANewTree - if the root of the subplan matches a possible goal, it is adding the subplan as the top levels of a new plan in the explanation's forest of plans.
- Filter. This function takes a set of explanations, calculates the average age, frontier size and amount of trees per explanation and filtered away all explanations with values above average. This means it prefers explanations with small frontier (less future expected observations), small age (observations continue existing plans instead of creating new ones) and small amount of trees (observations related to the same plan rather than describe different plans).
- Main. This is the main function of the new recognition process. It is made out of the two previous described stages - Extend and Filter - performed alternatly for each new observation encountered.

6. EMPIRICAL METHODOLOGY

The purpose of this section is to evaluate CRADLE to PHATT algorithm for real-world data sets of students' interactions with VirtualLabs . The PHATT approach is representative of an array of algorithms in the literature for performing on-line plan recognition by maintaining sets of observations (see for example the ELEXIR and YAPPR algorithm [13, 15]) and would behave similarly on our ELE data sets.

Specifically, we sampled 16 logs of students' interactions who solved two problems. The first was the Oracle problem described earlier. The second problem was called "Unknown Acid" and required students to determine the concentration level of an unknown solution. The length of the logs were chosen to have a wide range, between 4 to 152 actions.

6.1 Completeness and Run-time

The number of explanations maintained by the PHATT approach grows exponentially in the number of observations. It can be shown that for n observations and a set g of possible extensions for an explanation, the number of possible explanations is bounded by $n * |g|^n$. To illustrate, a 4 observation log outputted 142 different explanations, and a log of 12 observations generated more than 10,000 explanations. Most of these explanations included an abundance of plan instances with extremely large frontiers, clearly not the most coherent descriptions of the students' work.

Figure 7 shows the performance obtained using PHATT, augmentation of PHATT with single filter, and CRADLE. The x-axis in the figure corresponds to ranges of different log sizes. The y-axis determines the success ratio by measuring whether the algorithm was able to terminate and produce the explanations describing the student's activities within an upper bound of two hours of CPU time. As shown by the figure, PHATT was not able to terminate on logs

```
1: function EXPAND(o, Exps)  ▷ o: a new observation, Exps is the set of all explanations until o
2:     newExps = []
3:     for all explanation e ∈ Exps do
4:         newExps += e.combineInExistingTrees(o)
5:         newExps += e.extendWithANewTree(o)
6:     end for
       return newExps
7: end function

8: function FILTER(Exps)  ▷ Exps is the set of all explanations collected so far
9:     filteredExps = []
10:    for all explanation e ∈ Exps do
11:        if e.age ≤ averageAge & e.frontierSize ≤ averageFrontierSize & e.trees ≤ averageTrees then
12:            filteredExps += e
13:        end if
14:    end for
       return filteredExps
15: end function

16: function MAIN(Obs)  ▷ Obs is the set of all observations
17:    tempExps = [⟨emptyExp⟩]  ▷ Only one explanation - the empty explanation
18:    for all observation o in Obs do
19:        allExps = Expand(o, tempExps)
20:        filteredExps = Filter(allExps)
21:        tempExps = filteredExps
22:    end forreturn tempExps
23: end function
```

Figure 6: Main functions of the CRADLE algorithm

over 4 actions within this designated time frame. In contrast, CRADLE was able to significantly increase the performance of PHATT algorithm by applying the filters. Specifically, applying the different filters independently allowed to improve the success ratio for some of the logs, with the highest improvement attributed to the CRADLE approach which applied the age, frontier size and explanation size filters. Interestingly, there was not a single filter method that outperformed all of the other methods for all log size.

Next, we compare the run-time of CRADLE and PHATT on fragments of logs for which PHATT was able to terminate. Figure 8 shows the average run-time on each size of log, measured in seconds, presented in a logarithmic scale. It can be seen that the average run-time of CRADLE is exponentially better than the average run-time of PHATT for the aforementioned logs.

6.2 Domain Expert Evaluation

In this section, we show that although the CRADLE approach reduces the number of possible explanations that is maintained by the plan recognition algorithm, it does not hinder the correctness of the algorithm. To this end, we sampled 20 logs of the Oracle problem and presented the output of the PHATT and CRADLE approach to a domain expert.[3]. We ran the cut logs on PHATT and CRADLE and collected the outputted set of explanations for each log. For

[3]Logs of length greater than 6 actions were cut arbitrarily at 6,7,9,10 and 11 actions, in order to simulate incomplete interaction sequences and to allow PHATT to terminate on these logs in reasonable time.

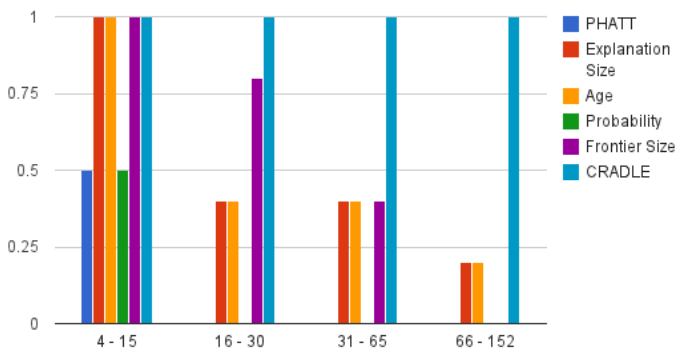


Figure 7: Performance of PHATT, CRADLE and Single Filter Variants on Various Log Sizes

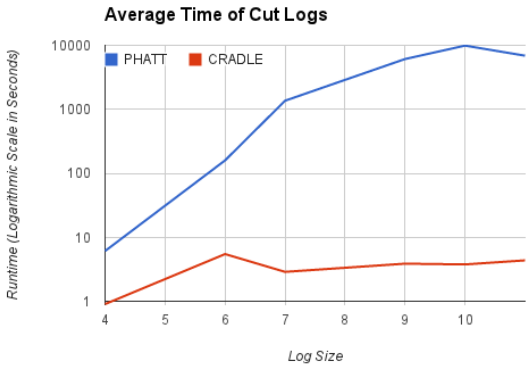


Figure 8: Runtime of PHATT and CRADLE

each of the approaches, we chose to present the domain expert with the explanation that did not include an open frontier (that is, the explanations provided a complete description of the activities of the student). If there was no explanations without an open frontier, we chose the most likely explanation as measured by its probability.

Out of the 20 examined logs, in 9 logs PHATT and CRADLE's explanations were the same (though CRADLE was able to output the solution exponentially faster). We presented the explanations for which CRADLE and PHATT different to a domain expert, who is one of the developers of the VirtualLabs software, who compared between the two explanations. We did not label the explanations with the algorithm that generated them. In 8 out of these 11 logs, the domain expert preferred explanations which were presented by CRADLE over the explanations of PHATT. In one case, the domain expert said none of the explanations describe the activities of the student correctly. To illustrate, Figure 4 shows the explanation outputted by CRADLE for a particular log which included a mix of 4 substances into a single flas. Figure 9 shows the PHATT explanation for that same log, using two plans to explains the observation sequence. In this case, the domain expert preferred the CRADLE explanation, which explained the observation sequence using a single plan.

7. DISCUSSION AND FUTURE WORK

Our results show that the CRADLE approach was able to extend the state-of-the-art (PHATT algorithm) towards successfully rec-

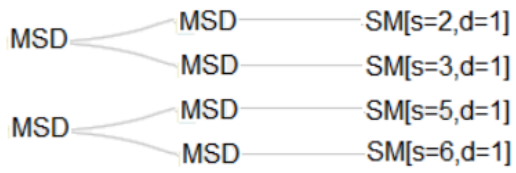


Figure 9: Example of PHATT explanation

ognizing students' activities in an ELE for chemistry education. We showed that CRADLE was able to produce better explanations than PHATT, and with exponentially faster running time. Specifically, the ouputted explanations of CRADLE were as good as or better than PHATT in 18 out of the 20 logs that we sampled, giving CRADLE a success rate of 90% at an exponentially lower runtime. The paper demonstrate that on-line plan recognition in ELEs is a challening computational problem, and show the efficacy of the CRADLE approach in addresssing these problems by reducing the number of explanations maintaind by the algorithms in an intelligent way. We are currently pursuing work with CRADLE in several directions. First, we are evaluating the scalability of the CRADLE approach by evaluating it with different ELEs for statistics education, as well as simulated data that simulates users' interactions with software. This ELE is significantly different than VirtualLabs in that student's interactions are more likely to engage in trial-and-error, which we predict will further challenge the recognition problem. Second, we are developing a formal langauge that explains students' activities with ELEs that will help us construct more accurate grammars for the recognition algorithms.

8. ACKNOWLEDGEMENTS

This work was supported in part by Israeli Science Foundation Grant no. 1276/12.

9. REFERENCES

[1] S. Amershi and C. Conati. Automatic recognition of learner groups in exploratory learning environments. In *Intelligent Tutoring Systems (ITS)*, 2006.

[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The Journal of Learning Sciences*, 4(2):167–207, 1995.

[3] D. Avrahami-Zilberbrand, G. Kaminka, and H. Zarosim. Fast and complete symbolic plan recognition: Allowing for duration, interleaved execution, and lossy observations. In *Proc. of the AAAI Workshop on Modeling Others from Observations, MOO*, 2005.

[4] N. Blaylock and J. F. Allen. Statistical goal parameter recognition. In *ICAPS*, volume 4, pages 297–304, 2004.

[5] H. H. Bui. A general model for online probabilistic plan recognition. In *IJCAI*, volume 3, pages 1309–1315, 2003.

[6] M. Cocea, S. Gutierrez-Santos, and G. Magoulas. S.: The challenge of intelligent support in exploratory learning environments: A study of the scenarios. In *Proceedings of the 1st International Workshop in Intelligent Support for Exploratory Environments on European Conference on Technology Enhanced Learning*, 2008.

[7] C. Conati, A. Gertner, and K. VanLehn. Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.

[8] C. Conati, A. Gertner, and K. Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction*, 12(4):371–417, 2002.

[9] A. Corebette, M. McLaughlin, and K. C. Scarpinatto. Modeling student knowledge: Cognitive tutors in high school and college. *User Modeling and User-Adapted Interaction*, 10:81—108, 2000.

[10] Y. Gal, S. Reddy, S. Shieber, A. Rubin, and B. Grosz. Plan recognition in exploratory domains. *Artificial Intelligence*, 176(1):2270 – 2290, 2012.

[11] Y. Gal, E. Yamangil, A. Rubin, S. M. Shieber, and B. J. Grosz. Towards collaborative intelligent tutors: Automated recognition of users' strategies. In *Intelligent Tutoring Systems (ITS)*, 2008.

[12] C. Geib and R. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[13] C. W. Geib. Delaying commitment in plan recognition using combinatory categorial grammars. In *IJCAI*, pages 1702–1707, 2009.

[14] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[15] C. W. Geib, J. Maraist, and R. P. Goldman. A new probabilistic plan recognition algorithm based on string rewriting. In *ICAPS*, pages 91–98, 2008.

[16] B. Grosz and S. Kraus. The evolution of sharedplans. *Foundations and Theories of Rational Agency*, pages 227–262, 1999.

[17] F. Kabanza, P. Bellefeuille, F. Bisson, A. R. Benaskeur, and H. Irandoust. Opponent behaviour recognition for real-time strategy games. In *Plan, Activity, and Intent Recognition*, 2010.

[18] S. Katz, J. Connelly, and C. Wilson. Out of the lab and into the classroom: An evaluation of reflective dialogue in andes. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 158:425, 2007.

[19] D. V. Pynadath and M. P. Wellman. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 507–514. Morgan Kaufmann Publishers Inc., 2000.

[20] K. VanLehn, C. Lynch, K. Schulze, J. A. Shapiro, R. H. Shelby, L. Taylor, D. J. Treacy, A. Weinstein, and M. C. Wintersgill. The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education*, 15(3), 2005.

[21] M. Vee, B. Meyer, and K. Mannock. Understanding novice errors and error paths in object-oriented programming through log analysis. In *Proceedings of Workshop on Educational Data Mining at ITS*, pages 13–20, 2006.

[22] D. Yaron, M. Karabinos, D. Lange, J. Greeno, and G. Leinhardt. The ChemCollective–Virtual Labs for Introductory Chemistry Courses. *Science*, 328(5978):584, 2010.

What is the Source of Social Capital?

The Association Between Social Network Position and Social Presence in Communities of Inquiry

Vitomir Kovanovic*
School of Interactive Arts and Technology
Simon Fraser University
250 - 13450 102nd Avenue
Surrey, BC, V3T0A3 Canada
vitomir_kovanovic@sfu.ca

Srecko Joksimovic
School of Interactive Arts and Technology
Simon Fraser University
250 - 13450 102nd Avenue
Surrey, BC, V3T0A3 Canada
sjoksimo@sfu.ca

Dragan Gasevic
School of Computing Science
Athabasca University
1 University Drive
Athabasca, AB, T9S 3A3
Canada
dgasevic@acm.org

Marek Hatala
School of Interactive Arts and Technology
Simon Fraser University
250 - 13450 102nd Avenue
Surrey, BC, V3T0A3 Canada
mhatala@sfu.ca

ABSTRACT

It is widely accepted that the social capital of students – developed through their participation in learning communities – has a significant impact on many aspects of the students' learning outcomes, such as academic performance, persistence, retention, program satisfaction and sense of community. However, the underlying social processes that contribute to the development of social capital are not well understood. By using the well-known Community of Inquiry (CoI) model of distance and online education, we looked into the nature of the underlying social processes, and how they relate to the development of the students' social capital. The results of our study indicate that the affective, cohesive and interactive facets of social presence significantly predict the network centrality measures commonly used for measurement of social capital.

General Terms

Social Network Analysis, Community of Inquiry, Social Presence

1. INTRODUCTION

Asynchronous online discussions have been frequently used both in blended and fully online learning [41]. However, with the broader adoption of social-constructivist pedagogies and the shift towards the collaborative learning [2], they are viewed as one of the important study tools for the computer-supported collaborative learning (CSCL) within the online learning environments. Their use has produced an enormous amount of data about the interactions between students and instructors [21]. The distance education and CSCL research communities have tried to use these data for gaining insights into the very complex nature of the learning phenomena. Among the different ways of researching students' social interactions *Quantitative Content Analysis* (QCA) [38, 19] and *Social Network Analysis* (SNA) [52, 46] represent two commonly used methods.

A widely accepted model of distance education which makes a use of QCA is the *Community of Inquiry* (CoI) model [28]. According to Garrison and Arbaugh [30], it is one of the leading models of distance education that describes the key constructs of the overall educational experience. The CoI model provides the in-depth assessment of teaching, cognitive and social dimensions of learning phenomena, and how those three dimensions affect: i) the overall success of the learning process, and ii) the attainment of learning objectives [28]. Empirical research showed that the social dimension of learning plays an important role in the learning communities by mediating the relationship between the teaching and cognitive dimensions [31]. Still, the CoI model does not explicitly address the question of student social networks, their structure, or the effects they have on the overall educational experience and learning outcomes. Given the amount of evidence from the studies of student social networks [46], this warrants further investigation.

One of the central aspects in the study of social networks is the idea of the *social capital* [13, 12]. Generally speaking, social capital can be defined as a value resulting from occupying a particularly advantageous position within a social network [12]. Over the years, the study of social capital has become increasingly popular in the field of education [14]. The large number of studies in the distance education field indicated an important connection between the students' social capital and many important aspects of education and learning including academic performance [33, 15, 7, 49, 43], retention [23], persistence [50], program satisfaction [7], and sense of community [17]. Still, research of the student social networks have involved mostly isolated studies that were focused on the understanding of the relationship between a particular set of constructs selected by the researchers and the students' network position. Likewise, the underlying mechanisms responsible for the observed social structure are typically not addressed, which is understandable given the lack of educational theories that explicitly

*Corresponding Author

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

take into the consideration student social networks.

In this paper, we present the results of the study which explored the links between the CoI model and the social network analysis of student networks. With the current advancement within the CoI research and most recent validations of the model [31], the model is mature enough and empirically sound to provide this missing theoretical foundation for understanding the structure of students' social networks. Likewise, the understanding of the structure of social networks can provide a more comprehensive overview of the social dimension of learning that it is already accounted for in the research of the CoI model.

Given the exploratory nature of this study, we focused on the relationship between social capital and social processes which are indicative of the student social presence development. The main question we aim to answer, in this paper is *which social processes, and to what extent, are indicative of the development of the social capital in a communities of inquiry*? Given the detailed characterization of social aspects of learning in the CoI model through the construct of *social presence*, we explored how this construct relates to the students' social capital, as characterized by their position in social networks formed around communities of inquiry. As the community of inquiry provides characterization of different sociological processes that constitute social presence, we looked how each of them contributed to the development of social capital withing students' social network.

2. THEORETICAL BACKGROUND

2.1 Social network analysis

2.1.1 Social capital

The study of social networks has attracted much attention in social and behavioral sciences [17, 14]. The focus in social network analysis is on the study of *relationships*, also known as *ties*, between a set of *actors*, or *participants* [14]. Through the relationships, members of a network engage in sharing, exchange or delivery of various resources including information [36]. Social network analysis draws much of its ideas from the mathematical graph theory and the sociometric studies of the human relationships [52].

An important concept in the study of social networks is the idea of *relation strength* [34], which is used to make a distinction between *strong social ties*, which require a substantial commitment (e.g., family, close friends), and *weak social ties* which do not obligate a strong commitment (e.g., acquaintances). Likewise, the idea of *network brokerage* builds on the fact that in a large network, the density of relationships is not uniform, which indicates the existence of smaller sub-communities within a large social network [12, 13]. In his seminal paper, Granovetter [34] stressed the tremendous importance of weak social ties, as they provide access to novel information from different parts of a social network and provide pathways of information exchange between sub-communities. An individual who possesses a large number of weak ties in many different sub-communities is able to take advantage by combining diverse information coming from different sub-communities, and to even control to a certain degree the spread of information from one sub-community to another [12]. This ability to create a value from occupying a particular position in a social network is known as *social capital* [13]. To study and assess values of different network positions, the principles of graph theory are the most commonly used [52]. The notion of *centrality* is particularly important. This notion captures the relative importance of individuals in social networks [52]. Given the complexity of measuring actors' relative importance, a large number of centrality measures were proposed over the years out of which degree, closeness and betweenness centralities are the most frequently used [26].

2.1.2 Social network analysis in education

While social network analysis has been widely adopted in social and behavioral sciences, its adoption in the field of education was initially very limited [14]. According to Carolan [14], the main reasons for this are *"overemphasis on individual explanations of educational opportunities and outcomes, a quest for scientific legitimacy, and a preference for experimental designs that estimate the causal effects of 'educational interventions' "* [14, 32]. Nevertheless, over the years, the number of studies that indicated the importance of social connections on the overall academic experience has grown considerably. A good example is the study of students' overall academic experience from early 1990s by Astin [5] in which he concluded that: i) the environment made by the instructors and students is crucial, and ii) the single most important environmental influence is *peer group*.

In the context of distance education, there have been many studies recently that looked at the connection between several important learning constructs and social capital of students. Likewise, in the fields of educational data mining (EDM) [6] and learning analytics [40], the interest in SNA has been growing. The recent review of the EDM field by Romero and Ventura [44] noted a growing interest in SNA; likewise, in the learning analytics community, SNA was recognized as one of the most important techniques of social learning analytics [11, 25].

As expected, academic performance was the focus of a large majority of the studies [33, 50, 15, 7, 49, 43] that have found positive effects of student positions in social networks on academic performance. Still, academic performance was not the only construct that was examined. The study of retention by Eckles and Stradley [23] found that for each friend that leaves an academic degree program makes a student five times more likely to leave as well, while every friend who stays makes a student 2.25 times more likely to also stay in college. The study of student persistence and integration by Thomas [50] found that students with a broader set of acquaintances are more likely to persist in the academic program of a higher education institution, and that students with a higher proportion of ties outside their peer group also perform better academically. This is aligned with the findings of Dawson [17] who showed that students' sense of community membership was positively related to their closeness and degree centrality measures. Similarly, in the study of a team-based MBA program by Baldwin et al. [7], it was found that the high embeddedness in the friendship network increased students' perception of learning and enjoyment in the program; as well, the centrality in the communication networks was found to be positively linked with the student grades.

One important thing to notice is that the majority of the studies did not draw their theoretical foundations of network formation from the established educational theories. As pointed out by Rizzuto et al. [43], there is a lack of *"theory of academic performance that combines individual characteristics as well as social and infrastructural factors"* (p180). The main exception is the use of retention theories by Tinto [51] and Bean [8] in the study of student persistence and retention. The other notable theories that are adopted, such as Feld's theory of focused choice [24], or Lin's theory of social resources [39] are general sociological theories that do not take into the account the specific of learning processes and educational contexts.

2.2 The community of inquiry (CoI) model

2.2.1 Overview

The Community of Inquiry (CoI) model is a general model of distance education which explains the constructs that contribute to the overall learning experience. It is rooted in the social constructivist

philosophy, most notably in the work of John Dewey [20], and is particularly well suited for understanding different aspects of learning within the learning communities. The main goal of the CoI model was to define the constructs that characterize a worthwhile educational experience, and a methodology for their assessment. The CoI model consists of the three interdependent constructs, also known as *presences*, that together provide a comprehensive coverage of the distance learning phenomena:

1) **Cognitive Presence** explains different phases of students' knowledge construction process through social interactions within a learning community [28].
2) **Teaching Presence** describes the instructor's role in course delivery and during course design and preparation [3].
3) **Social Presence** explains the social relationships and the social climate within a learning community that have a significant effect on the success and quality of social learning [45].

The CoI model is well-researched and widely accepted within the distance learning research community as shown by a recent two-part special issue of The Internet and Higher Education journal [1]. The model defines its own coding schemes that are used to assess the levels of the three presences through the QCA in transcripts of asynchronous online discussions. More recently, instead of relying on the QCA, a CoI survey instrument [4] was developed as an alternative way of assessing the levels of the three presences.

2.2.2 *Social presence*

Social presence is defined as the *"ability of participants in a community of inquiry to project themselves socially and emotionally, as "real" people (i.e., their full personality), through the medium of communication being used"* [28, p3]. Critical thinking, social construction of knowledge and the development of the cognitive presence are more easily developed in the cases where the appropriate levels of social presence have been established [28].

Given the form of delivery in distance education, face-to-face communication that is typical for more traditional forms of education delivery is not possible. Hence, establishing and sustaining social presence is more challenging. Distance education was often criticized as being inferior to more traditional forms of education, particularly because of the inability to create social presence between the members of a learning community [2]. However, according to Garrison et al. [28], the form of communication is not the solely factor determining the development of social presence. A key aspect of establishing social presence in face-to-face settings are visual cues, while participants in online communities use different techniques – such as emoticons – to convey the affective dimension of communication that lacks in typical text-based communications.

As described by Rourke et al. [45], the origins of social presence can be found in the work of Mehrabian [42] and his notion of *immediacy* which is defined as *"the extent to which communication behaviors enhance closeness to and nonverbal interaction with another"* [42, p203]. This, and the set of follow-up studies by communication theorists, defined the theoretical background on which the construct of social presence was based [45]. The social presence in the CoI model is defined as consisting of three different dimension of communication:

1) **Affectivity and expression of emotions**: Since emotions are strongly associated with motivation and persistence, they are indirectly connected to critical thinking and communities of inquiry. More formally, emotional expression has been indicated by the *"ability and confidence to express feelings related to the educational experience"* [28, p99].
2) **Interactivity and open communication**: In order to promote the development of higher-order critical thinking skills, the notion that the other side is listening and attending is crucial [45]. Thus, activities such as praising of the student work, actions, or comments contribute to the teacher immediacy, which in turn leads to affective, behavioral and cognitive learning [45]. Similarly, open communication is defined as *"reciprocal and respectful exchanges of messages"* [28, p100] and together with interactivity provide a basis on which productive social learning can be established.
3) **Cohesiveness**: The activities that *"build and sustain a sense of group commitment"* [28, p101] define cohesiveness. The goal is to create a group where the members possess strong bonds to both i) each other and ii) the group as a whole. This in turn stimulates productive learning and the development of critical thinking skills.

Given that there are three different dimensions of social presence, the coding scheme for social presence (see Table 1) defines a list of indicators for each dimension. By looking at the content and the timing of each message, it is possible to see how the social climate unfolded during the course delivery. This provides a way of understanding and evaluating the different pedagogical interventions with respect to the development of a productive social climate in a learning community which enables for the meaningful social interactions [53].

2.3 Research Question: Characterization of social capital through social presence

As indicated in the previous sections, there is a strong evidence that social capital plays an important role in the shaping of the overall learning experience. The main research question that we investigate in this paper:

> What is the relationship between the students' *social capital*, as captured by social network centrality measures, and students' *social presence*, as defined by the three categories in the Community of Inquiry model?

The higher the social capital of a learner is, the more capable the learner is in terms of learning opportunities, information exchange, or integration within the academic environment. Still, the origins of social capital are not fully understood. Why certain students occupy advantageous positions in social networks? What are the social processes that enable them to take advantage of their social relationships? As for now, not a single theory of learning addresses the question of social capital directly, even though the impact of social context on learning is widely acknowledged.

As indicated by the previous study by de Laat et al. [18], content analysis techniques can be used in combination with SNA to provide a more comprehensive view of the social learning processes. In this paper, we propose the use of the Community of Inquiry model, given its holistic view of educational experience and extensive empirical evaluation by the research community [29], with the aim to characterize the origins of social capital in communities of inquiry. The CoI model description of important behavioral indices that contribute to the development of the positive social climate could be used to interpret the observed differences among students positions in a social network.

Likewise, the synergistic effect of using those two perspectives on student interactions provide a value for the CoI model by emphasizing the effects of the theorized social processes. For example, are interactivity and open communication important for the development of social capital? Are the students who show group cohesion the ones who take brokerage positions? Recently, there have been

Table 1: Social Presence Categories and Indicators as defined by Rourke et al. [45]

Category	Code	Name	Definition
Affective	A1	Expression of emotions	Conventional expressions of emotion, or unconventional expression of emotion, includes repetitions punctuation, conspicuous capitalization, emoticons.
	A2	Use of humor	Teasing, cajoling, irony, understatements, sarcasm.
	A3	Self-disclosure	Presenting details of life outside of class, or express vulnerability.
Interactive or Open Communication	I1	Continuing a thread	Using reply feature of software rather than starting a new thread.
	I2	Quoting from others' messages	Using software features to quote others entire messages or cutting and pasting selections of others' messages.
	I3	Referring explicitly to others' messages	Direct references to contents of others' posts
	I4	Asking questions	Students ask questions of other students or the moderator.
	I5	Complementing, expressing appreciation	Complimenting others or contents of others' messages.
	I6	Expressing agreement	Expressing agreement with others or content of others' messages.
Cohesive	C1	Vocatives	Addressing or referring to participants by name.
	C2	Addresses or refers to the group using inclusive pronouns	Addresses the group as *we,us, our, group*.
	C3	Phatics, salutations	Communication that serves a purely social function: greetings, closures.

some attempts [47, 48] that make use of SNA in conjunction with the CoI model to provide insights into particular aspects of learning, such as self-regulation [9]. Still, the central question of social capital is left unexplored and that is the goal in our study.

3. METHODS

3.1 Dataset

For our study, we used the dataset consisting of six offers (Winter 2008, Fall 2008, Summer 2009, Fall 2009, Winter 2010, Winter 2011) of the masters level software-engineering course offered through the fully online instructional condition at a Canadian open public university. The course is 13 weeks long, research-intensive, and focuses on understanding of current research trends and challenges in the area of software engineering. Students were requested: i) to participate in online discussions for which they received 15% of their final grade (see details in [32]), and ii) to work on a four tutor marked assignments. Overall, 81 student created the total of 1747 discussion messages which were then used as the main data source for this study. The total number of students and messages for all six course offerings are shown in Table 2.

Table 2: Course offering statistics

	Student count	Message count	Graph density
Winter 2008	15	212	0.52
Fall 2008	22	633	0.69
Summer 2009	10	243	0.84
Fall 2009	7	63	0.58
Winter 2010	14	359	0.84
Winter 2011	13	237	0.77
Average	13	291	0.71
Total	81	1747	

3.2 Social network measures

In order to measure students' social capital we extracted student social network graphs from the interactions on the discussion boards. We extracted *directed* social graphs, so that whenever a student $X1$ responded to a message from another student $X2$, we created a direct relationship between the two of them ($X1 \Rightarrow X2$). Since two students can exchange more than one message, we extracted a *weighted* graph where the weights corresponded to the number of exchanges between a given pair of students. We created a separate social graph for each of the course offerings independently and the graph densities for each offering are shown in Table 2.

From the constructed social network graphs, we extracted the three network centrality measures which are most frequently used for the study of the educational social networks [14]:

1) **Betweenness centrality** captures brokerage opportunities of actors in a network and is the most directly related to the social capital construct [13, 12]. For a given actor A, it is mathematically defined as the number of shortest paths between any two other actors that "pass through" the actor A [26].
2) **Degree centrality** measures the total number of relationships that each participant has [26]. Given that we constructed the directed social graphs, we considered separately the in-degree and out-degree centrality measures. They represent the total number of incoming and outgoing relations for a given individual, respectively. Degree is the simplest centrality measure, very easy to calculate, as it takes into account only the direct relationships between the actors [52].
3) **Closeness centrality** represents the distance of an individual participant in the network from all the other network participants [26]. It is defined as the inverse of the sum of the distances to all other participants [14], and hence takes into account both direct and indirect relationships [52]. Much like degree centrality, given that the student graphs are directed, we calculated the in-closeness and the out-closeness centrality measures. For a given actor A, in-closeness centrality measures how many indirect steps are needed for all other actors to reach the actor A, while out-closeness measures how many indirect steps the actor A requires in order to reach all the other actors in the network.

Table 3: Descriptive statistics of social network metrics

	$Mean$	SD	Min	Max
Betweenness	9.04	14.51	0.00	74.20
In-degree	19.84	8.62	4.00	42.00
Out-degree	19.86	9.37	3.00	44.00
In-closeness	0.09	0.04	0.04	0.17
Out-closeness	0.08	0.04	0.03	0.18

Table 3 shows the descriptive statistics for all five extracted centrality measures. We can see that on average the students wrote around 20 messages, and also received on average around 20 responses. This level of activity was expected, as by the course design the students were expected to spend a significant amount of time on the online discussions. Still, from the descriptive statistics reported in Table 3, we can observe the large differences between the individual students in the case of all five centrality measures.

3.3 Message coding

In order to assess students' social presence, all messages were manually coded by two coders in accordance with the coding scheme defined by Rourke et al. [45]. As the individual messages can

Table 4: Social Presence Indicators

Category	Code	Indicator	Count	Percent Agreement
Affective	A1	Expression of emotions	288 (16.5%)	84.4
	A2	Use of humor	44 (2.52%)	93.1
	A3	Self-disclosure	322 (18.4%)	84.1
Interactive	I1	Continuing a thread	1664 (95.2%)	98.9
	I2	Quoting from others messages	65 (3.72%)	95.4
	I3	Referring explicitly to other's messages	91 (5.21%)	92.7
	I4	Asking questions	800 (45.8%)	89.4
	I5	Complementing, expressing appreciation	1391 (79.6%)	90.7
	I6	Expressing agreement	243 (13.9%)	96.6
Cohesive	C1	Vocatives	1433 (82%)	91.8
	C2	Addresses or refers to the group using inclusive pronouns	144 (8.24%)	88.8
	C3	Phatics, salutations	1281 (73.3%)	96.1

Table 5: Social Presence Categories.

Category	Count	Percent Agreement
Affective	530 (30.3%)	80.8
Interactive (Excluded I1 and I5)	1030 (59%)	86.2
Cohesive (Excluded C1)	1326 (75.9%)	93.4

be simultaneously classified into more than one category of social presence, each message was coded with three binary codes indicating whether the message belongs to a particular social presence category. However, early in the coding process, we observed an extremely high frequency of some of the indicators in the cohesive and interactive categories. Because of this, almost all of the messages could be classified as both interactive and cohesive, which would limit the discriminatory power of those two categories. Thus, to resolve this issue, instead of coding on the levels of categories, the coding was done on the levels of the individual indicators, so that each message was coded with the twelve binary codes (i.e., three indicators of the affective category, six indicators of the interactive category and three indicators of the cohesive category) each indicating an occurrence of a particular social presence indicator within a given message. This enabled us to look at the distribution of the individual indicators and to be more selective in the type of the indicators that we wanted to investigate. Overall, the coding agreement was high, with all of the indicators reaching percent agreement of at least 84%, and all the coding disagreements were resolved through discussion between the coders in a follow-up meeting, after they first coded the messages independently. The coding results are shown in Table 4. The results show that some of the indicators were recorded in a disproportionately large number of messages. Thus, in order to evaluate different aspects of social presence captured by those three categories, we omitted some of the indicators from our analysis: i) Continuing a thread, ii) Complementing, expressing appreciation, and iii) Vocatives. We intentionally kept the "Phatics, salutations indicator" as its removal would render the cohesive category in only 8.24% of the messages. By using the remaining nine indicators, we categories all of the messages in the corpus, and the final results are shown in Table 5.

3.4 Statistical analysis

In order to investigate the relationships between the three categories of social presence, as defined by the CoI model, and *social capital*, as operationalized through the five network centrality measures, we conducted backward-stepwise multiple linear regression analyses [35] for each of the five extracted network centrality measures. To evaluate different regression models for a particular centrality measure, we used the popular Akaike Information Criterion (AIC) [35]. In order to control for the inflation of the Type-I error rate due to multiple statistical significance testing, we used the Holm-Bonferroni correction [37], also known as the sequential rejective Bonferroni correction. It provides a control for Type-I errors at a prescribed significance level – in our case $\alpha = 0.05$ – while providing a substantial increase in the statistical power over the commonly used Bonferroni correction [22]. In the case of testing the family of N null-hypothesis and significance level α, the Holm-Bonferroni method proceeds as follows:

1) Hypothesis with the smallest observed p-value, is tested using the adjusted significance level $\alpha' = \alpha/N$, in the same manner as in the traditional Bonferroni procedure.
2) However, the next smallest observed p-value is tested using differently adjusted significance level $\alpha' = \alpha/(N-1)$.
3) The same process repeats up to the hypothesis with the highest observed p-value which is tested using the unadjusted significance level α.
4) The important additional rule is that if any of the hypothesis in the family gets rejected, then *all the subsequent* hypotheses are rejected as well regardless of their observed p-values.

By using differently adjusted statistical significance levels, Holm-Bonferroni method guarantees that the family-wise error rate is kept at the prescribed level, while providing a significant increase in the statistical power over the more commonly used simple Bonferroni correction [22]. We used the Holm-Bonferroni correction for testing the overall significance of the regression models, and for testing the significance of the individual predictor variables. In our case, with five hypothesis tests, the values of the adjusted statistical significance levels were $\alpha = [0.01, 0.0125, 0.0167, 0.0250, 0.05]$.

We also inspected the QQ-Plots for the signs of the severe deviation from the normality of residuals, and we assessed the multicollinearity of the three predictor variables using the variance-inflation factors (VIFs). The QQ-Plots did not reveal deviations from the normality of the residuals and VIF values were substantially lower than the typically used thresholds such as 4 or 10 [10]. Thus, we considered the use of the multiple linear regression appropriate for our study.

4. RESULTS

The results of the regression analyses are shown in Table 6. The models for betweenness, in-degree, out-degree and in-closeness centralities were significant, while the model for out-closeness was marginally significant.

In the case of betweenness centrality, the multiple regression model explained 32% of the variability in the students scores of betweenness centrality. The backwards-stepwise regression analysis selection using the (AIC) criterion resulted in a regression model consisting of the affective and interactive categories of social presence, and both variables were found to be statistically significant predictors of betweenness centrality. In terms of their relative importance, the interactive category had a slightly larger standardized β coefficient than the affective category of social presence, indicating a slightly larger effect on the students' betweenness centrality scores.

With respect to degree centrality, the regression models explained 86% and 83% of the variability in the measures of in-degree and out-degree centralities, respectively. All three predictors were positively associated with the degree centrality measures, and all three reached the statistical significance. In terms of their relative importance, in both models, the interactive category of social presence

Table 6: Regression results for selected centrality measures after stepwise model selection using AIC criterion.

	Betweenness			In-degree			Out-degree			In-closeness			Out-closeness		
	β	SE	p	β	SE	p	β	SE	p	β	SE	p	β	SE	p
Affective	**0.27**	**0.12**	**0.024**	**0.18**	**0.054**	**0.001**	**0.23**	**0.059**	**<0.001**						
Interactive	**0.38**	**0.12**	**0.002**	**0.65**	**0.064**	**<0.001**	**0.65**	**0.07**	**<0.001**	**0.27**	**0.11**	**0.015**	0.37	0.15	0.017
Cohesive				**0.2**	**0.061**	**0.001**	**0.14**	**0.066**	**0.041**				-0.23	0.15	0.137
$F(3, 77)$	**19.6**		**<0.001**	**159**		**<0.001**	**130**		**<0.001**	**6.24**		**0.015**	3.03		0.054
Adjusted R^2	**0.32**			**0.86**			**0.83**			**0.061**			0.048		

had the largest standardized β coefficient, while the affective and cohesive categories had roughly the same standardized coefficients.

Regarding the two closeness centrality measures, the regression model for in-closeness was statistically significant, explaining 6.1% of the variability in the students' in-closeness centrality scores, while the model for out-closeness failed to reach the significance by a very small margin. The model for in-closeness consisted of only the interactive category, which was found to be a statistically significant predictor of in-closeness centrality. Similarly, the regression model for out-closeness consisted of the interactive and cohesive social presence categories, and explained 4.8% of the variation in the students' out-closeness centrality scores. In the model for out-closeness centrality, the only statistically significant predictor was the interactive category of social presence, while interestingly, the cohesive category of social presence was negatively associated with the change in the out-closeness centrality values, although statistically insignificantly.

5. DISCUSSION

One finding immediately stands out of the regression analyses results: *Interactive social presence is the most strongly associated with all of the network centrality measures, indicating a significant relation with the development of the students' social capital*. A possible explanation of this lies to some degree in the nature of students' social networks. Given that the primary goal of social networks in online courses is to serve as a communication medium for fostering of collaborative learning [27], it is reasonable to expect that interactivity in communication can explain a significant proportion of the differences in network positions, and ultimately the differences in the development of students' social capital. The reason why the interactive category is had the strongest association might be that only after the students have gott familiar with each other through focused, on-task interactions, and after they have started developing trust within a learning community, the expression of emotions and the sense of group belonging begins to emerge. This is aligned with the findings of Garrison [27] who suggested that interactive social presence is dominant at the beginning of a course, but decreases over time, while affective and cohesive social presence increase over time [27]. However, as Garrison [27] points out, too much of the interpersonal and affective interactions undermine the productivity of the collaborative learning activities. There is a certain amount of social interactions that is beneficial for learning [27], and the focus of the instructional interventions should be on: i) stimulating the right amount of the different social interactions that support productive and purposeful collaborative learning activities, and ii) the development of trust and the sense of community among the group of learners [17].

One practical implication of these results is that they suggest the effective way for fostering the productive social climate – and that is *focusing on the student interaction and open communication*. In order to guide the development of the social relationships in a learning community, it seems that the instructional emphasis should be on the interventions that require engaging in an open exchange of ideas and opinions, that would in turn lead to more affective expression, and eventually to the development of the sense of community belonging. Still, this hypothesis warrants further investigation, and in the future we plan to analyze the evolution of the students' social presence and the corresponding social network structures over time, which would shed new light on this important question.

The results of individual network centrality measures revealed that both in-degree and out-degree centrality measures were significantly predicted by all the three categories of students' social presence. By looking at the description (Section 2.2.2) and the indicators (Table 1) of the interactive category of social presence, we can see that interactive social presence is mainly about stimulating open and direct communication between the students. Thus, the students who exhibit a high level of interactive social presence have higher chances of "provoking" a response from the other students. Activities such as asking questions, explicitly referring to other students by name, quoting their messages, complementing them or agreeing with their messages, are all activities associated with an interactive and open communication, and can be used to elicit a response from the other students. It would be interesting to further investigate the relationship between different indicators of social presence and social capital, as certain indicators – such as `I4` "Asking questions" – seem to have more impact than the other indicators. Besides the interactive category, the regression model revealed that the affective and cohesive categories of social presence were also significant predictors of in-degree and out-degree centralities. These findings are even more interesting, as affective and cohesive exchanges are not directly stimulating discussions in the same manner as the interactive category. Further investigation is needed to examine particular time periods over the duration of a course in which those different dimensions of social presence contribute to the degree centrality measures of students.

With respect to betweenness centrality that is most closely related to the notion of social capital [13, 12], the regression model was statistically significant and explained 32% of the variability in the betweenness centrality scores. This corresponds to Cohen's $f^2 = 0.47$ effect size, which is considered to be a large effect size [16]. Both the interactive and affective categories of social presence were statistically significant predictors of the betweenness centrality, with the interactive category having a bit greater standardized β coefficient. This might be due to the nature of student communication networks and their focus on collaborative learning, which resulted in the emphasis on information exchange. Still, these are very intriguing findings, given that betweenness centrality is not directly related to the number of interactions the student has, but more to the overall diversity of the interactions within a group of learners. In a follow-up study, it would be very interesting to investigate whether there are any particular ways in which the students with the high betweenness centrality differ from the other students (e.g., asking many questions or exhibiting higher self-disclosure).

Regarding the closeness centrality measures, the regression model for in-closeness was also statistically significant. The model explained 6.1% of the variability, and the stepwise model selection

using the AIC criteria resulted in a simple regression model with only the interactive category of social presence. In contrast to degree centrality, which considers only direct relationships, closeness centrality also considers the indirect relationships. Such indirect relationships could be the reason why only interactive category was rendered as important. The affective and cohesive exchanges between students A and B, although very important, provide very little, or no influence on the indirect relations of student B and the rest of the students. The similar findings we could see in the model for out-closeness, which was marginally significant with the p-value of 0.054. However, it could be expected that the significance of this model would be conformed in a larger replication study.

The major limitations of this study is the sample size and the use of the single course from a single institution. Even though there were six offerings of the course taught by the two instructors, there might still be significant effects of the adopted pedagogical approach, which could have shaped a specific social dynamics, and thus, potentially distort the findings of our study. Likewise, we considered all interactions among the students as contributing to their social capital, it is very likely that the certain interactions (e.g., adversarial interactions) might have a negative effect on the student social capital. In the future work, we plan on replicating our findings on a bigger sample and with more diverse courses from different subject matter domains. Finally, we plan to investigate the temporal aspects of the relationship between social capital and the social presence, which might give us a deeper insight into the complexity of the social interactions in learning communities.

6. CONCLUSIONS

The study presented in this paper investigated some of the social processes that can contribute to the development of students' social capital. We have looked at the relationship between students' *social presence*, operationalized through the Community of Inquiry model, and students' *social capital*, operationalized through the three network centrality measures. The implications of our findings are twofold: *First*, our results indicate that a significant part of the variability in network centrality scores can be explained using the three dimensions of the social presence, and this in turn indicates the existence of the relationship between the development of social presence and social capital. All three categories of social presence were significant predictors of in-degree and out-degree centrality measures while interactive and affective categories were significant predictors of the betweenness centrality. Also, interactive category of social presence was significantly predictive of the in-closeness and out-closeness centrality measures, although the overall regression model for out-closeness was marginally significant. A possible explanation is that given the task-oriented nature of discussions in online courses, students' social presence develops mostly through interactions focused on learning, and then over time, with the development of trust among a group of learners, the other dimensions of social presence start to emerge. *Second*, the study shows the significant relationship between the interactive category of social presence and betweenness, in-degree, out-degree, and in-closeness network centrality measures. This provides an empirical basis for fostering the productive social climate in discussions through interventions that increase interactivity and open communication among the students. By engaging students to participate in discussions with the clearly defined expectations, students develop social relationships which can in turn have positive impact on the attainment of the learning objectives and their overall academic experience.

References

[1] Special issue on the community of inquiry framework: Ten years later. *The Internet and Higher Education*, 13(1–2), 2010.

[2] T. Anderson and J. Dron. Three generations of distance education pedagogy. *The International Review of Research in Open and Distance Learning*, 12(3):80–97, 2010.

[3] T. Anderson, L. Rourke, D. R. Garrison, and W. Archer. Assessing teaching presence in a computer conferencing context. *Journal of Asynchronous Learning Networks*, 5:1–17, 2001.

[4] J. Arbaugh, M. Cleveland-Innes, S. R. Diaz, D. R. Garrison, P. Ice, J. C. Richardson, and K. P. Swan. Developing a community of inquiry instrument: Testing a measure of the community of inquiry framework using a multi-institutional sample. *The Internet and Higher Education*, 11(3–4):133–136, 2008.

[5] A. W. Astin. *What Matters in College: Four Critical Years Revisited*. Jossey-Bass, 1 edition edition, 1997.

[6] R. S. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1(1):3–17, 2009.

[7] T. T. Baldwin, M. D. Bedell, and J. L. Johnson. The social fabric of a team-based M.B.A. program: Network effects on student satisfaction and performance. *The Academy of Management Journal*, 40(6):1369–1397, 1997.

[8] J. P. Bean. Conceptual models of student attrition: How theory can help the institutional researcher. *New Directions for Institutional Research*, 1982(36):17–33, 1982.

[9] R. A. Bjork, J. Dunlosky, and N. Kornell. Self-regulated learning: beliefs, techniques, and illusions. *Annual review of psychology*, 64:417–444, 2013.

[10] B. L. Bowerman and R. T. O'Connell. *Linear Statistical Models: An Applied Approach*. Duxbury Press, 1990.

[11] S. Buckingham Shum and R. Ferguson. Social learning analytics. *Journal of Educational Technology & Society*, 15(3):3–26, 2012.

[12] R. S. Burt. Structural holes versus network closure as social capital. In N. Lin, K. Cook, and R. S. Burt, editors, *Social Capital: Theory and Research*. Aldine Transaction, 2001.

[13] R. S. Burt. The social capital of structural holes. In M. F. Guillen, R. Collins, P. England, and M. Meyer, editors, *The New Economic Sociology: Developments In An Emerging Field*. Russell Sage Foundation, 2005.

[14] B. V. Carolan. *Social Network Analysis and Education: Theory, Methods and Applications*. SAGE Publications, Inc., 2014.

[15] H. Cho, G. Gay, B. Davidson, and A. Ingraffea. Social networks, communication styles, and learning performance in a CSCL community. *Computers & Education*, 49(2):309–329, 2007.

[16] J. Cohen. The analysis of variance. In *Statistical power analysis for the behavioral sciences*, pages 273—406. L. Erlbaum Associates, Hillsdale, N.J., 1988.

[17] S. Dawson. A study of the relationship between student social networks and sense of community. *Journal of Educational Technology & Society*, 11(3):224–238, 2008.

[18] M. F. De Laat, V. Lally, L. Lipponen, and R.-J. Simons. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: A role for social network analysis. *International Journal of Computer-Supported Collaborative Learning*, 2(1):87–103, 2007.

[19] B. De Wever, T. Schellens, M. Valcke, and H. Van Keer. Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education*, 46(1):6–28, 2006.

[20] J. Dewey. My pedagogical creed. *School Journal*, 54(3):77–80, 1897.

[21] R. Donnelly and J. Gardner. Content analysis of computer conferencing transcripts. *Interactive Learning Environments*, 19(4):303–315, 2011.

[22] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[23] J. E. Eckles and E. G. Stradley. A social network analysis of student retention using archival data. *Social Psychology of Education*, 15(2):165–180, 2011.

[24] S. L. Feld. The focused organization of social ties. *American Journal of Sociology*, 86(5):1015–1035, 1981.

[25] R. Ferguson and S. B. Shum. Social learning analytics: five approaches. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, LAK '12, page 23–33, New York, NY, USA, 2012. ACM.

[26] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.

[27] D. R. Garrison. *E-Learning in the 21st Century: A Framework for Research and Practice*. Routledge, New York, 2 edition edition, 2011.

[28] D. R. Garrison, T. Anderson, and W. Archer. Critical inquiry in a text-based environment: Computer conferencing in higher education. *The Internet and Higher Education*, 2(2–3):87–105, 1999.

[29] D. R. Garrison, T. Anderson, and W. Archer. The first decade of the community of inquiry framework: A retrospective. *The Internet and Higher Education*, 13(1–2):5–9, 2010.

[30] D. R. Garrison and J. Arbaugh. Researching the community of inquiry framework: Review, issues, and future directions. *The Internet and Higher Education*, 10(3):157–172, 2007.

[31] R. Garrison, M. Cleveland-Innes, and T. S. Fung. Exploring causal relationships among teaching, cognitive and social presence: Student perceptions of the community of inquiry framework. *The Internet and Higher Education*, 13(1–2):31–36, 2010.

[32] D. Gasevic, A. Olusola, S. Joksimovic, and V. Kovanovic. Externally-facilitated regulation scaffolding and role assignment to develop cognitive presence in asynchronous online discussions. *The Internet and Higher Education*, (submitted), 2014.

[33] D. Gasevic, A. Zouaq, and R. Janzen. "Choose your classmates, your GPA is at stake!": The association of cross-class social ties and academic performance. *American Behavioral Scientist*, 2013.

[34] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[35] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, New York, NY, 2013.

[36] C. Haythornthwaite. Social network analysis: An approach and technique for the study of information exchange. *Library & Information Science Research*, 18(4):323–342, 1996.

[37] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[38] K. H. Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, 2003.

[39] N. Lin. Social resources and instrumental action. In P. V. Marsden and N. Lin, editors, *Social structure and network analysis*, pages 131—145. Sage Publications, 1982.

[40] P. Long and G. Siemens. Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5):31–40, 2011.

[41] R. Luppicini. Review of computer mediated communication research for education. *Instructional Science*, 35(2):141–185, 2007.

[42] A. Mehrabian. Some referents and measures of nonverbal behavior. *Behavior Research Methods & Instrumentation*, 1(6):203–207, 1968.

[43] T. Rizzuto, J. LeDoux, and J. Hatala. It's not just what you know, it's who you know: Testing a model of the relative importance of social networks to academic performance. *Social Psychology of Education*, 12(2):175–189, 2009.

[44] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *Trans. Sys. Man Cyber Part C*, 40(6):601–618, 2010.

[45] L. Rourke, T. Anderson, D. R. Garrison, and W. Archer. Assessing social presence in asynchronous text-based computer conferencing. *The Journal of Distance Education*, 14(2):50–71, 1999.

[46] J. Scott and P. J. Carrington. *The SAGE Handbook of Social Network Analysis*. SAGE Publications, 2011.

[47] P. Shea, S. Hayes, S. U. Smith, J. Vickers, T. Bidjerano, M. Gozza-Cohen, S.-B. Jian, A. Pickett, J. Wilde, and C.-H. Tseng. Online learner self-regulation: Learning presence viewed through quantitative content- and social network analysis. *The International Review of Research in Open and Distance Learning*, 14(3):427–461, 2013.

[48] P. Shea, S. Hayes, J. Vickers, M. Gozza-Cohen, S. Uzuner, R. Mehta, A. Valchova, and P. Rangan. A re-examination of the community of inquiry framework: Social network and content analysis. *The Internet and Higher Education*, 13(1–2):10–21, 2010.

[49] R. A. Smith and B. L. Peterson. "Psst ... what do you think?" the relationship between advice prestige, type of advice, and academic performance. *Communication Education*, 56(3):278–291, 2007.

[50] S. L. Thomas. Ties that bind: A social network approach to understanding student integration and persistence. *The Journal of Higher Education*, 71(5):591–615, 2000.

[51] V. Tinto. *Leaving College: Rethinking the Causes and Cures of Student Attrition*. University of Chicago Press, 1993.

[52] S. Wasserman. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[53] Y. Woo and T. C. Reeves. Meaningful interaction in web-based learning: A social constructivist interpretation. *The Internet and Higher Education*, 10(1):15–25, 2007.

Cross-Domain Performance of Automatic Tutor Modeling Algorithms

Rohit Kumar
Raytheon BBN Technologies
Cambridge, MA, USA
rkumar @ bbn.com

ABSTRACT

In our recent work, we have proposed the use of multiple solution demonstrations of a learning task to automatically generate a tutor model. We have developed a number of algorithms for this automation. This paper describes the application of these domain-independent algorithms to three datasets from different learning domains (Mathematics, Physics, French). Besides verifying the applicability of our approach across domains, we report several domain specific performance characteristics of these algorithms which can be used to choose appropriate algorithms in a principled manner. While the Heuristic Alignment based algorithm (*Algorithm 2*) may be the default choice for automatic tutor modeling, our empirical finding suggest that the Path Pruning based algorithm (*Algorithm 4*) may be favored for language learning domains.

Keywords

Tutor Modeling, Automation, Domain Independence, STEM domains, Language Learning

1. INTRODUCTION

Wide-scale transition of Intelligent Tutoring Systems (ITS) to the real world demands a scalable ability to develop such systems. The past decade has seen the first instantiations of industrialization of ITS development in the form of commercial products for different learning domains as well as diverse user populations. In addition to addressing non-technical challenges such as designing robust production processes around multidisciplinary teams of domain and pedagogical experts [1], the industrialization of this technology is enabled by technical advancements such as the development of general purpose authoring tools [2] which has allowed a scalable workforce to contribute to ITS development.

In this paper, we extend our recent work [3][4] on automatically developing Example-Tracing Tutors (ETTs) [5] using multiple behavior demonstrations. Conventionally, ETTs are developed in three stages by trained domain experts: (1) User Interface (UI) development, (2) Behavior demonstration, (3) Generalization and annotation of the behavior graph. As ITS are being deployed to a large active user pool, it is now possible to pilot the UI with a small sample of learners to collect multiple behavior demonstrations. We can significantly reduce the Stage 3 effort of ITS developers by using algorithms that can automatically create a generalized behavior graph from multiple demonstrations. Several algorithms to address this challenge have been proposed and evaluated [4].

In this paper, we will study the applicability and performance of these algorithms on publicly available datasets from three different learning domains. Section 3 summarizes the key characteristics of the four algorithms used in our study. Section 4 describes learning domains and the corresponding datasets used in this work. Results and Analysis from our experiments are presented in Section 5. Before diving into the algorithms, the next section reviews related work on automation of tutor model development.

2. RELATED WORK

Automation of tutor model development process has been explored in different contexts using completely automated methods as well as augmentation of authoring tools [6][7]. For example, motivated by application in language learning, a series of workshops on the problem of automatic question generation [8] explored a number of information extraction and NLP techniques that employ existing linguistic resources. Barnes and Stamper [9] proposed a method that uses existing student solutions to generate hint messages for the Logic Proof tutor. Recently, Eagle et al. [10] have used clustering of interaction network states as an approach to the same problem.

In the context of knowledge-tracing and example-tracing tutors, McLaren et al. [11] proposed the use of activity logs from novice users to bootstrap tutor model development. They developed software tools that integrate access to novice activity logs with authoring tools. The baseline algorithm (Interaction Networks) used in our work is similar to the integrated data view used in this prior work. Furthermore, the algorithms used in our work address some of the shortcomings of their work (e.g. inability to identify "buggy" paths).

In addition to tutor modeling, recent work has investigated automated methods for improving domain and student models [12] [13]. Sudol et al. [14] aggregated solution paths taken by different learners to develop a probabilistic solution assessment metric. Johnson et al. [15] are creating visualization tools for interaction networks that combine learner traces from open-ended problem solving environments. They have developed an algorithm for reducing the complexity of combined networks to make them more readable/navigable. In a similar spirit, work by Ritter et al. [16] used clustering techniques to reduce the large feature space of student models to assist in qualitative model interpretation.

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

3. GENERATING BEHAVIOR GRAPHS

Automatic Behavior Graph Generation (ABGG) algorithms analyze the similarities and difference between multiple solution demonstrations of a problem to induce a behavior graph that can serve as a tutor model for the problem.

3.1 Behavior Graphs

Behavior graphs [5] are directed graphs. The nodes in this graph correspond to valid solution states. Non-terminal nodes represent partial solutions. Edges in the graph represent solution paths some of which are correct and lead to the next state while other are incorrect and usually lead back to the same state. Edges are annotated with the conditions that a behavior event must meet to traverse the path.

Behavior graphs may contain multiple paths between two nodes. Multiple paths are useful to facilitate learner's exploration of alternate solutions to a problem especially in ill-defined learning domains. Behavior graphs may also include unordered groups. As the name suggests, states within an unordered group may be traversed in any order.

Well-constructed behavior graphs have several desirable characteristics which motivate the design of metrics we use to evaluate ABGG algorithms.

3.1.1 Effective

Since the purpose of the behavior graphs is to serve as a tutor model, the primary metric for evaluating these models is their learning efficacy measured via use of the models by a relevant sample of learners. However, in this paper we focus only on the use of automated metrics that do not require access to a learner pool. Further, as we in section 5, the automatically generated behavior graphs are not perfect. They require checking and refinement by ITS developers before they can be used with learners.

3.1.2 Readable

One of the key characteristics of behavior graphs that makes them a popular model is that they are readable by ITS developers without requiring a deep understanding of computational or cognitive sciences. Automatically created behavior graphs should be editable with existing authoring tools to facilitate necessary manual annotation and modifications. Ideally, ABGG algorithms should create concise graphs without losing other desirable characteristics. This may involve collapsing redundant paths and even pruning spurious or infrequent edges.

The conciseness of a graph can be measured using the number of nodes and edges in the graph. Our primary readability metric, *Compression Ratio* measures the rate at which an algorithm is able to reduce behavior events into behavior states (i.e. nodes) by finding similarities between events.

3.1.3 Complete

In order to minimize author effort, generated behaviors graphs should be as complete for creating an ETT as possible. As a minimal criterion, at least one valid path to the final solution should be included♦. Additionally, complete behaviors graphs are annotated with all the expected inputs by the learner. We use the *Rate of Unseen Events* in held out demonstrations as the primary metric to measure the completeness of our automatically generated behavior graphs.

3.1.4 Accurate

Behavior graphs should be error free. This includes being able to accurately capture the correct and incorrect events by learners depending on the current solution state. Edge accuracy measures the percentage of Correct & Incorrect edges that were accurately generated by the algorithm. *Error Rate* is a frequency weighted combination of edge accuracy that measures the fraction of learner events that will be inaccurately classified by the automatically generated behavior graph. We use the error rate of an automatically generate behavior graph on held out demonstrations as the primary accuracy metric.

3.1.5 Robust

One of the reasons for the success of expertly crafted ETTs is the ability to use them with a wide range of learners under different deployment conditions. Automatically generated behavior graphs should retain this characteristic; e.g., by identifying alternate paths and unordered groups. It is not unforeseeable that the use of a data-driven approach could contribute to creating behavior graphs that are more robust than those authored by a human expert.

Branching factor is the average number of data values available at each UI element. A large branching factor indicates the capability to process a large variety of learner inputs at each state. Also, the number and size of unordered groups is indicative of flexibility a graph affords to learners to explore the solution paths of a problem.

Note that readability and robustness are complementary characteristics of a behavior graph. For example, a highly complex behavior graph may be very robust but may not be very readable.

3.2 ABGG Algorithms

We use four algorithms, introduced in our previous work [4], to generate behavior graphs using multiple solution traces of a problem. The first algorithm (*Algorithm 1*) generates interaction networks by sequentially collapsing identical events in solution traces into a shared node and creating a branch whenever two different events are found. Interaction networks have been used in prior work [10][15].

Algorithm 2 uses a heuristic alignment technique [3] to align similar events across multiple solution traces. The alignment is used to obtain a sequence of traversal through the problem's steps. Furthermore, this algorithm is able to use the positional entropy of a sequence of elements while obtaining the optimal sequence to identify unordered groups.

Similar to the above algorithm, *Algorithm 3* finds the optimal sequence between aligned events. However, this algorithm uses the Center Star Algorithm [17] to align the multiple solution traces instead of the heuristic used by *Algorithm 2*. The Center Star Algorithm is a foundational algorithm used for aligning more than two sequences of symbols. It is particularly suited for our application because it is polynomial time in computational complexity and it does not make any assumptions about the space and relationship of symbols comprising the sequence.

First order transition matrix computed from solution traces can be used to represent a directed graph. *Algorithm 4* considers ABGG as the process of finding multiple paths in a directed graph. Specifically, the longest (non-repeating) path in this directed graph represents the most likely path through the solution steps. Since, the problem of finding longest paths in general graphs is known to be NP-hard, we employ a combination of bounded

longest path finding and an algorithm for finding multiple shortest paths [18] in a transformed transition matrix to obtain a number of different paths through the directed graph. These paths are merged to construct a behavior graph similar to the process of constructing an interaction network.

Algorithm 2, 3 and 4 assume that if two or more events within a trace were generated by the same UI element, the latter event corresponds to a correction of the data value input at the former events. In this case, we refer to the former events as *retracted events* and data values entered at these events are assumed to be incorrect values. Using this assumption, these three algorithms are able to automatically generate incorrect paths in behavior graphs unlike *Algorithm 1*. This assumption is not applied to *Algorithm 1* to compare our work against prior work [11] on extracting tutor models from multiple demonstrations.

3.3 Discussion

Table 1 characterizes the four algorithms described above based on their capabilities. Incremental addition of demonstrations to generate interaction networks does not identify incorrect input data values. However, using the assumption about retracted events, the other three algorithms are able to identify incorrect inputs. Johnson et al. [15] used a similar assumption in their work on reducing the visual complexity of interaction networks. We notice that the *Algorithms 2 and 3* are complementary in terms of their ability to find alternate paths and unordered groups. *Algorithm 4* on the other hand offers both of these abilities.

Table 1. Comparison of Algorithm Capabilities

Capability▼ Algorithm►	1	2	3	4
Identifies incorrect answers	N	Y	Y	Y
Generates alternate paths	N	N	Y	Y
Finds unordered groups	N	Y	N	Y
Generalizes beyond training demonstrations	N	Y	Y	Y
Guarantees all training demnstrs. will pass	Y	N	N	N
Finds atleast one path to final solution♦	Y	Y	Y	N
Discovers new/unseen data values	N	N	N	N

None of the algorithms discussed in this paper are capable of discovering unseen inputs beyond those seen in the solution traces. This type of generative ability is particularly useful for learning tasks, such as language learning, where a large number of different inputs may be expected from the learners. In our ongoing work, we use a number of heuristics [7] as well as grammar induction techniques [6] to generate unseen inputs for certain nodes in the behavior graphs.

4. DATASETS

We use three datasets, accessed via DataShop[1] [19], to study the cross-domain applicability of ABGG algorithms. These datasets were filtered to use only problems that had six or more traces and had at least two UI elements. Also, we eliminated all events, such as help requests, that did not correspond to user input at a solution step. In this way, the datasets were transformed into solution traces. As discussed in Kumar et al. [4], a solution trace/demonstration comprises of a sequence of user interface (UI) events. Each event is represented as a 2-tuple $e = (u, d)$ that includes an identifier u of the UI element and data d associated with the event. A UI element may be visited any number of times within a trace. In general, data can include one or more attributes of the event such as the event type, user input, event duration, etc. In this paper, we assume single data attribute events where the data captures the learner input at the UI element.

Table 2. Problems & Traces for the three learning domains

	Math.	Physics	French
#Problems	1013	497	71
Max. #Unique Elements	33	62	10
Avg. #Unique Elements	4.6	9.7	2.5
Avg. #Training Traces	76.0	26.6	12.1
Avg. #Heldout Traces	38.0	13.3	6.1
Avg. #Events Per Trace	5.3	8.9	4.7

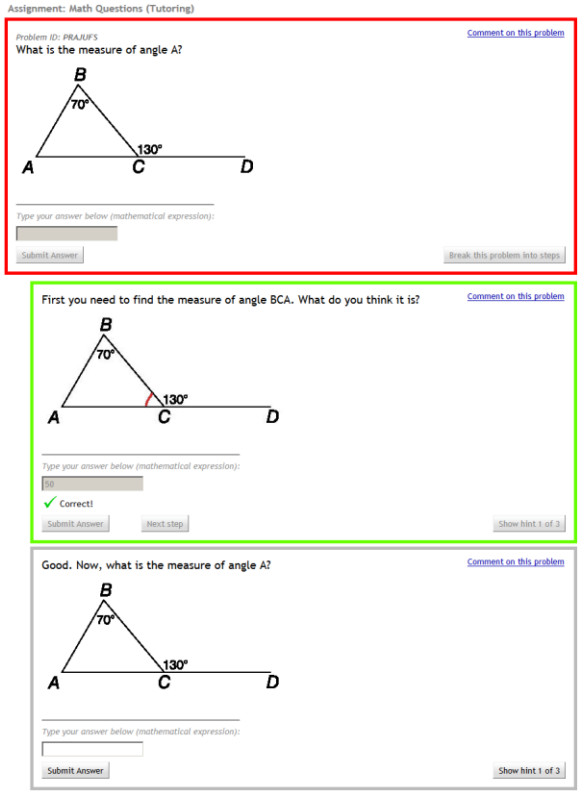


Figure 1. Example Math Problem from *Assistments*
Source: www.assistments.org, April 2014

Table 2 provides some statistics about the problem and traces for each of learning domains used in this work. The Mathematics traces were derived from three *Assistments* [20] datasets. *Assistments* is a web-based learning platform, developed by Worcester Polytechnic Institute (WPI), that includes a Mathematics intelligent tutoring system for middle & high school grades. Figure 1 shows an example math problem from the *Assistments* system. Together, these datasets are the largest of the three domains we use. Prior to filtering, these dataset comprised a total of 683,197 traces and 1,905,672 events from 3,140 problems. For our experiments, we treat the three datasets to be independent

[1] PSLC DataShop is available at http://pslcdatashop.org

of each other to account for change in UI designs of the problems common to the three datasets.

We used 10 (out of 20) of the largest datasets released under the *Andes2* project [22] to build the collection of Physics problems and traces. *Andes2* is an intelligent tutoring system that includes pedagogical content for a two-semester long college and advanced high-school level Physics course. These ten datasets are based on logs from several semesters of use of the *Andes2* system at the United States Naval Academy. Prior to filtering, these dataset comprised a total of 81,173 traces and 1,162,581 events from 2,187 different problems. Note that, as is case with the Math dataset, we treat the ten *Andes2* datasets independently. Note that, unlike typical domain independent example-tracing based tutor, the *Andes2* systems uses a model-tracing approach for tracking learner's solution of a problem and to provide feedback. The domain knowledge dependent model tracer is able to match highly inflected learner inputs (e.g. variable names) to its solution graph. Despite this difference in tutoring approach used by the *Andes2* system, we decided to include this domain in our experiments to study the performance of our algorithms on such solution traces.

Finally, the French traces are based on two dataset from the "French Course" project on DataShop. These datasets were collected from logs of student's use of the "French Online" course hosted by the Open Learning Initiative (OLI) [22] at Carnegie Mellon University. Figure 2 shows steps from couple of example problems from this course. These datasets comprised a total of 37,439 traces and 253,744 events from 1,246 different problems. Note that a significantly larger fraction of French problems were eliminated due to the filtering criterion compared to Mathematics or Physics.

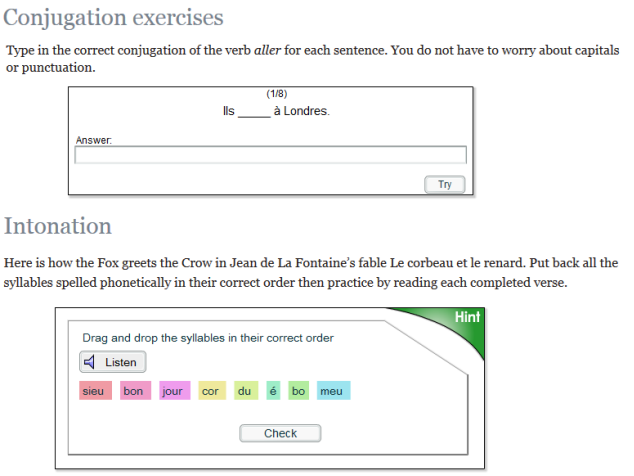


Figure 2. Example Steps from Problem from the French Online Course *Source: oli.cmu.edu, April 2014*

The datasets used in our experiments contain solution traces. Traces are paths through an existing behavior graph, unlike behavior demonstrations which are unconstrained by existing tutor models. In addition to the fact that these are the only available large scale collection of solution paths, we use these datasets in our experiments because these traces have been

Table 3. Averaged Metrics for the Graphs Generated by ABGG Algorithms
*indicates significant ($p < 0.05$) difference with the other algorithms (within the same dataset)

	Mathematics (*Assistments*)				**Physics (*Andes2*)**				**French (OLI)**			
Algorithm ►	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
#Nodes	79.2	**5.4***	6.0*	6.6*	147.8	**7.9***	11.5*	11.7*	25.6	**3.8***	4.5*	4.5*
#Correct Edges	148.0	**12.9***	18.3*	17.5*	182.2	43.5*	76.4	**34.5***	37.2	**6.9**	9.8	9.5
#Incorrect Edges		23.9	33.5	**19.5***		35.1	53.0	**13.4***		**4.2**	11.0	8.0
Compression Ratio	6.7	**76.8***	66.8	60.2	2.3	**31.6***	21.9	21.7	2.2	**14.6**	12.8	12.8
% Accurate Correct Edges	39.1	41.9	42.5*	**44.1***	61.4	80.2*	58.9	**80.8***	22.5	27.7*	26.9*	**29.8***
% Accurate Incorrect Edges		**99.9***	97.2	99.5*		**92.5***	67.3	85.5		**97.8***	86.1	87.2
Training Error Rate	51.4	25.4	17.7*	**17.5***	33.6	**17.2***	25.8	24.3	75.2	56.1	**22.3***	25.3*
Heldout Error Rate	42.8	23.5	16.1*	**15.7***	29.1	**25.5***	33.3	30.8	45.3	35.9	19.9*	**18.5***
% Training Unseen Events	**0.0***	10.7	2.2	6.8	**0.0***	14.1	12.2	24.6	**0.0***	13.4	5.2	4.5
% Heldout Unseen Events	**10.2***	19.1	11.5*	13.9	**35.9***	41.7	38.4*	42.6	**31.7***	40.7	34.4*	34.3*
Branching Factor	2.2	10.9	**12.6***	8.5	1.5	**13.4***	12.9*	6.0	1.6	6.7*	**9.4***	7.8*
#Groups		**0.5***		0.0		0.8		**1.4***		**0.3***		0.1
Avg. Group Size		**1.9***		0.0		**2.0**		**2.0**		**0.6***		0.3
% Group Coverage		**31.8***		0.5		27.2		**30.6***		**15.4***		6.1

collected from a large set of real users. They contain realistic variations in learner inputs similar to demonstrations.

5. EXPERIMENTS

We use a three-fold cross validation design that splits the available traces into three different training and held out sets. The readability metrics (i.e. number of nodes, number of edges and compression ratio) as well as the robustness metrics (branching factor, number of unordered groups, average group size and coverage of graph within groups) are reported on the behavior graphs generated by the algorithms. On the other hand, some accuracy metrics such as the accuracy of correct and incorrect edges are measured on generated graphs whereas others such as error rate are measured on event sequences which could be the training traces; i.e., sequences used to generate the graphs, or held out traces. Similarly, our completeness metrics, i.e. the rate of unseen events in a sequence, can be measured on both training as well as held out traces. Note that the metrics computed on training traces used to generate the graphs may not accurately indicate the performance of an algorithm due to over-fitting. This is the motivation for choosing the cross validation based experimental design.

5.1 Results

Table 3 shows our results along 14 metrics for each of the four algorithms applied to the three learning domains under consideration. Reported metrics are averaged over three cross validation splits as well as over all the problems for each domain. The metrics are organized by the four desirable characteristics discussed earlier. Primary metric for each characteristic is highlighted.

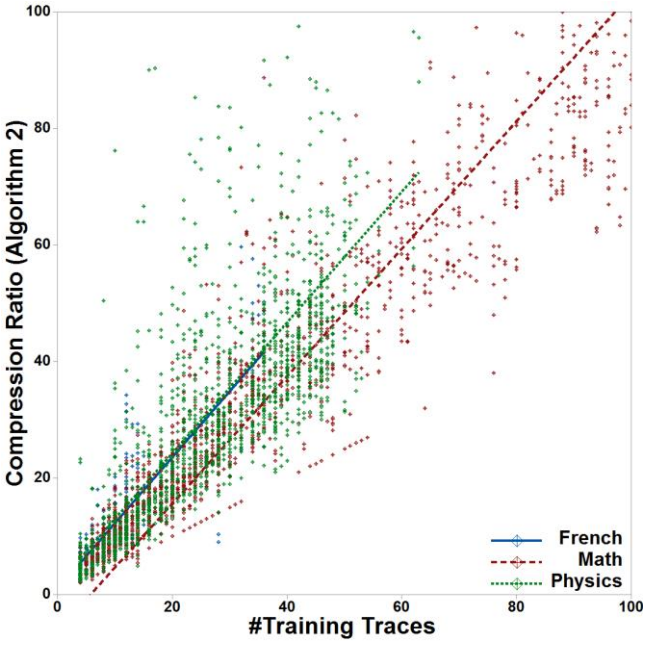


Figure 3. Compression Ratio of *Algorithm 2*

5.1.1 Mathematics

As expected, the interaction networks comprise of a large number of nodes and edges that lead them to have significantly smaller compression ratio. *Algorithm 2* (Heuristic Alignment) outperforms all other algorithms on three of the readability metrics. On the other hand, *Algorithm 4* (Path Pruning) significantly outperforms the other algorithms on three of the accuracy metrics for this dataset and is not significantly worse on the fourth metric. Because of their lossless nature, *Algorithm 1* (Interaction Network) performs the best on Completeness metrics (% unseen events). However, it is not significantly better than *Algorithm 3* (Center-Star Alignment). We find evidence of over-fitting of the algorithms to training traces on this metric as indicated by the approximately 9% higher rate of unseen events for held out traces for all the algorithms. *Algorithm 3* significantly outperforms the other algorithms on the primary robustness metric (Branching Factor) for this domain. *Algorithm 2* is better than *Algorithm 4* for the metrics based on unordered groups.

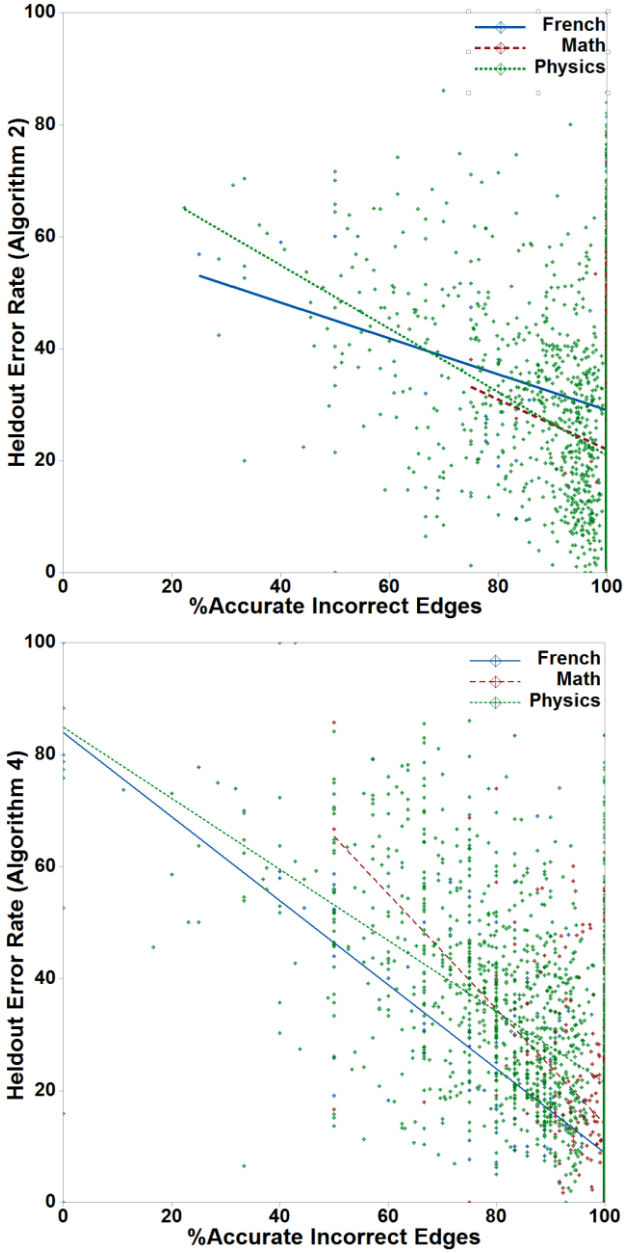


Figure 4. Heldout Error Rate of *Algorithms 2 and 4*

5.1.2 Physics

On the primary readability metric (Compression Ratio), *Algorithm 2* outperforms the others on the Physics dataset as was the case with Mathematics. This is consistent with prior conclusion [4] on the use of *Algorithm 2* for readability. We note that the Physics

dataset has significantly lower compression ratio than the previous dataset. Figure 3 shows a scatter plot and domain-specific regression fits for the compression ratio of *Algorithm 2* for different problems with different number of training traces and UI elements. We see that for equivalent number of training traces, the compression ratio for Physics is actually slightly better than Mathematics. However, as we know from Table 2, fewer training traces are available for the Physics problems on average.

On the primary accuracy metric, we find that *Algorithm 2* works best for Physics unlike the case with the Mathematics domain. We can note that the *Algorithm 2* is significantly better on the accuracy of incorrect edges. Figure 4 shows the relationship between the error rate in heldout traces and the accuracy of incorrect edges. We also see that the percentage of unseen events in heldout traces is significantly higher for Physics. The lower incorrect edge accuracy and higher percentage of unseen events can be attributed to the differences in the tutoring approach underlying the *Andes2* system which uses domain-specific knowledge to match a large variety of inputs from the learner at each step of the solution. Because of this, *Andes2* elicits significantly diverse (& hence novel) inputs across traces. *Algorithms 2 and 3* are not significantly different in terms of the primary robustness metric.

5.1.3 French

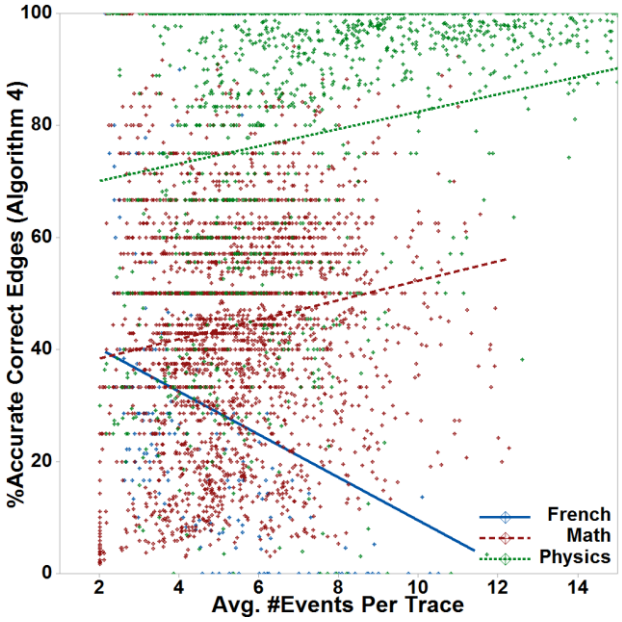


Figure 5. Accuracy of Correct Edges for *Algorithm 4*

The results for our non-STEM domain are largely consistent with the Mathematics domain. This may be attributed to the similarities of the underlying tutoring approach for the *Assistments* system and the French Online course which has been developed using the Cognitive Tutor Authoring Tools (CTAT) [2]. However, we can notice two key differences. First, the accuracy of correct edges for this domain is significantly lower. Because the French Online Course is deployed on an publicly accessible platform, its likely that a large number of the solution traces were generated by beginners as well as non-serious users leading to the dataset containing many incomplete solution traces containing no correct answers. This is evidenced in Figure 5 as we see that correct edge accuracy dramatically degrades for long traces which is contrary to the case with the other two domains.

Second, we expect the branching factor to be higher for a language learning domain, due to the high degree of linguistic variation in learner inputs. The results in Table 3 do not indicate this. However, Figure 6 verifies this intuition. Branching factor for the French behavior graphs is higher than those for the STEM domain for problems that have 10 or more traces.

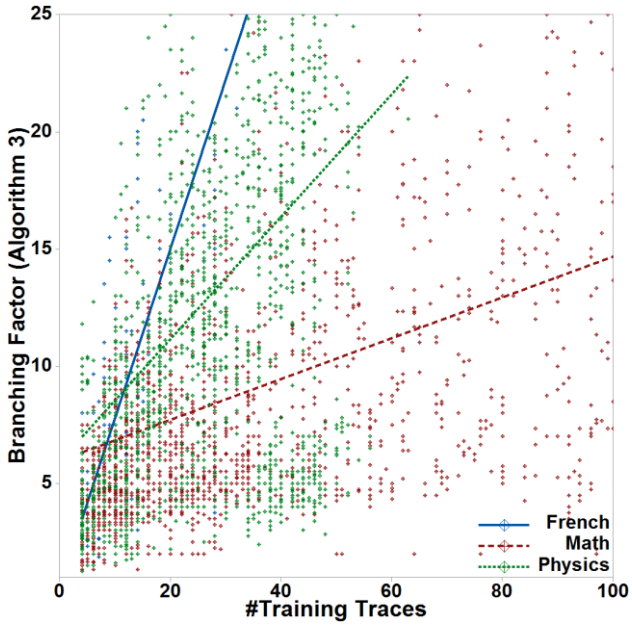


Figure 6. Branching Factor of *Algorithm 3*

5.1.4 Automatically Generated Behavior Graphs

Figures 7, 8 and 9 showcase several qualitative characteristics of automatically generated behavior graphs (truncated to fit) for the problems in the three datasets used in this work. We use the following visual convention: Circular nodes represent states and are labeled with identifiers *u* of the corresponding UI element. Edges are labeled with the data values *d*. Correct edges are labeled with green rectangles and incorrect edges are labeled with red rectangles. Unordered groups are shown using blue containers.

Figure 7 shows graphs generated by two different algorithms for the same Mathematics problem in the *Assistments* dataset using 241 solution traces by learners. The graph generated by *Algorithm 1* is dense and hardly readable due to the large number of nodes and edges in this graph. Also, as discussed in Section 3, this algorithm is unable to identify incorrect paths. Contrary to that, the graph in Figure 7b is composed of only 6 nodes. The various paths taken by learners are compressed into 46 correct and 39 incorrect edges. We can notice that not all paths are accurate. However, the accurate paths are more frequent, as indicated by the thicker arcs associated with the edge. In our ongoing work, we are extending these algorithms to use this frequency attribute to eliminate inaccurate paths (either automatically, or by providing additional controls to model developers in authoring tools).

A behavior graph from the Physics dataset is shown in Figure 8. As discussed earlier, the large variation in learner input at each state is depicted in the edge labels of this graph. We notice that for the last state (s6) which corresponds to the learners filling in the answer to a problem, many minor variations of the correct answer are accurately captured. Due to the domain independent nature of our algorithms, these answers are treated as different string. Integration of domain knowledge can lead to further compression of these answers into a single path.

The linguistic variation in the inputs to a problem in the French dataset is also noticeable in the two graphs for the same problem in Figure 9. We can see the several wrong answers are marked as correct answers (and vice versa), although the frequency-based edge notation identifies the correct answer as was the case in Figure 7b. In this problem, learners are asked to listen to an audio file and type in the French word they hear. Learners are allowed to go back and forth between these two steps. The first step has no wrong answer. We notice that our assumption to consider retracted events as incorrect fails in this case.

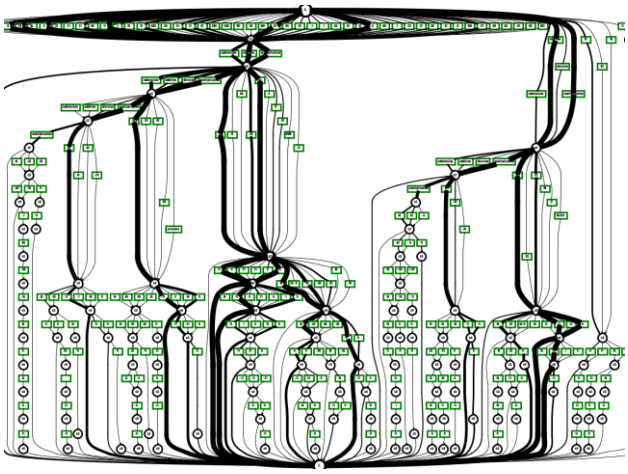


Figure 7a. Behavior Graph: Mathematics, *Algorithm 1*

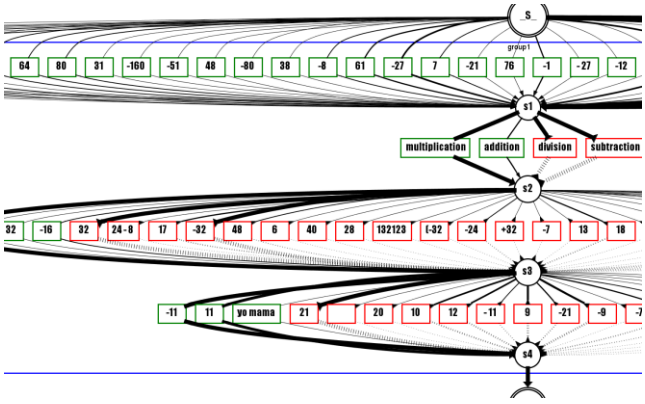


Figure 7b. Behavior Graph: Mathematics, *Algorithm 2*

It is particularly interesting to note the differences in the way *Algorithm 2* and *Algorithm 4* encode robustness into the learnt tutor model. While *Algorithm 2* identifies an unordered group containing the *listen* and *answer* nodes which allows learners to traverse these nodes in any order, *Algorithm 4* identifies that the *listen* step is optional and create two different way to reach the *answer* step based on the solution behaviors exhibited by learners in the traces.

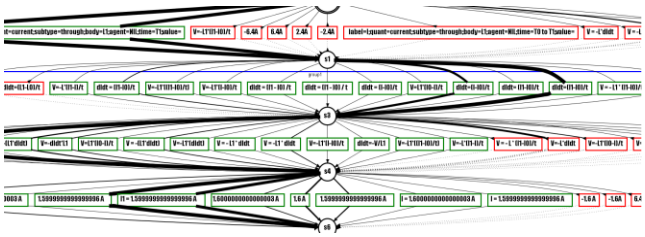


Figure 8. Behavior Graph: Physics, *Algorithm 2*

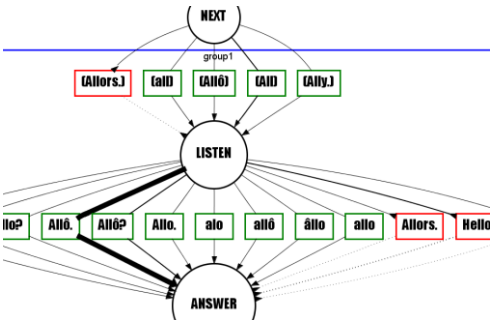


Figure 9a. Behavior Graph: French, *Algorithm 2*

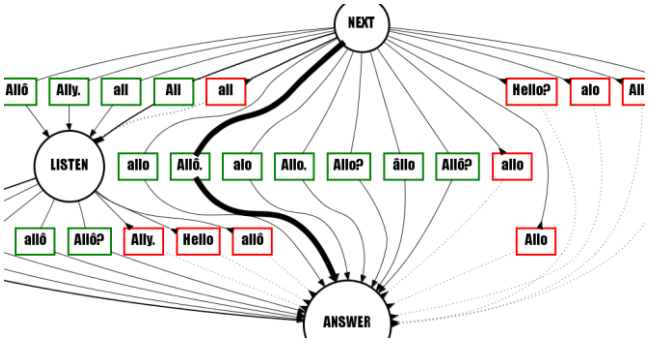


Figure 9b. Behavior Graph: French, *Algorithm 4*

6. CONCLUSIONS

In this paper, we have shared results from an empirical analysis of application of ABGG algorithms to three different learning domains. Several similarities and differences between the performances of four algorithms on problems from these three domains were discussed in the previous section.

We find that the accuracy of these algorithms suffers when they are applied to solution traces collected from a tutoring system that uses domain knowledge to process a large variety of inputs from learners. While in our previous work [4], we have recommended the use of *Algorithm 2* as the default ABGG algorithm for use within authoring tools, we find that for language learning domains, *Algorithm 4* may be preferable since it is the most accurate on the French dataset and not significantly worse than the other algorithms on the other primary metrics.

We identified multiple potential improvements to the ABGG algorithms based on these analyses. There are several domain specific nuances to the UI elements that comprise the problems in each domain. For example, in the French domain, we found steps that do not have any wrong answer. For broad use, ABGG algorithms should identify these UI elements and selectively apply the powerful assumption about retracted events. Furthermore, the algorithms can exploit additional features computed from across the multiple traces, such as the frequency of a data value at a node, to improve the accuracy of the automatically generated behavior graphs.

Finally, this paper extends our recent work on use of multiple behavior demonstrations to automatically generate tutor models using ABGG algorithms. While these algorithms can be improved in specific ways discussed above, we find evidence for their applicability to multiple domains.

ACKNOWLEDGEMENTS

This research was funded by the US Office of Naval Research (ONR) contract N00014-12-C-0535.

7. REFERENCES

[1] Johnson, W. L., and Valente, A. 2008. Collaborative authoring of serious games for language and culture. In *Proceedings of SimTecT* (March 2008).

[2] Aleven, V., McLaren, B. M., Sewall, J., and Koedinger. K. R. 2006. The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (ITS'06), Ikeda, M., Ashley, K. D., and Chan, T.W. (Eds.). Springer-Verlag, Berlin, Heidelberg, 61-70.

[3] Kumar, R., Roy, M.E, Roberts, R.B., and Makhoul, J.I. 2014. Towards Automatically Building Tutor Models Using Multiple Behavior Demonstrations. In *Proceedings of 12th Intl. Conf. on Intelligent Tutoring Systems* (ITS 2014), Honolulu, HI.

[4] Kumar, R., Roy, M.E, Roberts, R.B., and Makhoul, J.I. 2014. Comparison of Algorithms for Automatically Building Example-Tracing Tutor Models. In *Proceedings of 7th Intl. Conf. on Educational Data Mining* (EDM 2014), Honolulu, HI.

[5] Aleven, V., Mclaren, B. M., Sewall, J., and Koedinger. K. R. 2009. A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. *Int. J. Artif. Intell. Ed.* 19, 2 (April 2009), 105-154.

[6] Kumar, R., Sagae, A., and Johnson, W. L. 2009. Evaluating an Authoring Tool for Mini-Dialogs. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education*, Dimitrova, V., Mizoguchi, R., du Boulay, B., and Graesser, A. (Eds.). IOS Press, Amsterdam, The Netherlands, The Netherlands, 647-649.

[7] Kumar, R., Roy, M.E, Pattison-Gordon, E. and Roberts, R.B. 2014. General Purpose ITS Development Tools. In *Proceedings of Workshop on Intelligent Tutoring System Authoring Tools, 12th Intl. Conf. on Intelligent Tutoring Systems* (ITS 2014), Honolulu, HI.

[8] Question Generation Workshops. 2008-2011. http://www.questiongeneration.org/

[9] Barnes, T. and Stamper, J. 2008. Toward Automatic Hint Generation for Logic Proof Tutoring Using Historical Student Data. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (ITS '08). Woolf, B. P., Aimeur, E., Nkambou, R., and Lajoie , S. (Eds.). Springer-Verlag, Berlin, Heidelberg, 373-382.

[10] Eagle, M., Johnson, J., and Barnes, T., 2012. Interaction Networks: Generating High Level Hints Based on Network Community Clusterings, In *Proceedings of the 5th International Conference on Educational Data Mining* (EDM 2012). Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Eds.). 164-167

[11] McLaren, B.M., Koedinger, K.R., Schneider, M., Harrer, A., and Bollen, L. 2004. Bootstrapping Novice Data: Semi-Automated Tutor Authoring Using Student Log Files. In *Proceedings of the Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, 7th International Conference on Intelligent Tutoring Systems* (ITS 2004). August 2004

[12] Pavlik, P.I., Cen, H., and Koedinger, K.R. 2009. Learning Factors Transfer Analysis: Using Learning Curve Analysis to Automatically Generate Domain Models, In *Proceedings of the 2nd International Conference on Educational Data Mining* (EDM 2009). Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.). 121-130

[13] Koedinger, K.R., Mclaughlin E.A., and Stamper, J.C. 2012. Automated student model improvement, In *Proceedings of the 5th International Conference on Educational Data Mining* (EDM 2012). Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Eds.). 17-24

[14] Sudol, L.A, Rivers, K., and Harris, T.K. 2012. Calculating Probabilistic Distance to Solution in a Complex Problem Solving Domain, In *Proceedings of the 5th International Conference on Educational Data Mining* (EDM 2012). Yacef, K., Zaïane, O., Hershkovitz, H., Yudelson, M., and Stamper, J. (Eds.). 144-147

[15] Johnson, M., Eagle, M., Stamper, J., and Barnes, T. 2013. An Algorithm for Reducing the Complexity of Interaction Networks, In *Proceedings of the 6thInternational Conference on Educational Data Mining*, (EDM 2013). D'Mello, S. K., Calvo, R. A., Olney, A. (Eds.). 248-251

[16] Ritter, R., Harris, T.K, Nixon, T., Dickison, D., Murray, R.C., and Towle, B. 2009. Reducing the Knowledge Tracing Space, In *Proceedings of the 2ndInternational Conference on Educational Data Mining* (EDM 2009). Barnes, T., Desmarais, M., Romero, C., Ventura, S. (Eds.). 151-160

[17] Gusfield, D. 1997. *Algorithms on Strings, Trees and Sequences.* Cambridge University Press, New York.

[18] Yen, J. Y. 1971. Finding the K Shortest Loopless Paths in a Network. *Management Science* 17(11). 712-716

[19] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., and Stamper, J. 2010. A Data Repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.). Boca Raton, FL: CRC Press

[20] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T. E., Upalekar. R, Walonoski, J.A., Macasek. M.A. and Rasmussen, K. P. 2005. The Assistment project: Blending assessment and assisting. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) IOS Press. 555-562.

[21] VanLehn, K., Lynch, C., Schulze, K. Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. 2005. The Andes physics tutoring system: Lessons Learned. In *International Journal of Artificial Intelligence and Education*, 15 (3), 1-47

[22] Strader, R. and Thille, C. 2012. The Open Learning Initiative: Enacting Instruction Online. In *Game Changers: Education and Information Technologies*. Oblinger, D.G. (Ed.) Educause. 201-213.

AGG: Augmented Graph Grammars for Complex Heterogeneous Data

Collin F. Lynch
Intelligent Systems Program
and Learning Research & Development Center
Pittsburgh, Pennsylvania, U.S.A.
collinl@cs.pitt.edu

ABSTRACT

The central goal of educational datamining is to derive crucial pedagogical insights from student, course, and tutorial data. Real-world educational datasets are complex and heterogeneous comprising relational structures, social connections, demographic information, and long-term assignments. In this paper I describe *Augmented Graph Grammars* a robust formalism for graph rules that provides a natural structure for evaluating complex heterogeneous graph data. I also describe *AGG* an Augmented Graph Grammar engine written in Python and briefly describe its use.

Keywords

Augmented Graph Grammars, Graph Analysis, Argument Diagrams, Complex Data, Heterogeneous Data

1. INTRODUCTION

The central goal of educational datamining is to draw pedagogical insights from real-world student data, insights which can inform instructors, students, and other researchers. While robust analytical formalisms have been defined for categorical, numerical, and relational data most real-world educational data is complex and heterogeneous combining textual, numerical, and relational features. In large course settings such as a lecture course or MOOC, for example, students may form dynamic working groups and collaborate on complex assignments. They may also be given a flexible set of reading, writing, or problem-solving tasks that they can choose to complete in any order. This process data can be encoded as a graph with nodes representing individual assignments and reading materials and arcs representing group relationships or traversal order. In order to capture important features of this rich graph data and to identify key relationships between teamwork, written text, and performance, it is necessary to apply a rule structure that can capture them naturally.

Individual student assignments can also contain heterogeneous data. Argument diagrams, for example, have been used to teach writing, argumentation, and scientific reasoning [10, 2, 19]. These structures reify real-world arguments as graphs using complex node and arc types to represent argumentative components such as hypothesis statements, citations, and claims. These complex elements can include types, text fields for short notes or free-text assertions, links to external resources, and other data.

A sample student-produced argument diagram drawn from my thesis work at the University of Pittsburgh is shown in Figure 1. This work focused on the use of argument diagrams to support students in developing written scientific reports and in identifying *pedagogically-relevant* diagram structures that can be used to predict students' subsequent performance (see [8]). The diagram contains a central *claim* node representing a research claim. This node has a single text field in which the claim is stated. This is, in turn, connected to a set of *citation* nodes representing related work via a set of *supporting*, *opposing*, and *undefined* arcs colored green, red, and grey, respectively. The citation nodes each contain two text fields, one for the citation information and the other for a summary of the cited work, while the arcs contain a single text field for the *warrant* or explanation of why the relationship holds. At the top of the diagram there is a single disjoint *hypothesis* node which contains two text fields: a *conditional* or IF field, and a *conditional* or THEN field.

This diagram contains a number of pedagogically-relevant issues. Some of them are purely structural such as the disjoint hypothesis node, and the fact that the supporting and opposing arcs are drawn from the claim to the citations and not vice-versa. It also contains more complex semantic issues such as the fact that the text fields on the arcs contain summary information for the cites not explanations of the relationship, and the fact that the opposing citations, citations that disagree about the central claim node have not been distinguished from one-another via a comparison arc. Problems such as these can be detected via complex rules, and I have previously shown that the presence of such problems are predictive of students' subsequent performance [8, 10, 9]. This detection and remediation, however requires the development of rules that can incorporate complex structural and textual information.

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

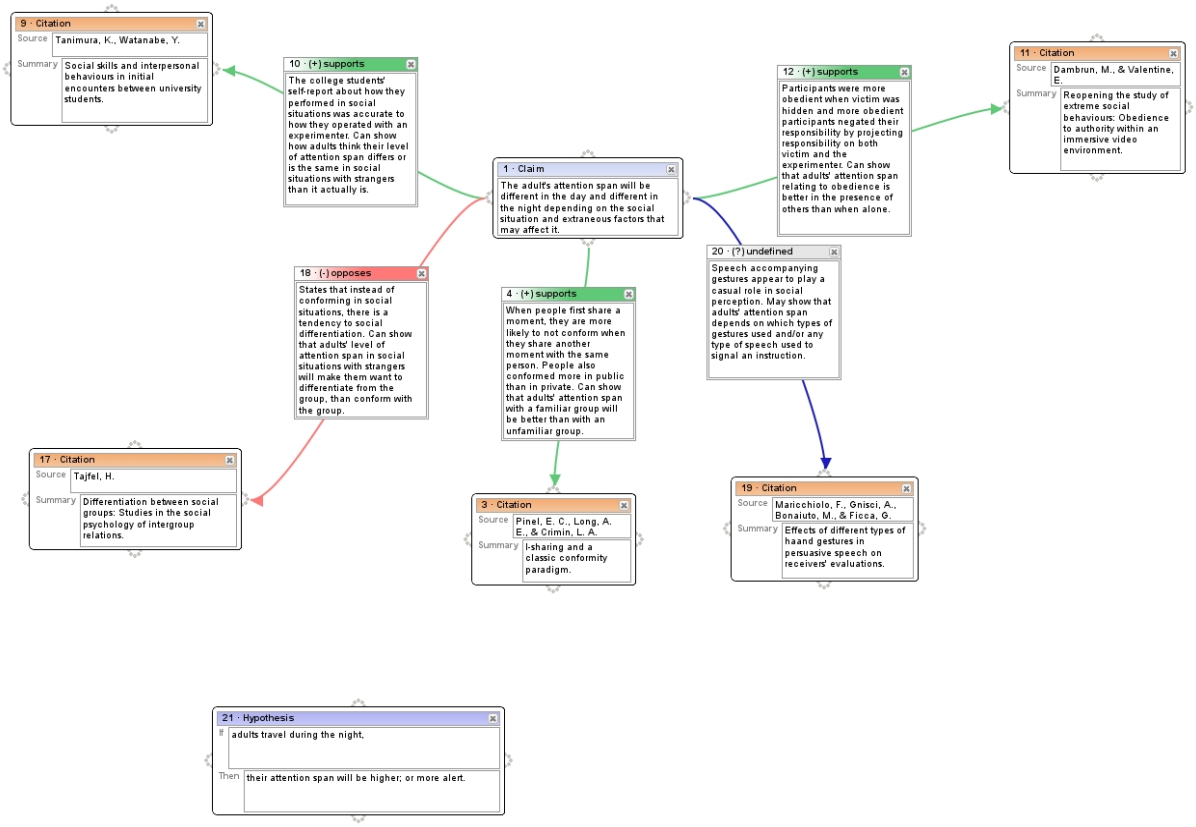


Figure 1: A segment of a student-produced LASAD diagram representing an introductory argument. It contains a central *claim* node surrounded by *citation* nodes. The isolated node is a *hypothesis* that has not been integrated into the argument.

Automatic graph analysis is central to a number of research domains including strategy transfer in games [4], automatic recommendations [1], cheminformatics [12], and social network detection [11]. Graph analysis algorithms have been used to define educational communities [15, 16, 5]) and to automatically grade existing datasets [8, 10, 9]. Graphical structures have also been used in tutoring contexts to represent student work via argument diagrams of the type shown above (see [14, 7] or to provide connection representations [19] for student guidance.

My focus in the present work is on the development of graph *rules* that is logical graph patterns that match arbitrary graph structures based upon content and structure information. While arbitrary graph matching is NP-Hard (see [18]) it is of practical importance, particularly in relational domains such as argument diagrams or student groups where our goal is to identify complex structures that may be evidence of deeper pedagogical issues. To that end, I will introduce *Augmented Graph Grammars* a robust rule formalism for complex graph rules and will describe *AGG* and augmented graph grammar engine for educational datamining. Both were developed as part of my thesis work at the University of Pittsburgh.

2. AUGMENTED GRAPH GRAMMARS

Graph Grammars, as described by Rekers and Schürr, are formal grammars whose atomic components are graphs or graph elements, and whose productions transpose one graph to another [17]. More formally, they define graph-grammars and productions as:

Definition 3.6 A graph grammar GG is a tuple $(A; P)$, with A a nonempty initial graph (the axiom), and P a set of graph grammar productions. To simplify forthcoming definitions, the initial graph A will be treated as a special case of a production with an empty left-hand side. The set of all potential production instances of GG is abbreviated with $PI(GG)$.

Definition 3.2 A (graph grammar) production p := (L; R) is a tuple of graphs over the same alphabets of vertex and edge labels LV and LE. Its left-hand side lhs(p) := L and its right-hand side rhs(p) := R may have a common (context) subgraph K if the following restrictions are fulfilled:

- $\forall e \in E(K) \Rightarrow s(e) \in V(K) \wedge t(e) \in E(K)$ with $E(K) := E(L) \cap E(R)$ and $V(K) := V(L) \cap V(R)$ i.e. sources and targets of common edges are common verticies of L and R, too.

- $\forall x \in L \cap R \Rightarrow l_L(x) = l_R(x)$ i.e. common elements of L and R do not differ with respect to their labels in L and R.

Thus graph grammars are systems of production rules analogous to context-sensitive string grammars (see [18]). For reasons of efficiency Rekers and Schürr restrict their focus to *layered graph-grammars* where all productions must be *expansive* with the left-hand-side being a subgraph of the right. Classical graph grammars, like string grammars, assume a fixed alphabet of simple statically-typed node and arcs and can be used both to generate matching graphs programmatically or to parse matching graphs via mapping and decomposition. My focus in the present work is on graph matching which occurs via iterative mapping.

Let $G_i =< \{n_o, \ldots\}, \{e(n_p, n_q), \ldots\} >$ and $G_j =< \{m_o, \ldots\}, \{e(m_k, m_l), \ldots\} >$ be graphs and let $M = \{< n_a, M - b > \ldots\}$ me a *mapping* from G_i to G_j that links nodes of the two. In the context of a mapping, G_i and G_k are called the *source* and *target* graphs respectively. A mapping M_{G_i,G_j} from G_i to G_j is *valid* if and only if the following holds:

$$\forall n_x \in G_i : \exists < n_x, m_y > \in M_{G_i,G_j}$$

$$\neg\exists\{< n_x, m_y >, < n_r, m_k >\} \subseteq M_{G_i,G_j} : (x = r) \vee (y = k)$$

$$\forall e(n_x, n_y) \in G_i : \{< n_x, m_y >, < n_r, m_k >\} \subseteq M_{G_i,G_j} : \exists e(m_y, m_k) \in G_j$$

For the remainder of this paper all elements in a source graph will be labeled alphabetically (e.g. a, Q) while elements in the target graphs will be referenced numerically (e.g. $1, 2, e(2,3), \overrightarrow{e(4,5)}$).

Augmented Graph Grammars are a richer formalism for graph rules that treat nodes and arcs as complex components with optional sub-fields including flexible text elements or other types. Augmented graph grammars have been previously described by Pinkwart et al. in [13]. There the authors focused on the use of augmented graph grammars for tutoring. An Augmented Graph Grammar is defined by: a graph ontology that specifies the complex graph elements and functions available; a set of graph classes that define matching graphs; and optional graph productions and expressions that provide for recursive class mapping and logical scoping. I will describe each of these components briefly below. For a more detailed description see [8].

2.1 Graph Ontology

In a simple graph grammar of the type used by Rekers and Schürr the set of possible node and arc types ($\sum$) is fixed with the elements being atomic, static, and unique. In order to process complex structures such as the argument diagram shown in Figure 1, a more complex structure is required. Thus augmented graph grammar ontologies are defined by a set of element types $O = \{N_0, \ldots N_m, E_0, \ldots, E_p\}$ such that each element has a unique list of fields and field types as well as applicable functions over those fields. The ontology must also specify appropriate relationships between the fields and operations that can be used on them.

While showing a complete ontology is beyond the scope of this paper an illustrative example can be found in Figure 2. This illustrates the field definitions for the citation and hypothesis nodes shown above. Both node types contain two sub-fields of type String. For each of these fields an additional function is defined '*.Words' which returns a set of all the words found in the field.

```
{
 Nodes:{
   Citation:{
     Cite(String)
     Cite.Words(StringSet)
     Summary(String)
     Summary.Words(StringSet)
   }

   Hypothesis: {
     If(String)
     If.Words(StringSet)
     Then(String)
     Then.Words(StringSet)
   }
 }
Arcs:{
  Comparison: {
   ...

Types: { String, StringSet }
...
```

Figure 2: An illustrative subset of a sample graph ontology for scientific argument diagrams.

2.2 Graph Classes

The core component of an augmented graph grammar is the graph *class*. A class C_i is defined by a 2-tuple $< S_i, O_i >$ where S_i is a graph *schema* and O_i is a set of *constraints*. A class defines a space of possible graphs which satisfy both the schema and the constraints. Classes are not required to be unique nor are the set of matching graphs for a given pair of classes required to be disjoint. A sample named class $R07a$ is shown in 3. This class is designed to detect instances of *Related Uncompared Opposition* in scientific argument diagrams. That is subgraphs where there exists a pair of citation nodes a, and b that disagree about a shared target node t, are not connected via a comparison arc c, and which share some relevant textual content. As I noted above, this type of structure can be found in Figure 1.

2.2.1 Graph Schema

A *Schema* is a graph structure that defines a space of possible graphs topologically. Schema are defined by a set of *ground nodes* (e.g. t, a, & b in Figure 3) which must match a single node in a target graph, a set of *ground arcs* that must likewise match a single arc in the target graph (e.g. c), and an optional set of *variable arcs* which must match a nonempty subgraph defined by a graph production. By convention, ground elements are denoted via lower-case names while variable elements are denoted by capitalized names.

In addition to the ground and variable distinctions arcs within a schema may be one of four types: *directed* (e.g. O, &

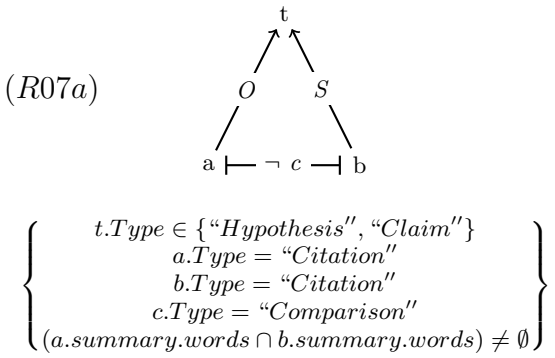


Figure 3: *Related Uncompared Opposition* A simple augmented graph grammar rule that detects related but uncompared counterarguments. The rule shows a two citation nodes (*a*, & *b*) that have opposing relationships with a shared hypothesis or claim node (*t*) and do not have a comparison arc (*c*) drawn between them. The arcs *S* and *O* represent recursive supporting and opposing paths.

S), of *unknown direction*, *undirected* (e.g. c), and *undefined*. Directed arcs will only match directed arcs in the base graph oriented in the same direction. Thus, given a base graph containing an arc $\overrightarrow{e(1,2)}$ and a schema with a directed arc $\overrightarrow{e(n,m)}$ the schema will only match cases where $\{<n,1>,<m,2>\} \subseteq M$. Unknown direction schema arcs may match a directed arc oriented in any order but will *not* match an undirected arc (e.g. $e(2,3)$). Undirected arcs (e.g. $\neg c$) will not match a directed arc. And, undefined arcs may match a directed or undirected arc in any order.

As the example shows arcs may be also be negated (e.g. $\neg c$) in which case the schema matches a graph if and only if no match can be found for the negated arc. Thus the schema shown will only match ground graphs with no arc between the elements assigned to a and b. More complicated cases of negation may be formed using graph expressions which are defined below.

The elements of a Schema must also be *non-repeating* that is, no two elements in a schema may be matched to the same element in the target graph. Thus each element in a schema must match at least one unique node or arc with variable elements possibly accounting for more than one element.

2.2.2 Constraints

Constraints represent individual bounds or limits on the ground elements of a schema. Constraints are specified using a set-theory syntax (e.g. $t.Type \in \{\text{"}Hypothesis\text{"}, \text{"}Claim\text{"}\}$) and may draw on any of the node or arc features, subfields, or functions specified in the ontology. *Unary Constraints* apply to a single element (e.g. $a.Type = \text{"}Citation\text{"}$). *Binary Constraints* (e.g. $(a.summary.words \cap b.summary.words) \neq \emptyset$) specify a relationship between two distinct ground elements.

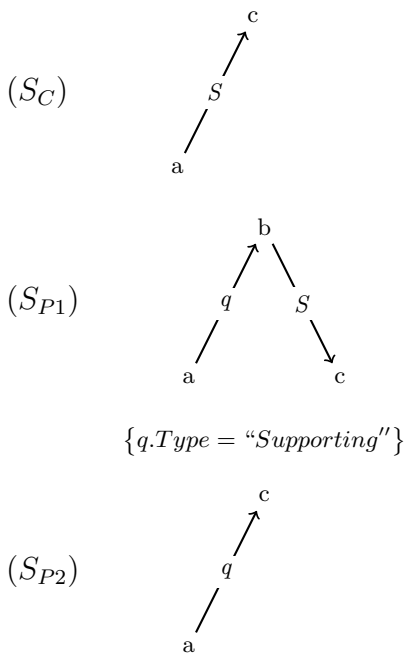


Figure 4: A simple recursive rule production for S that defines a *supporting path*.

2.3 Graph Productions

A graph production $C_l \Rightarrow C_{r1}|C_{r2}...$ is a context-sensitive production rule that maps from a graph class containing a single *production variable* to one or more alternate expansions. Graph productions are used to match layered subgraphs to the variable arcs. A simple recursive production rule for the variable element $\overrightarrow{S(b,t)}$ is shown in Figure 4.

The rule is defined by the *context class* S_C, and the two *production classes* S_{P1} and S_{P2}. The context class is used as a key for the production application. It must contain exactly one variable arc, the *production variable*, and no constraints. The ground nodes a and c are *context nodes* and are used to ground the production for mapping. They must be present in all of the production rules. All production rules must be *expansive* with each of the production classes containing at least one ground element not present in the context class. Recursive productions are thus handled by iteratively grounding the mapping with additional context and, as per the *non-repeating* requirement, these rules must consume additional elements of the graph. Production rules are thus mapped in a layered fashion like the grammars defined by Rekers and Schürr.

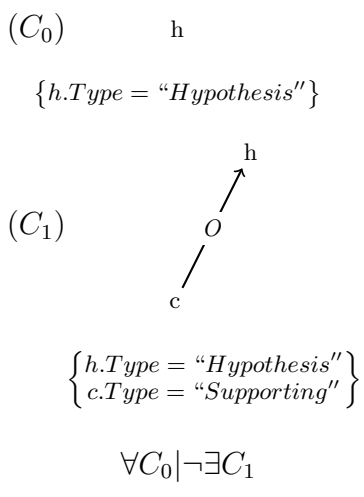


Figure 5: A simple Graph expression that tests for unopposed hypotheses.

2.4 Graph Expressions

Graph expressions are logical rules of the form:

$$S_0C_0 \quad | \quad S_1C_1 \quad | \quad \dots \quad | \quad S_mC_m$$

where each C_i is a graph class and each S_i is a logical quantifier from the set: $\{\forall, \neg\forall, \exists, \neg\exists\}$. The expressions allow for existential and universal scoping and arbitrary negation of graph classes. The expressions represent chained logical structures with each '|' being read as "...such that ...". A sample graph expression is shown in Figure 5. This sample expression asserts that for all hypothesis nodes in the target graph there exist no citation nodes that oppose the target hypothesis. Thus it is a universal claim about a negated existential item. As this example illustrates graph expressions allow for more complex negation structures than are supported by the graph schema.

Graph expressions must be expansive or *right-grounded* such that the following constraints hold:

$$\forall C_{m \leq i > 0} \in E : C_{i-1} \subseteq_g C_i$$

$$S_m \in \{\exists, \neg\exists\}$$

That is, the schema component of class C_i must be a subgraph of all classes class C_{i+n}. This also holds true for the constraints with all constraints present in class C_i being present in classes C_{i+n}. And the rightmost class in the expression must also be an existential ($\exists$) test with optional negation.

3. AGG

AGG is a general-purpose augmented graph grammar engine that implements recursive graph matching. The system was developed in Python to support analysis of the student-produced argument diagrams described above. As such it is flexible, functions across platforms, and supports complex graph ontologies and user-defined functions. The system was designed in a modular fashion and can be linked with third-party libraries such as the NLTK [6].

At present the system uses a straightforward depth-first stack matching algorithm. Given a graph and a set of named *rules*, defined by a single graph class or expression, the system will first match all ground nodes and arcs in the leftmost *target* class. Once each ground element has been matched then the system will recursively match all variable elements in the target. If at any point the system cannot continue to match elements it will pop up the stack and repeat. Rule matching is governed by the aforementioned restrictions of expansiveness and non-repetition. If a rule is defined by a graph expression then each class match will set the context for subsequent rightmost matches. Rules defined by a single class are complete once a single match is found. The system is designed to find matches serially and can be called iteratively to extract all matching items.

In addition to basic graph grammars the AGG toolkit has the capacity to define named rules. These are named graph expressions or individual classes that will be recorded if they match. In my thesis work, I applied the AGG engine to develop a set of 42 such rules the scientific argument diagrams. These ranged in complexity from graph classes defined by a single node to more complex recursive expressions that sought to identify disjoint subgraphs and unsupported hypotheses. The example rules and expressions shown in figures 3 - 5 were adapted from this set. The rules were used for offline processing of the graphs and for prediction of student grades [10, 9].

As part of the analysis process the rules were evaluated on a set of 526 diagrams containing between 0 and 41 nodes each. While exact efficiency data was not retained the performance of the rules varied widely depending upon their construction. General recursive rules such as a test for disjoint subgraphs performed quite inefficiently while smaller chained expressions were able to evaluate in a matter of seconds on a quad-core system.

4. APPLICATIONS & FUTURE WORK

The focus of this paper was on introducing Augmented Graph Grammars and the AGG engine. The formalism provides for a natural and robust representation of complex graph rules for heterogeneous datasets. In prior work at the University of Pittsburgh I applied Augmented Graph Grammars to the detection of pedagogically relevant structures like *Related Uncompared Opposition* (see Figure 3) in argument diagrams of the type shown in Figure 1. The focus of that study was on testing whether student-produced argument diagrams are diagnostic of their ability to produce written argumentative essays. The study was conducted in a course on Psychological Research Methods at the University of Pittsburgh.

The graph features examined in that study included *chained counterarguments* which feature chains of oppositional information, and *ungrounded hypotheses* which are unrelated to cited works, and so on. The study is described in detail in [8], and a discussion of the empirical validity of the individual rules can be found in [9]. The rules were also used as the basis of predictive models for student grades described in [10]. The Augmented Graph Grammars were ideally-suited for this task as they allowed me to define clear and robust rules that incorporated the structural information in the graph, textual information within the nodes and arcs, and the static

element types. It was also possible to clearly present these rules to domain experts for evaluation.

While the AGG system is robust more work remains to be done to make it widely available, and several open problems remain for future development. As noted above, arbitrary graph parsing is NP-Hard. Consequently, many rule classes are extremely inefficient. Despite this limitation, however, real efficiency gains may be made via parallelization and memoization. I am presently researching possible improvements to the system and plan to test them with additional datasets.

Acknowledgments

This work was supported by National Science Foundation Award No. 1122504, "DIP: Teaching Writing and Argumentation with AI-Supported Diagramming and Peer Review," Kevin D. Ashley PI with Chris Schunn and Diane Litman, co-PIs.

5. REFERENCES

[1] Euijin Choo, Ting Yu, Min Chi, and Yan Lindsay Sun. Revealing implicit communities to incorporate into recommender systems. In *Proceedings of the* 15^{th} *ACM Conference on Economics and Computation*, Palo Alto, CA, 2014. Association For Computing Machinery. (in press).

[2] Evi Chryssafidou and Mike Sharples. Computer-supported planning of essay argument structure. In *Proceedings of the 5th International Conference of Argumentation*, June 2002.

[3] Diane J. Cook and Lawrence B. Holder, editors. *Mining Graph Data*. John Wiley & Sons, 2006.

[4] Diane J. Cook, Lawrence B. Holder, and G. Michael Youngblood. Graph-based analysis of human transfer learning using a game testbed. *IEEE Trans. on Knowl. and Data Eng.*, 19:1465–1478, November 2007.

[5] Rosta Farzan and Peter Brusilovsky. Annotated: A social navigation and annotation service for web-based educational resources. *Journal of the New Review of Hypermedia and Multimedia (NRHM)*, 2008.

[6] Dan Garrette, Peter Ljunglöf, Joel Nothman, Mikhail Korobov, Morten Minde Neergaard, and Steven Bird. The natural language toolkit for python (NLTK), 2014. [Online; accessed 04-29-2014].

[7] Frank Loll and Niels Pinkwart. Lasad: Flexible representations for computer-based collaborative argumentation. *Int. J. Hum.-Comput. Stud.*, 71(1):91–109, 2013.

[8] Collin F. Lynch. The Diagnosticity of Argument Diagrams, 2014. (defended January 30th 2014).

[9] Collin F. Lynch and Kevin D. Ashley. Empirically valid rules for ill-defined domains. In John Stamper and Zachary Pardos, editors, *Proceedings of The* 7^{th} *International Conference on Educational Data Mining (EDM 2014)*. International Educational Datamining Society IEDMS, 2014. (In Press).

[10] Collin F. Lynch, Kevin D. Ashley, and Min Chi. Can diagrams predict essays? In Stefan Trausan-Matu, Kristy Elizabeth Boyer, Martha E. Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, volume 8474 of *Lecture Notes in Computer Science*, pages 260–265. Springer, 2014.

[11] Sherry E. Marcus, Melanie Moy, and Thayne Coffman. Social network analysis. In Cook and Holder [3], chapter 17, pages 443–468.

[12] Takashi Okada. Mining from chemical graphs. In Cook and Holder [3], chapter 14, pages 347–379.

[13] Niels Pinkwart, Kevin D. Ashley, Vincent Aleven, and Collin F. Lynch. Graph grammars: An its technology for diagram representations. In David Wilson and H. Chad Lane, editors, *FLAIRS Conference*, pages 433–438. AAAI Press, 2008.

[14] Niels Pinkwart, Kevin D. Ashley, Collin F. Lynch, and Vincent Aleven. Evaluating an intelligent tutoring system for making legal arguments with hypotheticals. *International Journal of Artificial Intelligence in Education*, 19(4):401–424, 2009.

[15] Eve Powell and Tiffany Adviser-Barnes. A framework for the design and analysis of socially pervasive games. 2012.

[16] Eve M Powell, Samantha Finkelstein, Andrew Hicks, Thomas Phifer, Sandhya Charugulla, Christie Thornton, Tiffany Barnes, and Teresa Dahlberg. Snag: social networking games to facilitate interaction. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 4249–4254. ACM, 2010.

[17] J. Rekers and Andy Schürr. Defining and parsing visual languages with layered graph grammars. *J. Vis. Lang. Comput.*, 8(1):27–55, 1997.

[18] Michael Sipser. *Introduction to the Theory of Computation*. PWS Publishing Company, San Francisco, 1997.

[19] Daniel D. Suthers. Empirical studies of the value of conceptually explicit notations in collaborative learning. In Alexandra Okada, Simon Buckingham Shum, and Tony Sherborne, editors, *Knowledge Cartography*, pages 1–23. Springer Verlag, 2008.

Graph Mining and Outlier Detection Meet Logic Proof Tutoring

Karel Vaculík
Knowledge Discovery Lab
Faculty of Informatics
Masaryk University
Brno, Czech Republic
xvaculi4@fi.muni.cz

Leona Nezvalová
Knowledge Discovery Lab
Faculty of Informatics
Masaryk University
Brno, Czech Republic
324852@mail.muni.cz

Luboš Popelínský
Knowledge Discovery Lab
Faculty of Informatics
Masaryk University
Brno, Czech Republic
popel@fi.muni.cz

ABSTRACT

We introduce a new method for analysis and evaluation of logic proofs constructed by undergraduate students, e.g. resolution or tableaux proofs. This method employs graph mining and outlier detection. The data has been obtained from a web-based system for input of logic proofs built at FI MU. The data contains a tree structure of the proof and also temporal information about all actions that a student performed, e.g. a node insertion into a proof, or its deletion, drawing or deletion of an edge, or text manipulations. We introduce a new method for multi-level generalization of subgraphs that is useful for characterization of logic proofs. We use this method for feature construction and perform class-based outlier detection on logic proofs represented by these new features. We show that this method helps to find unusual students' solutions and to improve semi-automatic evaluation of the solutions.

Keywords

logic proofs, resolution, educational data mining, graph mining, outlier detection

1. INTRODUCTION

Resolution method is, together with tableaux proof method, one of the advanced methods taught in undergraduate courses of logic. To evaluate a student solution properly, a teacher needs not only to check the result of a solution (the set of clauses is or is not contradictory) but also to analyse the sequence of steps that a student performed—with respect to correctness of each step and with respect to correctness of that sequence. We need to take into account all of that when we aim at building a tool for analysis of students' solutions. It has to be said that for an error detection (e.g. resolution on two propositional letters, which is the most serious one) we can use a search method. However, detection of an error does not necessarily mean that the solution was completely incorrect. Moreover, by a search we can hardly discover patterns, or sequence of patterns, that are typical for wrong solutions.

To find typical patterns in wrong solutions, we developed a new method for analysis of students' solutions of resolution proofs [13, 14] and showed its good performance. Solutions were manually rewritten into GraphML and then analysed. First, the frequent patterns were found by Sleuth [16], which was suitable for this type of data—unordered rooted trees. This algorithm finds all frequent subtrees from a set of trees for a given minimum support value. Such frequent subgraphs were generalized and these generalizations used as new attributes.

The main drawback of a frequent subgraph mining algorithm itself is its strong dependence on a particular task, i.e. on the input set of clauses that has to be proved, or unproved, as contradictory. Moreover, a usage of such an algorithm is quite limited, because by setting the minimum support to a very small value, the algorithm may end up generating excessively many frequent subtrees, which consumes both time and space. The problem is that we wish to include the infrequent substructures as well because they often represent mistakes in students' solutions.

In this paper we propose a novel way of subgraph generalization that solves the problems mentioned above and is independent on the input set of clauses. We show that by means of graph mining and class outlier detection, we are able to find outlying students' solutions and use them for the evaluation improvement.

The structure of this paper is following. Section 2 brings related work. In Section 3 we introduce the source data. In Section 4 we introduce the improved method for construction of generalized resolution graphs. In Section 5 we bring the main result—detection of anomalous student solutions. Discussion and conclusion are in Sections 6 and 7, respectively.

2. RELATED WORK

Overview of graph mining methods can be found in [5]. Up to our knowledge, there is no work on analysis of student solutions of logical proofs by means of graph mining. Definitely, solving logic proofs, especially by means of resolution principle, is one of the basic graph-based models of problem solving in logic. In problem-solving processes, graph mining has been used in [15] for mining concept maps, i.e. structures that model knowledge and behaviour patterns of a student, for finding commonly observed subconcept structures. Combination of multivariate pattern analysis and hidden Markov models for discovery of major phases that students go through in solving complex problems in algebra is introduced in [1]. Markov decision processes for generating hints to students in logic proof tutoring from historical data has been solved in [2, 3, 12].

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

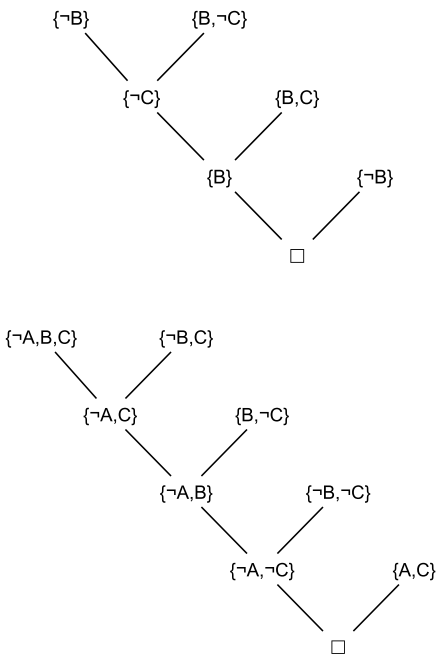


Figure 1: A correct and an incorrect resolution proof.

3. DATA

By means of a web-based tool, each of 351 students solved at least three tasks randomly chosen from 19 exercises. All solutions were stored in a PostgreSQL database. The data set contained 873 different students' solutions of resolution proofs in propositional calculus, 101 of them being incorrect and 772 correct. Two examples of solutions are shown in Fig. 1.

Common errors in resolution proofs are the following: repetition of the same literal in the clause, resolving on two literals at the same time, incorrect resolution—the literal is missing in the resolved clause, resolving on the same literals (not on one positive and one negative), resolving within one clause, resolved literal is not removed, the clause is incorrectly copied, switching the order of literals in the clause, proof is not finished, resolving the clause and the negation of the second one (instead of the positive clause). For each kind of error we defined a query that detects the error. For automatic evaluation we used only four of them, see Table `ERRORS` described in appendix A. As the error of resolving on two literals at the same time is very common and referred later in text, we denote this error as E3.

All actions that a student performed, like adding/deleting a node, drawing/removing an edge, writing/deleting a text into a node, were saved into a database together with time stamps. More details on this database and its tables can be found in appendix A.

In the data there were 303 different clauses occurring in 7869 vertices, see frequency distribution in Fig. 2. Approximately half of the clauses had absolute frequency less than or equal to three.

4. GENERALIZED SUBGRAPHS

In this section we describe feature construction from graph data. Representing a graph by values of its vertices and edges is insufficient as the structure of the graph also plays a significant role. Common practice is to use substructures of graphs as new features [5]. More specifically, boolean features are used and the value of a feature depends on whether the corresponding substructure occurs in the given instance or not.

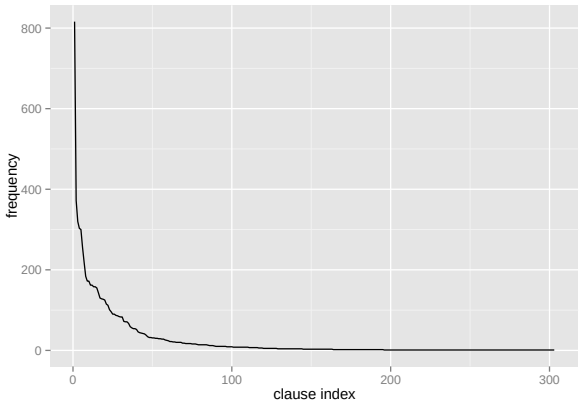


Figure 2: Distribution of clause labels ordered by frequency.

As we showed earlier, a frequent subgraph mining algorihm is inappropriate. To overcome the discussed problems, we created a new method for feature construction from our data. The idea of feature construction is to unify subgraphs which carry similar information but they differ in form. An example of two subgraphs, which differ only in variable letters and ordering of nodes and literals, is shown on the left side of Fig. 3. The goal is to process such similar graphs to get one unique graph, as shown in the same figure on the right. In this way, we can better deal with different sets of clauses with different sets of variable letters. To deal with the minimum-support problem, the algorithm for frequent subgraphs was left out completely and all 3-node subgraphs, which are described later, were looked up.

4.1 Unification on Subgraphs

To unify different tasks that may appear in student tests, we defined a unification operator on subgraphs that allows finding of so called *generalized subgraphs*. Briefly saying, a generalized subgraph describes a set of particular subgraphs, e.g., a subgraph with parents $\{A, \neg B\}$ and $\{A, B\}$ and with the child $\{A\}$ (the result of a correct use of a resolution rule), where A, B, C are propositional letters, is an instance of generalized graph $\{Z, \neg Y\}$, $\{Z, Y\} \rightarrow \{Z\}$, where Y, Z are variables (of type *proposition*). An example of incorrect use of resolution rule $\{A, \neg B\}$, $\{A, B\} \rightarrow \{A, A\}$ matches with the generalized graph $\{Z, \neg Y\}$, $\{Z, Y\} \rightarrow \{Z, Z\}$. In other words, each subgraph is an instance of one generalized subgraph. We can see that the common set unification rules [6] cannot be used here.

In this work we focused on generalized subgraphs that consist of three nodes, two parents and their child. Then each generalized subgraph corresponds to one way—correct or incorrect—of resolution rule application.

4.2 Ordering on Nodes

As a resolution proof is, in principal, an unordered tree, there is no order on parents in those three-node graphs. To unify two resolution steps that differ only in order of parents we need to define

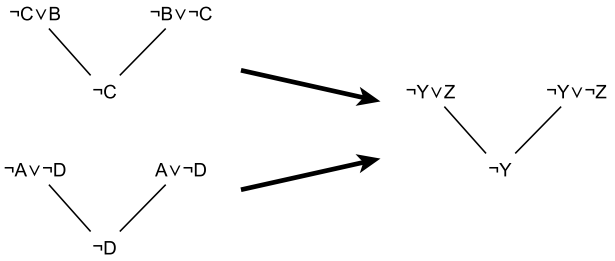


Figure 3: An example of pattern unification.

ordering on parent nodes[1]. We take a node and for each propositional letter we first count the number of negative and the number of positive occurrences of the letter, e.g., for $\{\neg C, \neg B, A, C\}$ we have these counts: (0,1) for A, (1,0) for B, and (1,1) for C. Following the ordering Ω defined as follows: $(X,Y) \leq (U,V)$ iff $(X < U \vee (X = U \wedge Y \leq V))$, we have a result for the node $\{C, \neg B, A, \neg C\}$: $\{A, \neg B, C, \neg C\}$ with description Δ = ((0,1), (1,0), (1,1)). We will compute this transformation for both parent nodes. Then we say that a node is smaller if the description Δ is smaller with respect to the Ω ordering applied lexicographically per components. Continuing with our example above, let the second node be $\{B, C, A, \neg A\}$ with Δ = ((0,1), (0,1), (1,1)). Then this second node is smaller than the first node $\{A, \neg B, C, \neg C\}$, since the first components are equal and (1,0) is greater than (0,1) in case of second components.

4.3 Generalization of Subgraphs

Now we can describe how the generalized graphs are built. After the reordering introduced in the previous paragraph, we assign variables Z,Y,X,W,V,U,... to propositional letters. To accomplish this, we initially merge literals from all nodes into one list and order it using the Ω ordering. After that, we assign variable Z to the letter with the smallest value, variable Y to the letter with the second smallest value, etc. If two values are equal, we compare the corresponding letters only within the first parent, alternatively within the second parent or child. For example, let a student's (incorrect) resolution step be $\{C, \neg B, A, \neg C\}, \{B, C, A, \neg A\} \rightarrow \{A, C\}$. We order the parents getting the result $\{B, C, A, \neg A\}, \{C, \neg B, A, \neg C\} \rightarrow \{A, C\}$. Next we merge all literals into one list, keeping multiple occurrences: $\{B, C, A, \neg A, C, \neg B, A, \neg C, A, C\}$. After reordering, we get $\{B, \neg B, C, C, C, \neg C, A, A, A, \neg A\}$ with Δ = ((1,1), (1,3), (1,3)). This leads to the following renaming of letters: $B \rightarrow Z$, $C \rightarrow Y$, and $A \rightarrow X$. Final generalized subgraph is $\{Z, Y, X, \neg X\}$, $\{Y, \neg Z, X, \neg Y\} \rightarrow \{X, Y\}$. In case that one node contains more propositional letters and the nodes are equal (with respect to the ordering) on the intersection of propositional letters, the longer node is defined as greater. At the end, the literals in each node are lexicographically ordered to prevent from duplicities such as $\{Z, \neg Y\}$ and $\{\neg Y, Z\}$.

4.4 Complexity of Graph Pattern Construction

Complexity of pattern generalization depends on the number of patterns and the number of literals within each pattern. Let r be the maximum number of literals within a 3-node pattern. In the first step, ordering of parents must be done, which takes $O(r)$ for counting the number of negative and positive literals, $O(r \log r)$ for sorting and $O(r)$ for comparison of two sorted lists. Letter substitution in the second step consists of counting literals on merged list in $O(r)$, sorting the counts in $O(r \log r)$ and renaming of letters in $O(r)$. Lexicographical reordering is performed in the last step and takes $O(r \log r)$. As construction of advanced generalized patterns is less complex than the construction of patterns mentioned above, we can conclude that the time complexity for whole generalization process on m patterns with duplicity removal is $O(m^2 + m(4r + 3r \log r))$.

[1]Ordering on nodes, not on clauses, as a student may write a text that does not correspond to any clause, e.g., $\{A, A\}$.

4.5 Higher-level Generalization

To improve performance of used algorithms, e.g. outlier detection algorithms, we created a new generalization method. This method can be viewed as a higher-level generalization as it generalizes the method described in previous paragraphs. This method uses domain knowledge about general resolution principle. It goes through all literals in a resolvent and deletes those which also appear in at least one parent. Each such literal is also deleted from the corresponding parent or parents in case it appears in both of them. In the next step, remaining literals in parents are merged into a new list *dropped* and remaining literals in the resolvent form another list, *added*. These two lists form a pattern of the higher-level generalization and we will write such patterns in the following format:

$$\underset{\text{(added)}}{\{L_{i_1}, L_{i_2}, ..., L_{i_n}\}}; \underset{\text{(dropped)}}{\{L_{j_1}, L_{j_2}, ..., L_{j_m}\}}$$

For example, if we take the generalized subgraph from the right side of Fig. 3, there is only one literal in the resolvent, $\neg Y$. We remove it from the resolvent and both parents and we get $dropped = [Z, \neg Z]$, $added = []$.

As a result, there may be patterns which differ only in used letters and order of literals in lists *dropped* and *added*. For this reason we then apply similar method as in the lower-level generalization. Specifically, we merge lists *dropped* and *added* and compute number of negative and positive literals for each letter in this new list. The letters are then ordered primarily according to number of occurrences of negative literals and secondly according to number of occurrences of positive literals. In case of tie we check ordering of affected letters only in *added* list and if needed, then also in *dropped* list. If tie occurs also in these lists, then the order does not matter. At the end, the old letters are one by one replaced by the new ones according to the ordering and the new lists are sorted lexicographically. For example, let $dropped = [X, \neg X]$, $added = [Y, Z, Z, \neg Z]$. Then $merged = [X, \neg X, Y, Z, Z, \neg Z]$ and number of occurrences can be listed as count(X, merged) = (1, 1), count(Y, merged) = (0, 1), count(Z, merged) = (1, 2). Ordering on letters can be expressed as $Y \leq X \leq Z$. Using letters from the end of alphabet we perform following substitution according to created ordering: $Y \rightarrow Z$, $X \rightarrow Y$, $Z \rightarrow X$. Final pattern will have lists $dropped = [\neg Y, Y]$, $added = [\neg X, X, X, Z]$, provided that $\neg$ sign is lexicographically before alphabetic characters. Examples of patterns with absolute support ≥ 10 are shown in Tab. 1.

4.6 Generalization Example

In this section we illustrate the whole generalization process by an example. Assume that the following 3-node subgraph has to be generalized:

Table 1: Higher-level patterns with support ≥ 10

Pattern (*added*;*dropped*)	Support
$\{\};\{\neg Z,Z\}$	3345
$\{\};\{\neg Y,\neg Z,Y,Z\}$	59
$\{\neg Z\};\{\neg Y,Y\}$	18
$\{\};\{\neg Z\}$	13
$\{\};\{\}$	10

$$P1 = \{\neg C,\neg A,\neg C,D,\neg D\}, P2 = \{\neg D,\neg A,D,C\} \rightarrow \{\neg A,A,\neg C\}$$

First, the parents are checked and possibly reordered. For each letter we compute the number of negative and positive literals in either parent. Specifically, count(A, P1) = (1,0), count(C, P1) = (2,0), count(D, P1) = (1,1), count(A, P2) = (1,0), count(C, P2) = (0,1), count(D, P2) = (1,1). Obtained counts are lexicographically sorted for both parents and both chains are lexicographically compared:

$$((1,0),(1,1),(2,0)) > ((0,1),(1,0),(1,1))$$

In this case, the result was already obtained by comparing the first two pairs, (1,0) and (0,1). Thus, the second parent is smaller and the parents should be switched:

$$P1' = \{\neg D,\neg A,D,C\}, P2' = \{\neg C,\neg A,\neg C,D,\neg D\} \rightarrow \{\neg A,A,\neg C\}$$

Now, all three nodes are merged into one list:

$$S = \{\neg D,\neg A,D,C,\neg C,\neg A,\neg C,D,\neg D,\neg A,A,\neg C\}$$

Once again, the numbers of negative and positive literals are computed: count(A, S) = (3,1), count(C, S) = (3,1), count(D, S) = (2,2). Since count(A, S) = count(C, S), we also check the counts in the first parent, P1'. As count(C, P1') = count(C, P2) < count(A, P2) = count(A, P1'), letter C is inserted before A. Finally, the letters are renamed according to the created order: $D \rightarrow Z, C \rightarrow Y, A \rightarrow X$. After the renaming and lexicographical reordering of literals, we get the following generalized pattern:

$$\{\neg X,\neg Z,Y,Z\},\{\neg X,\neg Y,\neg Y,\neg Z,Z\} \rightarrow \{\neg X,\neg Y,X\}$$

Next, we want to get also the higher-level generalization of that pattern. The procedure goes through all literals in the resolvent and deletes those literals that occur in at least one parent. This step results in a prunned version of the pattern:

$$\{\neg Z,Y,Z\},\{\neg Y,\neg Z,Z\} \rightarrow \{X\}$$

Parents from the pruned pattern are merged into a new list *dropped* and the resolvent is used in a list *added*. Thus, $added = \{X\}$ and $dropped = \{\neg Z,Y,Z,\neg Y,\neg Z,Z\}$. Now it is necessary to rename the letters once again. Lists *added* and *dropped* are merged together and the same subroutine is used as before—now the lists can be seen as two nodes instead of three. In this case, the renaming goes as follows: $X \rightarrow Z, Y \rightarrow Y, Z \rightarrow X$. At the end, literals in both lists are lexicographically sorted and the final higher-level pattern is:

$$\underset{\text{(added)}}{\{Z\}};\underset{\text{(dropped)}}{\{\neg X,\neg X,\neg Y,X,X,Y\}}$$

4.7 Use of Generalized Subgraphs

This section puts all the information from previous sections together and describes how generalized patterns are used as new features. Input data in form of nodes and edges are transformed into attributes of two types. Generalized patterns of the lower level can be considered as the first type and the patterns of higher-level generalization as the second type. One boolean attribute is created for each generalized pattern. Value of such attribute is equal to $TRUE$, if the corresponding pattern occurs in the given resolution proof, and it is equal to *FALSE* otherwise. Thus following this procedure, the resolution proofs can be transformed into an attribute-value representation as shown in Table 2. Such representation allows us to use a lot of existing machine learning algorithms.

Table 2: Attribute-value representation of resolution proofs

Instance	$Pattern_1$	$Pattern_2$	...	$Pattern_m$
1	TRUE	FALSE	...	FALSE
...	...	...	...	...
n	FALSE	FALSE	...	TRUE

5. OUTLIER DETECTION

5.1 Mining Class Outliers

In this section we present the main result, obtained from outlier detection. We observed that student creativity is more advanced than ours, and that results of the queries for error detection must be used carefully. Detection of anomalous solutions—either abnormal, with picturesque error, or incorrectly classified—helps to improve the tool for automatic evaluation, as will be shown later.

Here we focus only on outliers for classes created from error E3, the resolution on two literals at the same time, as it was the most common error. This means that the data can be divided into two groups, depending whether the instances contain error E3 or not. For other types of errors, the analysis would be similar. We also present only results computed on higher-level generalized patterns. The reason is that they generally achieved much higher outlier scores than lower-level patterns.

The data we processed had been labeled. Unlike in common outlier detection, where we look for outliers that differ from the rest of "normal" data, we needed to exploit information about a class. That is why we used weka-peka [9] that looks for class outliers [8, 10] using Random Forests (RF) [4]. The main idea of weka-peka lies in different computation of proximity matrix in RF—it also exploits information about a class label [9]. We used the following settings:

```
NumberOfTrees=1000
NumberOfRandomFetaures=7
FeatureRanking=gini
```

Table 3: Top outliers for data grouped by error E3

instance	error E3	outlier score	significant patterns [(AScore) *added*;*dropped*]	significant missing patterns [(AScore) *added*;*dropped*]
270	no	131.96	(0.96) *looping*	(-0.99) $\{\};\{\neg Z,Z\}$
396	no	131.96	(0.96) *looping*	(-0.99) $\{\};\{\neg Z,Z\}$
236	no	73.17	(0.99) $\{\};\{\neg Y,\neg Z,Y\}$	
187	no	61.03	(0.99) $\{\neg Z\};\{\neg Y,Y\}$ (0.99) $\{\};\{\neg Y,\neg Z,Y\}$	
438	yes	54.43	(1.00) $\{Z\};\{\neg X,\neg Y,X,Y\}$	(-0.94) $\{\};\{\neg Y,\neg Z,Y,Z\}$
389	yes	52.50	(1.00) $\{\};\{\neg Y,\neg Z,Y\}$	(-0.94) $\{\};\{\neg Y,\neg Z,Y,Z\}$ (-0.81) $\{\};\{\neg Z,Z\}$
74	yes	15.91	(0.98) $\{\neg Z\};\{\neg X,\neg Y,X,Y\}$ (0.98) $\{\};\{\neg X,\neg Y,\neg Z,X,Y,Z\}$	(-0.94) $\{\};\{\neg Y,\neg Z,Y,Z\}$
718	yes	15.91	(0.98) $\{\neg Z\};\{\neg X,\neg Y,X,Y\}$ (0.98) $\{\};\{\neg X,\neg Y,\neg Z,X,Y,Z\}$	(-0.94) $\{\};\{\neg Y,\neg Z,Y,Z\}$

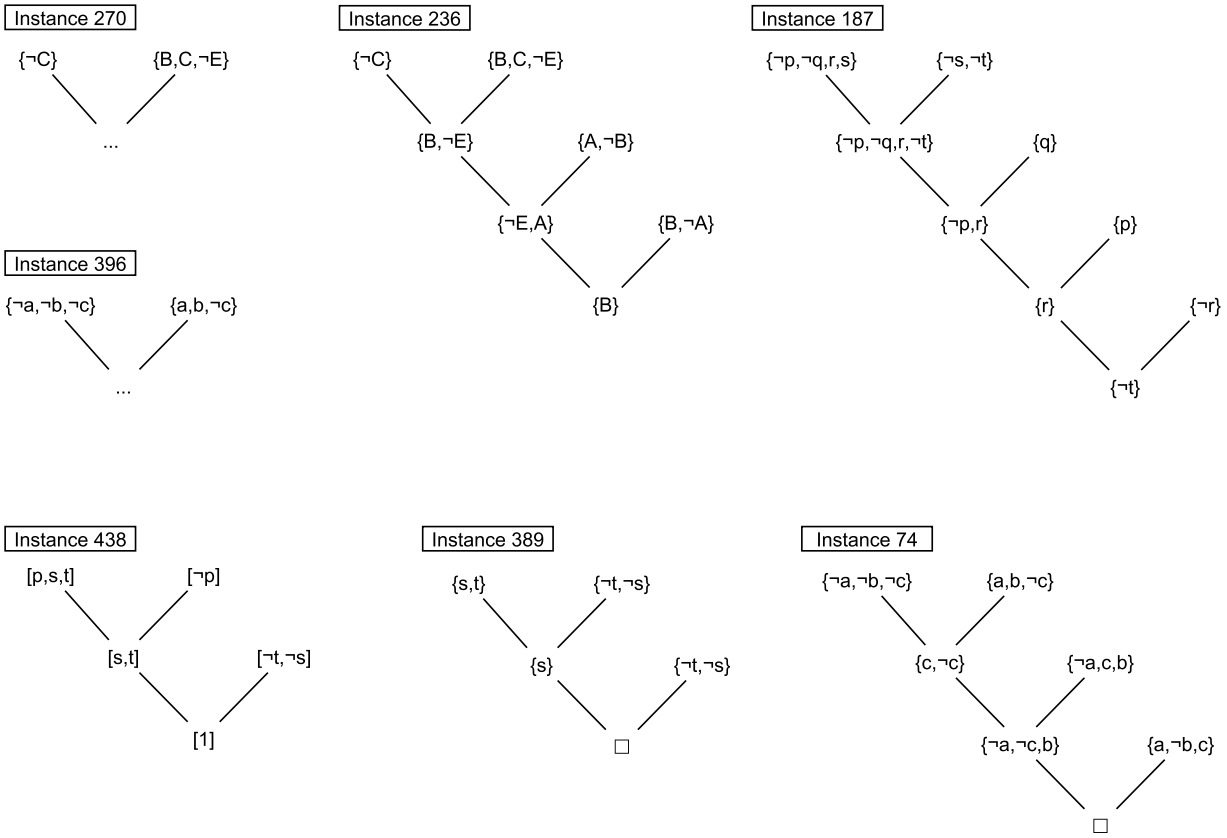


Figure 4: Drawings of the outlying instances from Table 3.

Table 4: Classification results for frequent subgraphs

Used attributes	Algorithm	Accuracy [%]	Precision for incorrect proofs	Recall
low-level generalization	SVM (SMO)	*95.2	0.94	0.61
both levels of generalization	SVM (SMO)	*96.9	0.95	0.74
both levels of generalization	J48	96.1	*0.98	0.68
both levels of generalization E3	J48	*95.4	0.87	0.72

```
MaxDepthTree=unlimited
Bootstrapping=yes
NumberOfOutliersForEachClass=50
```

Main results of outlier detection process are summarized in Table 3. When analyzing the strongest outliers that weka-peka found, we can see that there are three groups according to the outlier score. The two most outlying examples, instances numbered 270 and 396, significantly differ from the others. The second cluster consists of four examples with the outlier score between 50 and 100, and the last group is comprised of instances with the lowest score of 15.91.

As weka-peka is based on Random Forest, we can interpret an outlier by analyzing trees that classify given instance to a different class than it was labeled. Such trees show which attribute or combination of attributes lead to the resulting class. If we search for repeating patterns in those trees, we can find the most important attributes making the given instance an outlier. Using this method to interpret the instance 270, we found out that high outlier score is caused by not-applying one specific pattern (see Table 3). When setting this attribute equal TRUE, outlier score decreases to -0,40. Values of attributes of instances 396 and 270 are equal, it means that also interpretation is the same as in previous case. Similary, we found that outlierness of instance 236 is given by occurence of specific pattern in solution and non-occurence of another pattern. The value of the corresponding attribute is the only difference between instance 236 and 187. Occurence/non-occurence of this pattern is therefore the reason why instance numbered 236 achieves higher outlier score than instance 187. See again Table 3 for information about particular patterns. We further elaborated this approach of outlier explanation in the following section.

5.2 Finding Significant Patterns

As the outlier score is the only output information about the outliers, we created a simple method for finding the attributes with the most unusual values. Let x_{ij} denote the value of the jth attribute of the ith instance, which is either $TRUE$ or $FALSE$ for the pattern attributes, and $cl(i)$ denote the class of the ith instance. Then for instance i we compute the score of attribute j as:

$$AScore(i,j) = \begin{cases} \frac{|\{k|k\neq i \wedge cl(i)=cl(k) \wedge x_{kj}=FALSE\}|}{|\{k|k\neq i \wedge cl(i)=cl(k)\}|} & \text{if } x_{ij} = TRUE \\ -\frac{|\{k|k\neq i \wedge cl(i)=cl(k) \wedge x_{kj}=TRUE\}|}{|\{k|k\neq i \wedge cl(i)=cl(k)\}|} & \text{if } x_{ij} = FALSE \end{cases}$$

AScore expresses the proportion of other instances from the same class which have different value of the given attribute. If outlier's attribute equals *FALSE*, then the only difference is in the sign of the score. For example, consider our data set of 873 resolution proofs, out of which 53 proofs contain error E3. Assume that one of the 53 proofs is an outlier with an attribute equal to $TRUE$ and from the rest of 52 proofs only two proofs have the same value of this attribute as the outlier. Then the outlier's AScore on this attribute is approximately 50/52 = 0.96 and it indicates that the value of this attribute is quite unusual.

In general, the AScore ranges from -1 to 1. If the outlier resolution graph contains a pattern which is unique for the class of the graph, then the AScore of the corresponding attribute is equal to 1. On the other hand, if the outlier misses a pattern and all other graphs contain it, then the AScore is equal to -1. An AScore equal to 0 means that all other instances are equal to the outlier on the specified attribute.

5.3 Interpretation of the Outliers

Using the AScore metrics we found the patterns which are interesting for outliers in Table 3. Patterns, with AScore > 0.8 are listed in the *significant patterns* column and patterns with AScore < -0.8 in the *significant missing patterns* column.

All outliers from Table 3, except for the last one as it is almost identical to the penultimate one, are also displayed in Fig. 4. Analysis of individual outliers let us draw several conclusions. Let us remind that higher-level patterns listed in Table 3 are derived from lower-level patterns consisting of three nodes, two parents and one resolvent, and that the component *added* simply denotes literals which were added erroneously to the resolvent and the component *dropped* denotes literals from parents which participated in the resolution process. Two most outlying instances, numbered 270 and 396, also contain one specific pattern, *looping*. This pattern represents the ellipsis in a resolution tree, which is used for tree termination if the tree cannot lead to a refutation. Both instances contain this pattern, but neither of them contains the pattern of correct usage of the resolution rule, which is listed in the *significant missing patterns* column. The important thing is that these two instances do not contain error E3, but also any other error. In fact, they are created from an assignment which always leads to the *looping* pattern. This shows that it is not sufficient to find all errors and check the termination of proofs, but we should also check whether the student performed at least few steps by using the resolution rule. Otherwise we are not able to evaluate the student's skills. Moreover, there may be situations in which a student only copies the solution.

Instances with the outlier score less than 100 are less different from other instances. In particular, instances number 236 and 187 are more similar to correct resolution proofs than the instances discussed above. Yet, they both contain anomalous patterns such as $\{\}; \{\neg Y, \neg Z, Y\}$. This particular error pattern does not indicate error E3, as can be seen in Table 3. It is actually not marked as any type of error, which tells us that it is necessary to extend our list of potential errors in the automatic evaluator.

Continuing with outlier instances we get to those which contain error E3. Two of them exceed the boundary of outlier score 50, which suggests that they are still relatively anomalous. The first outlier, instance number 438, differ from other instances in an extra literal which was added into a resolvent. Specifically, the number 1, which is not even a variable, can be seen at the bottom of the resolution proof in Fig. 4. More interesting is the second instance with number 389. Error E3 was detected already in the first step of resolution, specifically when resolved on parents $\{s,t\}$ and $\{\neg t, \neg s\}$. This would not be a strange thing, if the resolvent was not s. Such a resolvent raises a question whether it is an error of type E3 or just a typing error. The latter is a less serious error.

Last two outliers in the table are almost the same so only the instance number 74 is depicted in Fig. 4. These two instances have quite low outlier score and they do not expose any shortcomings of our evaluation tool. Yet, they exhibit some outlying features such as resolving on three literals at the same time.

6. DISCUSSION

As we observed it is not sufficient to detect only the errors but we need to analyze a context in which an error appeared. Moreover, there are solutions that are erroneous because they do not contain a particular pattern or patterns. Outlier detection helped to find wrong students' solutions that could not be detected by the system

of queries even though the set of queries has been carefully built and tested on the test data. We also found a situation when a query did not detected an error although it appeared in the solution. We are convinced that with increasing number of solutions we will be able to further increase performance of wrong solution detection.

As we stressed in the introduction, this method has not been developed for recognition of correct or incorrect solutions. However, to verify that the feature construction is appropriate, we also learned various classifiers of that kind. In previous work we used only generalized patterns as attributes for classification with allerrors class attribute. However, these patterns were not sufficient for our current data. Repeating the same experiments we got the best result for SMO Support Vector Machines from Weka [7], which had 95.2% accuracy, see Table 4. Precision and recall for the class "incorrect" were 0.94 and 0.61, respectively. Minimum support for pattern selection was 0% in this case. To improve performance of classification we used the new level of generalization. Using the same settings, but now with both levels of generalized patterns, we achieved 96.9% accuracy, 0.95 precision and 0.74 recall for the class "incorrect". Similar results were obtained when only the new level of generalization was used, again with SMO. When ordered according to precision, value 0.98 was achieved by J48, but the accuracy and recall were only 96.1 and 0.68, respectively.

As one of the most common errors in resolution proofs is usage of resolution rule on two pairs of literals at the same time, we repeated the experiment, but now discarding all patterns capturing this specific kind of error. In this scenario the performance slightly dropped but remained still high—J48 achieved 95.4% accuracy, 0.87 precision and 0.72 recall. For the sake of completeness, F1 score for the class "correct" varied between 0.97 and 0.99 in all the results above.

We also checked whether inductive logic programming (ILP) can help to improve the performance under the same conditions. To ensure it, we did not use any domain knowledge predicates that would bring extra knowledge. For that reason, the domain knowledge contained only predicates common for the domain of graphs, like node/3, edge/3, resolutionStep/3 and path/2. We used Aleph system [11]. The results were comparable with the method described above.

7. CONCLUSION AND FUTURE WORK

In this paper we introduced a new level of generalization method for subgraphs of resolution proof trees built by students. Generalized subgraphs created by this special graph mining method are useful for representation of logic proofs in an attribute-value fashion. We showed how a class-based outlier detection method can be used on these logic proofs by utilization of the generalized subgraphs. We also discussed how the outlying proofs may be used for performance improvement of our automatic proof evaluator. This method may also be used for other types of data such as tableaux proofs.

As a future work we are going to analyse the temporal information, which was saved together with the structural information of logic proofs.

ACKNOWLEDGEMENTS

This work has been supported by Faculty of Informatics, Masaryk University and the grant CZ.1.07/2.2.00/28.0209 Computer-aided-teaching for computational and constructional exercises.

8. REFERENCES

[1] J. R. Anderson. Discovering the structure of mathematical problem solving. In *Proceedings of EDM*, 2013.

[2] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, pages 373–382, 2008.

[3] T. Barnes and J. Stamper. Automatic hint generation for logic proof tutoring using historical data. *Educational Technology and Society*, 13(1):3–12, 2010.

[4] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[5] D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[6] A. Dovier, E. Pontelli, and G. Rossi. Set unification. *CoRR*, cs.LO/0110023, 2001.

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[8] N. Hewahi and M. Saad. Class outliers mining: Distance-based approach. *International Journal of Intelligent Technology*, 2.

[9] Z. Pekarcikova. Supervised outlier detection, 2013. `http://is.muni.cz/th/207719/fi_m/diplomova_praca_pekarcikova.pdf`.

[10] P. Spiros and F. Christos. Cross-outlier detection. In *Proceedings of SSTD*, pages 199–213, 2003.

[11] A. Srinivasan. The Aleph Manual, 2001. `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/` [Accessed: 2014-01-09].

[12] J. C. Stamper, M. Eagle, T. Barnes, and M. J. Croy. Experimental evaluation of automatic hint generation for a logic tutor. *I. J. Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[13] K. Vaculik and L. Popelinsky. Graph mining for automatic classification of logical proofs. In *Proceedings of the 6th International Conference on Computer Supported Education CSEDU 2014*, 2014.

[14] K. Vaculik, L. Popelinsky, E. Mrakova, and J. Jurco. Tutoring and automatic evaluation of logic proofs. In *Proceedings of the 12th European Conference on e-Learning ECEL 2013*, pages 495–502, 2013.

[15] J. S. Yoo and M. H. Cho. Mining concept maps to understand university students' learning. In *Proceedings of EDM*, 2012.

[16] M. J. Zaki. Efficiently mining frequent embedded unordered trees. *Fundam. Inf.*, 66(1-2):33–52, Jan. 2005.

APPENDIX
A. DESCRIPTION OF DATA

CLAUSE – list of nodes from all graphs
. idclause – ID of the node
. coordinatex – x position in drawing
. coordinatey – y position in drawing
. timeofcreation – when the node was created
. timeofdeletion – when the node was deleted (if not deleted, value is "NA")
. idgraph – in which graph the node appears
. text – text label

EDGE – list of (directed) edges from all graphs
. idedge – ID of the edge
. starting – ID of the node from which this edge goes
. ending – ID of the node to which this edge goes
. timeofcreation
. timeofdeletion
. idgraph

ERRORS – errors found in resolution graphs (found by means of SQL queries)
. idgraph – ID of the graph
. error3 – resolving on two literals at the same time (1 = error occurred, 0 = not occurred)
. error4 – repetition of the same literal in a set
. error5 – resolving on identical literals
. error8 – no resolution performed, only union of two sets
. allerrors – any of the previously listed errors occurred / not occured

GRAPH – list of graphs
. idgraph – ID of the graph
. logintime – start of graph creation
. clausetype – either set or ordered list
. resolutiontype – type of resolution, encoded by numbers (see table RESOLUTIONTYPES)
. assignment – textual assignment of task
. endtime – end of graph creation

MOVEMENT – list of coordinate changes of nodes
. idmovement – ID of the change
. idclause – ID of the node whose coordinates were changed
. coordinatex – new x coordinate
. coordinatey – new y coordinate
. time – time of the change

RESOLUTIONTYPES – encoding of resolution types
. typeid – ID (numeric encoding)
. typetext – textual value

TEXT – list of text (label) changes of nodes.
. idtext – ID of the change
. idclause – ID of the node whose text label was changed
. time – time of the change
. text – new text (label) value

TYPES – list of resolution type and clause type changes
. idtypes – ID of the change
. resolutiontype – new value of resolution type for specific graph
. clasetype – new value of clause type for specific graph
. timeofchange – time of the change
. idgraph – ID of the graph whose values were changed

Snag'em: Graph Data Mining for a Social Networking Game

Veronica Cateté
North Carolina State University
911 Oval Drive
Raleigh, NC 27606
vmcatete@ncsu.edu

Drew Hicks
North Carolina State University
911 Oval Drive
Raleigh, NC 27606
aghicks3@ncsu.edu

Collin Lynch
North Carolina State University
911 Oval Drive
Raleigh, NC 27606
cflynch@ncsu.edu

Tiffany Barnes
North Carolina State University
911 Oval Drive
Raleigh, NC 27606
tmbarnes@ncsu.edu

ABSTRACT

New conference attendees often lack existing social networks and thus face difficulties in identifying relevant collaborators or in making appropriate connections. As a consequence they often feel disconnected from the research community and do not derive the desired benefits from the conferences that they attend. In this paper we discuss Snag'em, a social network game designed to support new conference attendees in forming social connections and in developing an appropriate research network. Snag'em has been used at seven professional conferences and in four student settings and is the subject of active research and development. The developers have sought to make the system engaging and competitive while preventing players from 'gaming' it and thus accruing points while neglecting to form real-world connections. We briefly describe the system itself, discuss its impact on users, and describe our ongoing work on the identification of critical *hub* players and important social networks.

Keywords

Social Networks, Gamification, Conferences, Underrepresented Populations

1. INTRODUCTION

Social networking is an essential task at any academic conference or professional venue. One of the primary goals of attendees is to seek out relevant work, identify potential collaborators, and to maintain existing connections. Many of these contacts are made by building upon existing relationships and by expanding the attendees existing social network. New conference goers however, particularly students and historically underrepresented groups, lack these foundational networks and thus face difficulties making connections. Based on Tinto's Theory of University Departure, increased interaction with other students, faculty, staff and community supporters can increase the retention rate of minority populations and sense of community within secondary and post-secondary academic communities [7].

In academia, sense of community has a strong positive correlation with retention [7]. Research indicates that students who do not feel as if they are part of a larger academic community are less likely to participate in extracurricular activities and organizations. This leads to lower retention rates, especially amongst minority students who suffer without a strong student support group [7]. A feeling of community can be nurtured with small group activities that augment the individual's role within a setting and helps students to foster connections [8].

Snag'em was designed as a pervasive game to encourage valuable professional networking and promote sense of community. The system's pervasive features are designed to help players translate their in-game networks directly into real world peer groups. The system was originally created for the 2009 Students and Technology Academia Research & Service (STARS) conference. This conference is unusual in that it is an academic conference designed specifically to engage with minority and female undergraduates majoring in computing fields. Students who attend the conference participate in competitions and attend training sessions to support engagement and research. Studies conducted at prior conferences has shown that while students were engaged in the training sessions and vigorously involved in learning they did not develop the lasting social connections that can arise out of conferences. Snag'em was designed to engage students in social networking through gamification of the process. Prior research has shown that social games can help people to engage in otherwise challenging or uncomfortable situations [6, 4, 2, 3].

Snag'em functions as a large human scavenger hunt. Players are assigned a set of relevant tags (e.g. "I'm a games researcher", or "I'm interested in data-mining"). They are

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Figure 1: The browser interface for mission assignments. Snag Snapshots highlight missions recently completed.

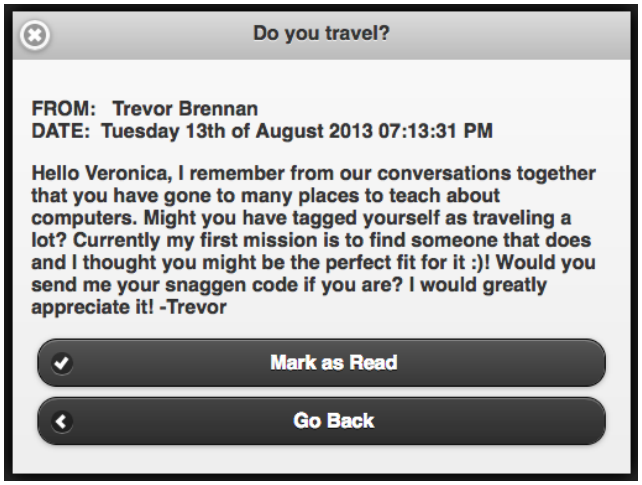


Figure 2: Here is an example of a message sent in game after a conversation between players.

then assigned a set of missions (e.g. "Find someone who specializes in HCI") which they must complete by identifying and engaging with an appropriate individual. The system was developed in PHP with a MySQL backed and provides a web-based front end for players to edit their profile and to record interactions. We have also developed a mobile version of Snag'em which allows players to access the game via tablets and smartphones. The game itself is designed for easy deployment to new conferences and we are presently adding features that will allow us to automatically populate the database with initial tags.

Figure 1 shows a snapshot of the mission browser screen from the web version of Snag'em. Contact is registered when the players enter a 4-digit ID from the other person. In addition to missions the systems also allows players to record notes about one-another for future reference (e.g. "I should e-mail my proposal to him after the conference") and to send one-another messages. A sample message from the mobile interface is shown in Figure 2. Snag'em can also be configured to suggest specific individuals that students should make contact with based upon their mutual interests or social connections.

The system logs all player interactions including tag updates, missions completed, notes made, messages, sent, connections added, and so on. This provides a rich dataset of information that we can use to analyze social patterns at conferences and to improve the impact of the intervention. In addition to the raw logs the game contains a number of features to support easy analysis. The developers have created a set of badges that allowed administrators to easily track the number of people playing via the mobile or web interfaces as well as the number of missions completed. The badge system also provides a simple visual record of the types of features (i.e. notes, tags, avatars) each player is using. The badge systems also allows administrators to note the frequency of use, time of day that players are online and so on.

To date, Snag'em has been used at seven academic conferences. It has also been deployed to help incoming freshman and transfer students connect at four academic institutions. In 2009, for example, Snag'em was used by new students in the College of Computing and Informatics at the University of North Carolina at Charlotte. Students were able to play the game during the freshman orientation week with kiosks available for students to sign up located in the College of Computing and Informatics. SNAG'EM was used alongside other social activities to get students acquainted with each other, the faculty, and the CCI campus.

2. PRIOR ANALYSIS

We have studied the impact of Snag'em on users and found that playing the game improved conference attendees' sense of community [6, 1]. We have also analyzed the existing dataset both to test the implementation of the Snag'em features, and to identify *hubs* or critical players whose activity predicts the behavior of others.

In analyzing the game mechanisms we have focused primarily on the STARS 2009 dataset. As mentioned above STARS is primarily targeted at undergraduate students specifically females and underrepresented minorities. We deployed the system via the conference infrastructure and set up a table near the registration booth. The game was active during the first two full days of the conference. The conference had 280 attendees 60.0% of whom were female (N=168) and 70% of which (N=196) were students. Roughly 28% of the conference-goers played the game (N=80) of whom 50% were female. In previous analysis 35.0% of the players were classified as active. It is important to note that this data was collected on an earlier version of SNAG'EM where players could snag each other only once, and only a single mission was available at a time. Because completing missions was significantly more difficult in this version of the game, players were classified as active if they completed at least two missions. An additional 50% of the players were classified as Interested, meaning they did more than just register for the game or that they completed one mission.

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

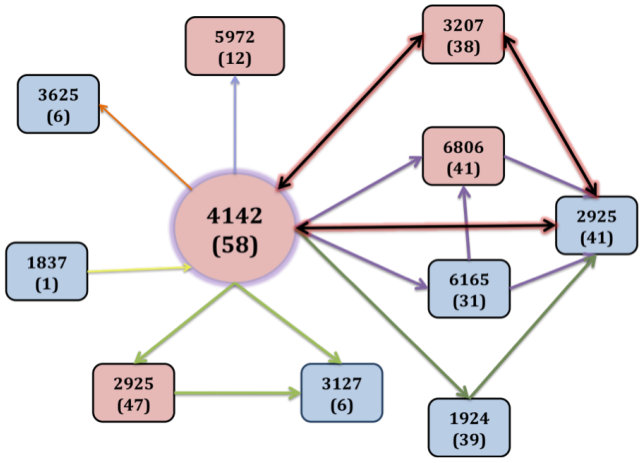


Figure 3: Visualization of community center 4142, with one of that user's maximal cliques highlighted.

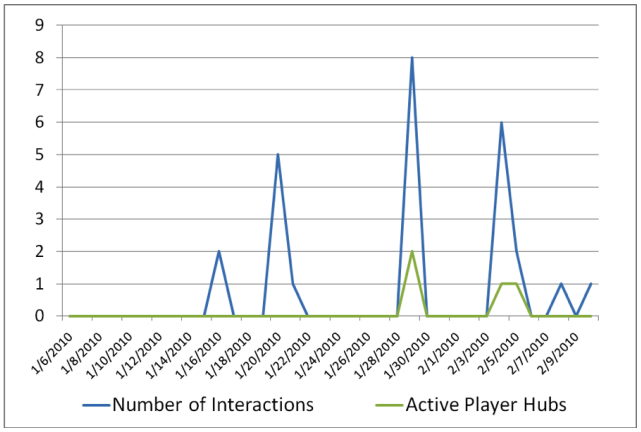


Figure 4: Correlation between active player hubs and number of interactions.

Our analysis of this data was focused primarily on the mission and scoring systems. In 2009 the mission system was relatively simple and focused solely on guiding students to locate a single individual with a desired tag. Players were then guided to record the match via the ID system discussed above. Both the missions generated and points received were determined by the state of the current network. When generating missions we attempted to ensure that they were of varying difficulty, and were relevant to the current user. In this iteration of the system the missions could only be satisfied by identifying someone whom the user had not previously snagged. The target tags were selected from the full set listed in the system. Easy missions were assigned high frequency tags (more than $\frac{1}{2}$ of the non-adjacent users), while medium missions were assigned tags that are present in $\frac{1}{4}$ of non-adjacent users and hard missions required tags present in less than $\frac{1}{4}$ of the non-adjacent community.

The difficulty of the mission determined the base score which was then modified by a *connectedness factor*. This factor was greater than 1 if adding this connection expanded your "Friends of friends," that is, the number of vertices less than 2 edges distant from the user. The connectedness factor was less than 1 if you completed the mission using the ID of a person you were already adjacent to, In this way we hoped to encourage players to branch out.

When developing the system we had hoped that players would develop social networks that exhibited breadth (i.e. meeting lots of people), depth (i.e. getting to know some individuals well), and mutuality (i.e. snags in both directions). We therefore hoped that users' immediate neighborhoods would be large and relatively dense with multiple snags between some people and bidirectional connections. When analyzing the STARS 2009 dataset, however, we found that this was not the case. Rather the game mechanics encouraged players to make a relatively large number of unrelated connections which, in turn, produced relatively broad and shallow social neighborhoods with very few inbound arcs. In fact some players actually opted to hide their IDs so that no other player could gain points by using them to complete a mission. As a consequence the attendees were actually less likely to engage in the deep and meaningful conversations required or to form lasting connections.

In response to these results we have overhauled the scoring system. This included changing the connectivity bonus to reward players based upon the size of the largest clique that they participate in. Players are now rewarded more for expanding this clique, thus deepening their social networks, than they are for adding an unrelated individual to their friends of friends. We have also allowed players to re-snag the same individual for multiple missions with a low penalty for re-snags, and have begun to reward players with points for allowing themselves to be snagged to help others complete a mission. We have not yet analyzed the effects of these changes on a the dataset.

We have used two measures of importance when identifying critical players. The first is the simple interaction frequency as measured by the number of outgoing arcs from a player in the network. The second is membership in maximal cliques, that is, cliques which are not part of a larger clique. Players that participate in a large number of maximal cliques are *hubs*. We were able to identify three distinct user communities in the STARS 2009 dataset that centered on these hubs. A sample community graph is shown in Figure 3. We also found that the activity of these hub players was highly correlated with the activity of the other players in the community (r=0.827). A graph of these spikes is shown in Figure 4. More specifically, on any day where one or more of the hub players were active, we observed spikes in the number of interactions taking place across users. We were able to observe a similar effect (r = 0.659) on days when the developers had a booth/kiosk available.

We also performed an analysis of hub players using the UNCC Student Orientation dataset described above. In this dataset 91 of the 1290 potential students registered to play Snag'em of which 22% (N=20) were female [5]. This data was collected on a version of Snag'em permitting multiple missions and allowing players to connect with the same user multiple times.We classified players as active if they completed 5 or more missions. In total, 9 users were active users during this study. However, all of these players were

moderators or members of the development team. In this deployment almost all of the game interaction took place at the registration table thus making the administrators responsible for most of the activity. We had hypothesized that the moderators would only need to initiate the game and then it would be self-sustaining. As our analysis shows however, this was not the case. In general the players did not think about the game outside of the advertised area.

3. OPEN QUESTIONS & FUTURE WORK

Our prior research has focused on identifying key players using graph methods. We plan to continue examining these key players in future work and to modify the mission selection criteria to better engage players that have not been active recently. Our chosen method of community detection, based upon maximal cliques, is both computationally expensive on large networks and can change substantially based upon small shifts in the network. Using a simpler, less volatile measure to identify community centers would allow us to adapt the gameplay based upon those communities more efficiently. This would in turn enable us to encourage new players to specifically seek out these active players in an effort to better engage them from the start. Different community detection algorithms might identify different hub players, or provide different ways of scoring missions that help to foster larger communities. Further development in this area might facilitate play in the absence of an instigating 'active player' or outside of areas with an active game station or kiosk.

One open question is how to better identify hub players during the game, and modify mission selection criteria to engage inactive players or players who don't need motivation to network. These 'social elites' are important to attract, as they are precisely who we should be encouraging our players to network with. If we are better able to build and analyze our networks, we may be able to offer features to these social elites that would attract them to Snag'em as a system more than the gamification aspects would. We hope to explore techniques for reliably generating edges and tags for users based on existing data sources like conference proceedings or citations. This would reduce the burden of entry on new players, particularly elites, and make it more likely for those users to participate in networking (if not gameplay) using SNAG'EM.

We also plan to expand our in-game evaluation of Snag'em itself. We are presently adapting the system to poll players for their opinions as the system is used. This will better help us to identify the immediate impact of the system on users' social connections. We will be deploying some of these new features of the system during the 2014 Educational Datamining Conference in London as well as subsequent conferences in 2014 and 2014.

4. ACKNOWLEDGMENTS

This research was supported by the NSF GRFP Fellowships No. 0900860 & No. 1252376 and BPC Grant No. 0739216 and No. 1042468 Thanks to all developers who have worked on the SNAG'EM project. The authors also wish to thank Shaghayegh Sahebi for her expert advice.

5. REFERENCES

[1] S. L. Finkelstein, E. Powell, A. Hicks, K. Doran, S. R. Charugulla, and T. Barnes. Snag: using social networking games to increase student retention in computer science. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, pages 142–146. ACM, 2010.

[2] M. Montola. Exploring the edge of the magic circle: Defining pervasive games. In *Proceedings of DAC*, page 103, 2005.

[3] M. Montola. A ludological view on the pervasive mixed-reality game research paradigm. *Personal and Ubiquitous Computing*, 15(1):3–12, 2011.

[4] E. Powell and T. Adviser-Barnes. A framework for the design and analysis of socially pervasive games. 2012.

[5] E. Powell, F. Stukes, T. Barnes, and H. R. Lipford. Snag'em: Creating community connections through games. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 591–594. IEEE, 2011.

[6] E. M. Powell, S. Finkelstein, A. Hicks, T. Phifer, S. Charugulla, C. Thornton, T. Barnes, and T. Dahlberg. Snag: social networking games to facilitate interaction. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 4249–4254. ACM, 2010.

[7] V. Tinto. Taking Student Retention Seriously: Rethinking the First Year of College. *NACADA Journal*, 19(2):5–10, 2000.

[8] S. White. Algorithms for estimating relative importance in networks. *Proceedings of the ninth ACM SIGKDD international*, pages 266–275, 2003.

Social Positioning and Performance in MOOCs

Suhang Jiang
School of Education
University of California, Irvine
Irvine, CA 92697
suhangj@uci.edu

Sean M. Fitzhugh
Department of Sociology
University of California, Irvine
Irvine, CA 92697
sean.fitzhugh@uci.edu

Mark Warschauer
School of Education
University of California, Irvine
Irvine, CA 92697
markw@uci.edu

ABSTRACT
Literature indicates that centrality is correlated with learners' engagement in MOOCs. This paper explores the relationship between centrality and performance in two MOOCs. We found one positive and one null correlation between centrality and grade scores at the end of the MOOCs. In both MOOCs, we found out that learners tend to communicate with learners in different performance groups. This suggests that MOOCs' discussion forum serves to facilitate information flow and help-seeking among learners.

Keywords

MOOCs; Social Positioning; Performance

1. INTRODUCTION
Massive Open Online Courses (MOOCs) have attracted over 7 million users in the past two years. In addition to offering videos and online quizzes that users can watch and take, a key feature of MOOCS is that they contain some platform for discussion among users. Indeed, discussion forums can even be considered a defining feature of a MOOC, because, without such forums, a MOOC is more like a collection of online instructional resources rather than an interactive course.

Our own preliminary data analysis of 15 MOOCs offered at the University of California, Irvine, indicates that the number of posts in MOOC discussion forums significantly predicts the number of people who complete MOOCs. Online discussion forums serve an important role in the collaborative learning process of learners [9]; however, little research explores the relationship between social positioning in the forum and the performance at the end of the course in online learning environments. To better understand learners' interaction patterns in MOOC discussions, we employed social network analysis to study the collaborative learning process in the discussions of two large MOOCs. Social network analysis is a methodology that identifies the underlying patterns of social relations of actors [11]. This paper compares the discussion forum activities of two MOOCs and examines three centrality metrics of online learners—*degree centrality, betweenness centrality, and closeness centrality*—and their relationship with learner performance.

2. RELATED WORK
Threaded discussion forums, an important component of computer assisted collaborative learning, allow learners to connect, exchange ideas, and stimulate thinking [3]. Social network analysis (SNA) is valuable for analyzing the dynamics of these discussions, as it emphasizes the structure and the relationship of actors [2]. SNA is thus a practical means for gaining insight into the relations and collaborative patterns of learners in the forum [8]. Learners' behaviors measured by social network metrics (e.g. *authority* and *hub*) in discussion forums have been identified as positively correlated with learners' engagement in MOOCs [12]. Previous research on online education indicates that network measures of *centrality* (out-degree) and *prestige* (in-degree) is strongly associated with learners' cognitive learning outcomes [10]. Research in online collaborative learning community found out that central actors tend to have higher final grades and suggested that communication and social networks should be central elements in distributed learning environments [4].

The embedded theory states that learners' embeddedness in the social networks that pervades the educational programs predicts their satisfaction and performance [1]. We hypothesize that learners' embeddeness in online learning environment is also positively correlated with their performance. Three centrality metrics, i.e.degree centrality, betweenness centrality and closeness centrality are proposed to reflect embeddness in the online learning networks.

This paper explores whether the correlation between the three centrality metrics and academic performance exists in the MOOC settings. The study mainly focused on learners who took part in the discussion forum.

3. DATASET
The project focuses on two online courses named "Intermediate Algebra" and "Fundamentals of Personal Financial Planning" delivered via the Coursera platform. The Intermediate Algebra MOOC was 10 weeks long and developed by professors from University of California, Irvine. It was open for all to enroll for free. A total 63,100 learners registered in the course, among which 43,342 learners had a record in the gradebook and 23,662 learners accessed course materials. The course consisted of lecture videos, weekly quizzes, and the final exam. The quizzes accounted for 20% of the final course grade while the final exam accounted for 80% of the final grade. Learners who obtained 65% or more of the maximum possible score were awarded with the Statement of Accomplishment, i.e. the Normal certificate. Learners who achieved 85% or more of the maximum possible score were rewarded the Statement of Accomplishment with Distinction, i.e. the Distinction certificate.

The Financial Planning MOOC was 7 weeks long and developed by a certified financial planner practitioner from University of California, Irvine. Over 110,000 learners had enrolled in the course, among which 84,234 leaners have record in the gradbook and about 55, 000 learners accessed course materials. The course evaluation consisted of weekly quizzes (30%), one peer assessment (30%) and the final exam (40%). Learners who

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

received a minimum of 70% on all graded assignment received the Statement of Accomplishment; those who received a minimum of 85% of all graded assignment obtained the Statement of Accomplishment with Distinction.

In the Algebra course, 2,126 learners participated in the forum during the 10 week course duration. Among them, 1,558 were identified as learners with an academic record, who can be found in the gradebook. It is unclear why a certain percentage of users who participated in the forum, but did not have a record in the gradebook. A possible explaination is that some are instructors and teaching assistants. The percentage of MOOC forum participation of the three performance groups is relatively constant, with 68% of forum participants as none-certificate earners. Table 1 shows the composition of forum participants.

Table 1 Composition of Discussion Forum Participants

Performance Group	Algebra		Financial Planning	
Distinction	311	20%	998	24%
Normal	193	12%	337	8%
None	1054	68%	2897	68%
In total	1558	100%	4232	100%

3.1 Network Descriptive

To create each network we used the following procedure. The forum consists of several sub-forums. Users can initiate a thread in a sub-forum, make posts to a thread, and make comments to a post. Each thread and post serves as a site of interaction among learners. Learners engage in a variety of actions: asking questions, seeking help, and providing assistance to fellow learners. We treat individuals as tied if they co-participate in a thread or a post. These ties represent communication among learners. Although one could create directed ties between individuals who address each other directly in the posts/comments, doing so would require extensive reading and coding of the data and tackling issues such as how to define direct communication (e.g., is implied communication sufficient, or must the alter be directly named?). Given the size of our data, such an approach is infeasible for our purposes.

The Algebra course discussion network has 1,389 nodes, as not all 1,558 individuals participated in the discussion forum have a record in the gradebook. The network has 3,540 edges. We illustrate it below in Figure 1. Nodes colored according to their performance groups. The network is dominated by a large, dense component with a periphery of low-degree actors. A few isolates and lone dyads are also present. Nodes of different performance groups appear to be intermixed throughout the main component and the rest of the graph.

Mean degree is 5.10, although mean degree varies slightly by performance group. Those in the "none" category have the lowest mean degree (4.36) while those in the "normal" performance have a mean degree of 8.249 and individuals earning "distinction" have a mean degree of 5.502.

More than twice as large as the algebra course discussion network, the financial planning course discussion network has 3,317 nodes and 5,505 edges. We depict the network in Figure 2. Like the algebra network, the financial planning network is

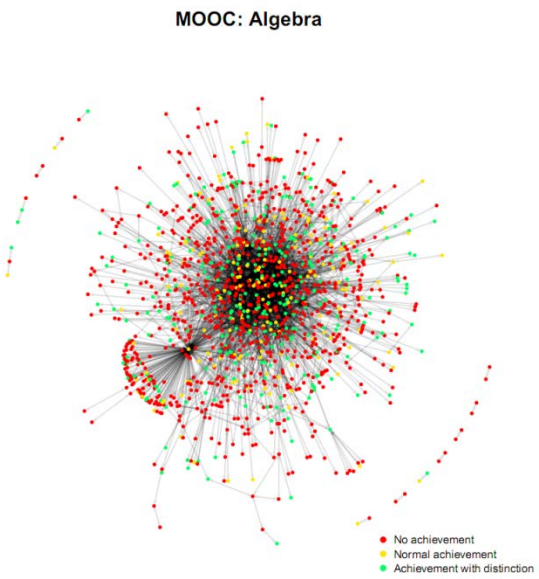


Figure 1: Algebra Network

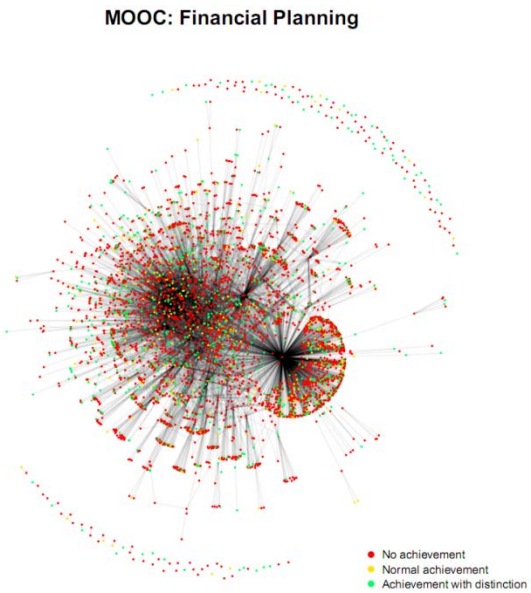


Figure 2: Financial Planning Network

dominated by a large component with a mix of isolates and smaller components. Although the financial planning discussion network is much larger than the algebra network, mean degree is lower. The average degree is 3.32. Like the algebra network, nodes with performance achievements of "normal" or "distinction" have higher degree than those in the "none" category. Those in the "none" category have an average of 2.80 ties, followed by the "normal" category with 4.15 ties, and "distinction" which has an average of 4.48 ties.

4. METHOD

Our analysis consists of analyzing the graph-level centralization and node-level centrality with permutation tests.

4.1 Centrality

Among the most common structural indices employed in the analysis of networks are centrality indices. These measures demonstrate the extent to which a node has a central position in the network [5][11]. Several measures of centrality exist and we utilize three of the most common measures in this paper: *degree*, *betweenness*, and *closeness*. One of the simplest centrality indices, *degree*, measures the total number of alters to which a node is tied. In the context of our MOOC network, this represents the number of other learners to which one is tied through participation in discussion forum threads. Those with high degree have greater levels of participation in a variety of threads that put them in contact with other learners. We also utilize *betweenness*, which measures the extent to which a node bridges other nodes by lying on a large number of shortest paths between them. Nodes with high betweenness have been described as having some degree of control over the communication of others [5] as well as greater opportunities to exert interpersonal influence over others [11]. Nodes with high betweenness in these MOOCs participate in discussions in such a way to learners across multiple forum threads. Finally, we measure *closeness*, which measures the extent to which a node has short paths to other nodes in the network. Nodes with high closeness centrality are described as being in the "middle" of the network structure [2]. Because the standard definition of closeness does not accommodate networks with multiple components, we use the Gil and Schmidt [6]approach of measuring closeness of a node as the sum of the inverse distances to all other nodes.

In addition to measuring node-level centrality, we also measure graph-level centralization. Unlike the node-level centrality indices described above, these graph-level indices produce one measure for the entire graph. These indices measure the difference between the most central node and the centrality scores for all other nodes in the network in order to provide a graph-level measure of the extent to which centrality is concentrated on a small portion of the network's nodes. We compute these centralization scores for the three aforementioned centrality measures: degree, betweenness, and closeness. These measures demonstrate the extent to which centrality is dominated by a small number of learners in the discussion network.

4.2 Permutation Test

Because we cannot guarantee the normality assumptions required by many statistical tests, we use a variety of permutation tests to assess various features of the network. While we use standard, non-parametric correlation tests, we also use non-parametric network methods. These network methods uncover structural biases by using baseline models to determine the likelihood of observing particular structural traits[2]. The results demonstrate the extent to which the network deviates from a reasonable baseline network. These tests allow us to test our hypotheses despite the statistical complexities of the network representation. We use conditional uniform graph (CUG) tests to determine whether features of our observed graph occur at levels exceeding what we would expect by chance. The CUG test conditions on a certain set of network features (typically, size, number of edges, or dyad census) and treats all graphs within that set as equally likely. It then draws at random from this set of graphs and measures whether the statistic of interest is greater, less than, or equal to the measure from our original, observed graph. To the extent that few graphs drawn from the set exceed our observed measure, the measure is higher than we expect by chance. In our analyses, we measure whether the observed levels of centralization in the discussion network are greater than what we could expect from graphs of the same size with the same number of edges.

The second non-parametric network method we employ is the matrix permutation test, often referred to as the quadratic assignment procedure or QAP test [7]. This test evaluates correlations between matrices by permuting rows and columns of the matrices, recalculating the test statistic, and measuring whether it is greater or less than the observed value. This test controls for the structure of the network and allows us to determine whether the labels (i.e., categorical attributes) of the network explain its structure. Where the correlation between the permuted graph rarely exceeds the observed test statistic, we find evidence that the observed statistic is greater than we would expect by chance. We use this technique in our MOOC network to measure whether similarity in grades between any given pair of individuals is associated with the presence of a tie between those individuals.

5. RESULTS

To determine whether observed graph-level centralization exceeds levels we would expect by chance, we use conditional uniform graph (CUG) tests conditioned on the dyad census. We hold constant the number of nodes and number of dyads (either mutual or null, given our undirected graph) when running the test. In our algebra network, degree centralization (.164), betweenness centralization (.269), and closeness centralization (.0001) all exceed chance levels, with p-values less than .01. These results are consistent with the financial planning course, where degree centralization (.354), betweenness centralization (.626), and closeness centralization (.001) were all significantly higher than baseline ($p < .01$). These results indicate that both of our observed networks have much higher levels of centralization than we would expect by chance. These networks are characterized by concentrations of centrality on a handful of nodes. While certain nodes have high levels of centrality, others lack centrality in the network.

We assess node-level centrality by relating our three centrality measures with attainment measures in the course. For each of the nodes in the network, we calculate its degree, betweenness, and closeness and measure the correlation of centrality with the final grade in the course. The correlation between the algebra course grade and degree ($r=.043$, $p=.029$), betweenness ($r=.046$, $p=.018$) are significant while closeness ($r=.028$, $p=.125$) failed to achieve significance in a non-parametric correlation test. Those with high levels of degree and betweenness centrality have higher grades in the algebra course. In the financial planning course we found no evidence of a significant correlation between course grade and degree ($r=.003$, $p=.811$), betweenness ($r=-.002$, $p=.848$), and closeness ($r=-.006$, $p=.582$). Individuals who are more central in the financial planning discussion network did not appear to have notable differences in performance compared to those with lower centrality. Although we find that both these networks have a high level of centralization, we find discrepancies between the correlation between centrality and course grade. While we find no relation between the two in the financial course, we find a weakly positive relation between centrality (except closeness) and grade in the algebra network.

Finally, we look for an association between learners' scores and their propensities to form ties with one another. We use the matrix permutation test, or QAP test, to find an association between tie formation and similar performance in the classes, where performance is measured as the overall grade or end-of-

course distinction status. To measure this association, we correlate the sociomatrix with a similarity matrix *m*, such that the *i*,*j* cell in the matrix represents the similarity in final grade between individual i and individual j. To produce this matrix we found the difference between *i*'s grade and *j*'s grade and subtracted it from 100, the maximum possible difference. The resulting scores represent similarity, where smaller scores indicate similar final grades while larger scores indicate large discrepancies between their final grades. We use the same approach to construct a distance matrix for achievement status, where learners who did not pass the class were scored as 0, while learners who passed received a 1. In the algebra course we found a significant, negative correlation between the observed sociomatrix and grade (r=-.005, p=.01) and achievement (-.007, p < .01). These results suggest that there is an association between tie formation and difference in achievement; that is, algebra learners with high achievement and high grades are *more* likely to be tied to learners with lower performance, and vice versa. In the financial planning course we found similar results: negative correlations between grade similarity (r=-.002, p=.08) and achievement status (r=-.005, p < .01). Although the relation is weak, it suggests that learners are *more* likely to form ties with learners who ended up with different achievement statuses. Learners who failed were *more* likely to communicate with learners who passed, and vice versa.

6. DISCUSSION AND CONCLUSTION

The descriptive statistic shows that the discussion forum is mainly dominated by a small percentage of learners who contributed far more than the rest of learners. This group of opinion leaders or knowledge source helps to build up and maintain the network. It also implies that the MOOCs' network is more an information network than a social network.

According to literature, a likely hypothesis would be that learners who perform well in a MOOC are more central in online discussions. However, our data demonstrated mixed results. In one MOOC (Algebra) we found a significant relationship between centrality in online discussions and student performance, while in the other MOOC (Financial Planning) we found no relationship.

It is worthwhile to consider why there might have been differences in outcomes between the two courses. Though our study was not designed to pinpoint the cause of these differences, they could be related to the differing purposes and audiences of the two MOOCs. The Algebra MOOC is more academically oriented and aims to prepare learners to succeed in higher education, whereas the Financial Planning MOOC is more geared toward assisting people in life skills. Due to the content of the Financial Planning MOOC, learners who were actively involved in the forum discussion may not have been very concerned about obtaining a certificate. Further social network analysis among a larger corpus of MOOC courses could reveal more about the relationship of course content to forum participation; we have recently obtained a corpus of data from 15 Coursera MOOCs at UCI and will conduct follow up research in this area. Additionally, moving beyond permutation tests to model-based approaches such as ERGMs could provide further insight into the properties of these networks and the relations between individual positions and outcomes.

In addition, we find in both networks a weak propensity for individuals to form ties with classmates with very different grades or attainment. This suggests that the discussion forum serves an important role in facilitating help seeking and promoting communication between the knows and the know nots.

The study also has some limitations. For example, it mainly analyzed the behavior of learners who participated in the discussion forum, which only takes up a small proportion of learners in MOOCs. In addition, we did not consider passive forum participation, such as posts or comments viewing. The future research shall include the content analysis to analyze the cognitive engagement of MOOC learners.

7. ACKNOWLEDGMENTS

We are very indebted to the Digital Learning Lab, University of California, Irvine.

8. REFERENCES

1. Baldwin, T.T., Bedell, M.D., and Johnson, J.L. The Social Fabric of a Team-Based M.B.A. Program: Network Effects on Student Satisfaction and Performance. *The Academy of Management Journal 40*, 6 (1997), 1369–1397.
2. Butts, C.T. Social network analysis: A methodological introduction. *Asian Journal of Social Psychology 11*, 1 (2008), 13–41.
3. Calvani, A., Fini, A., Molino, M., and Ranieri, M. Visualizing and monitoring effective interactions in online collaborative groups. *British Journal of Educational Technology 41*, 2 (2010), 213–226.
4. Cho, H., Gay, G., Davidson, B., and Ingraffea, A. Social networks, communication styles, and learning performance in a CSCL community. *Computers & Education 49*, 2 (2007), 309–329.
5. Freeman, L.C. Centrality in social networks conceptual clarification. *Social Networks 1*, 3 (1978), 215–239.
6. Gil, J. and Schmidt, S. The Origin of the Mexican Network of Power. *Proceedings of the International Social Network Conference*, (1996), 22–25.
7. Krackardt, D. QAP partialling as a test of spuriousness. *Social Networks 9*, 2 (1987), 171–186.
8. Nurmela, K., Lehtinen, E., and Palonen, T. Evaluating CSCL Log Files by Social Network Analysis. *Proceedings of the 1999 Conference on Computer Support for Collaborative Learning*, International Society of the Learning Sciences (1999).
9. Rabbany, R., Elatia, S., Takaffoli, M., and Zaïane, O.R. Collaborative Learning of Students in Online Discussion Forums: A Social Network Analysis Perspective. In A. Peña-Ayala, ed., *Educational Data Mining*. Springer International Publishing, 2014, 441–466.
10. Russo, T.C. and Koesten, J. Prestige, Centrality, and Learning: A Social Network Analysis of an Online Class. *Communication Education 54*, 3 (2005), 254–261.
11. Wasserman, S. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
12. Yang, D., Sinha, T., Adamson, D., and Rose, C.P. Anticipating student dropouts in Massive Open Online Courses. (2013).

Facilitating Graph Interpretation via Interactive Hierarchical Edges

Thomas S. McTavish
Center for Digital Data, Analytics & Adaptive Learning
Pearson
tom.mctavish@pearson.com

ABSTRACT

Graphs visualizations can become difficult to interpret when they fail to highlight patterns. Additionally, the data to be visualized may be hierarchical in nature. Therefore, graphs with hierarchical data need to offer means of telescoping that collapse or expand subgraphs while aggregating their data. In this paper, we demonstrate an interactive hierarchical edge graph on book prerequisite data, which can be generalized to a variety of hierarchical data. We illustrate the importance of ordering nodes (when possible) and coloring by various features. We then demonstrate various ways of performing exploratory data analysis by delivering various pieces of information on mouseovers and utilizing telescoping and filtering.

Keywords

Hierarchical edge bundling, prerequisite relationships

1. INTRODUCTION

When graphs contain many nodes and edges – especially different types of nodes and edges – they can quickly become difficult to visually interpret [7]. The common term is "hairball" as nodes and edges jumble into a tangled morass that occlude any meaningful patterns. Force-directed graphs operate to keep nodes with strong edges closer and nodes with weak or absent edges further apart [2]. This layout can aid in some contexts, but frequently exacerbates the hairball phenomenon. There are two striking visualization designs by Krzywinski and colleagues that aim at revealing interpretable patterns in graphs. At the core of each is at least one meaningful axis on which to align nodes. The first is *Circos*, which arranges sorted nodes along a circle [5]. Nodes are often displayed as arcs along the circle and edges between the arcs are visualized as chords or ribbons that cut through the middle of the circle. Circos has been used in over 500 publications, many related to large-scale genomic data. By arranging nodes along one axis in a circle, Circos easily discriminates nearby and distant edges. The widths of the nodes (length of the arc) can carry meaning and so can the width of the chord between connected arcs. Nodes and edges can also be colored to highlight features such as the node type, the source, and the target. It is also common to display many node features such as histograms of different measures within an arc, for example, Figure 3 in [6].

The other design by Krzywinski is *hive plots* [4]. Hive plots are comprised of multiple axes, each radiating from an inner ring. A given node may exist on one or more axes, aligned along the axis in some meaningful way. For example, an axis might sort nodes by different graph features such as a node's *closeness* – the average distance between a node and all others reachable from it. By placing nodes on various axes, a representation of where a node resides along some feature is captured. When edges are added, it may bring out relationships between adjacently-placed features. For example, anti-correlations of two features compared side-by-side will have many criss-crossed edges. In short, ordering nodes in some meaningful way(s) permits Circos and hive plots to better reveal patterns. Circos and hive plots, however, do not capture hierarchical relationships very well.

Hierarchical edge bundling is a visualization technique on hierarchical data that skews edges toward their parent nodes, which may be invisible in the graph [3]. The visual effect is that edges are channeled into larger, striking swaths while avoiding the direct clutter of the parent nodes. Any topology can be employed, but simpler geometric structures are most commonly used.

In this paper, we demonstrate an interactive hybrid of Circos plots with hierarchical edge bundling on mathematical book prerequisites. The books are structured hierarchically as a table of contents with chapters, sections, objectives, and exercises. Prerequisites map between objectives. The goal was to provide a means of highlighting prerequisites at the various levels, to call out important objectives, and also reveal holes. Through coloring, it is simple to discriminate chapters or to highlight nodes by features such as learner interaction frequency. Through telescoping, it is straightforward to determine those prerequisites that map across chapters, within a chapter, and within sections. Through filtering it is possible to display nodes and edges by their degree. Collectively, by aligning a curriculum along a circle, we demonstrate how this template can be used for displaying various relationships and features of hierarchical, educational data.

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

2. METHODS

Two higher education math books were selected that contained a table of contents and prerequisites as mapped by content matter experts. Interactivity data came from students, largely from the U.S., who were enrolled in courses spanning Fall semester 2012 through 2013 that used these books and the accompanying Pearson MathXL® homework system. All data was translated into JSON format for use in a web browser. The graph and its interactivity functionality was programmed using D3.js [1].

3. DEMONSTRATION

Figure 1 shows a screen shot of the graph and user controls. Displayed is a developmental math book with chapters starting at 12 o'clock and progressing clockwise. Nodes are colored by chapter and have ample spacing to easily discriminate them. Most nodes displayed are at the objective level. Within a chapter, slight separations between the nodes delineate the sections. Edges within a section are shown as little arcs. Edges within the chapter have a larger arc, and edges across chapters bend so that they bundle near to where a chapter node would be. We see various features at a glance. For example, the online appendix has no pre- or post-requisites across chapters. This is because it is shared across several books and is independent from this book.

Chapters 2, 11, and 13 are displayed at the chapter level hiding all of their section and objective nodes, whereas chapters 4 and 7 are at the section level. Chapters can be shown or hidden in the column of checkboxes on the right. For example, some appendix items have been removed from this display. The radio buttons correspond to the level of the hierarchy to display.

In Figure 1, the user has centered the mouse over the Chapter 2 node. The color of the node is green, so bold green edges reveal other chapters to which Chapter 2 is a prerequisite. Also shown are bold orange lines from Chapter 1 objectives coming into Chapter 2. The text of these pre- and post-requisites are listed on the left. We see at a glance that while Chapter 2 is prerequisite to Chapter 3, it links to other sections and objectives in a punctate fashion, completely ignoring the middle chapters of the book.

Edges are colored by their outgoing node color. Of course, they can be colored by their incoming node color or other feature. We have also colored nodes by their degree, highlighting critical objectives and important chapters. We have also colored nodes by performance measures such as the frequency of user interactions. Coloring can be selected through the pulldown menu on the top-left.

This utility also has some filtering capabilities to show/hide edges within sections, within chapters, and across chapters. Nodes can also be filtered out their degree or feature by which they are colored. Similarly, edges can be filtered out if they are below a weight threshold.

3.1 Limitations

While this visualization works well with book prerequisites, making graphs interactive as we have demonstrated, limits the quantity of nodes and edges because they have to be large enough to be selectable. Additionally, while edge bundling facilitates an interpretation of convergence, it also makes it difficult to select or hover over any individual edge for information. As presented, more than 3000 edges begins to be problematic. Similarly, when there are over 500 nodes along the circle, it can become difficult to select a node of interest with a mouse.

3.2 Next steps

This plot and circos plots have only one axis. Avenues to explore include reordering nodes along this axis by different features. Alternatively, hive plots could be extended with the ideas presented here, where various axes could utilize hierarchical data by swapping child nodes with aggregated parent nodes.

4. CONCLUSION

It is difficult to interpret graphs without an adequate visualization. In this work, we demonstrated a template that can be used on hierarchical data aligned along the axis of a circle. At a glance, it can reveal a lot of features, but through filtering, telescoping, and interactivity, exploratory data analysis can be performed to reveal features at various scales. As a template, it is quite useful for contrasting several graphs, or alternatively, illuminating various features within a general structure. For example, in our case using prerequisite data, nodes and edges might be colored by difficulty, fraction correct, time-on-task, or other measures of students interacting with these book objectives. Furthermore, this technique can be generically applied to other datasets.

5. REFERENCES

[1] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011.

[2] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, Nov. 1991.

[3] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sept. 2006.

[4] M. Krzywinski, I. Birol, S. J. Jones, and M. A. Marra. Hive plots—rational approach to visualizing networks. *Briefings in Bioinformatics*, 13(5):627–644, Sept. 2012. PMID: 22155641.

[5] M. Krzywinski, J. Schein, İ. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, Sept. 2009. PMID: 19541911.

[6] E. S. Mace, S. Tai, E. K. Gilding, Y. Li, P. J. Prentis, L. Bian, B. C. Campbell, W. Hu, D. J. Innes, X. Han, A. Cruickshank, C. Dai, C. Frère, H. Zhang, C. H. Hunt, X. Wang, T. Shatte, M. Wang, Z. Su, J. Li, X. Lin, I. D. Godwin, D. R. Jordan, and J. Wang. Whole-genome sequencing reveals untapped genetic potential in africa's indigenous cereal crop sorghum. *Nature Communications*, 4, Aug. 2013.

[7] D. Merico, D. Gfeller, and G. D. Bader. How to visually interpret biological data using networks. *Nature Biotechnology*, 27(10):921–924, Oct. 2009.

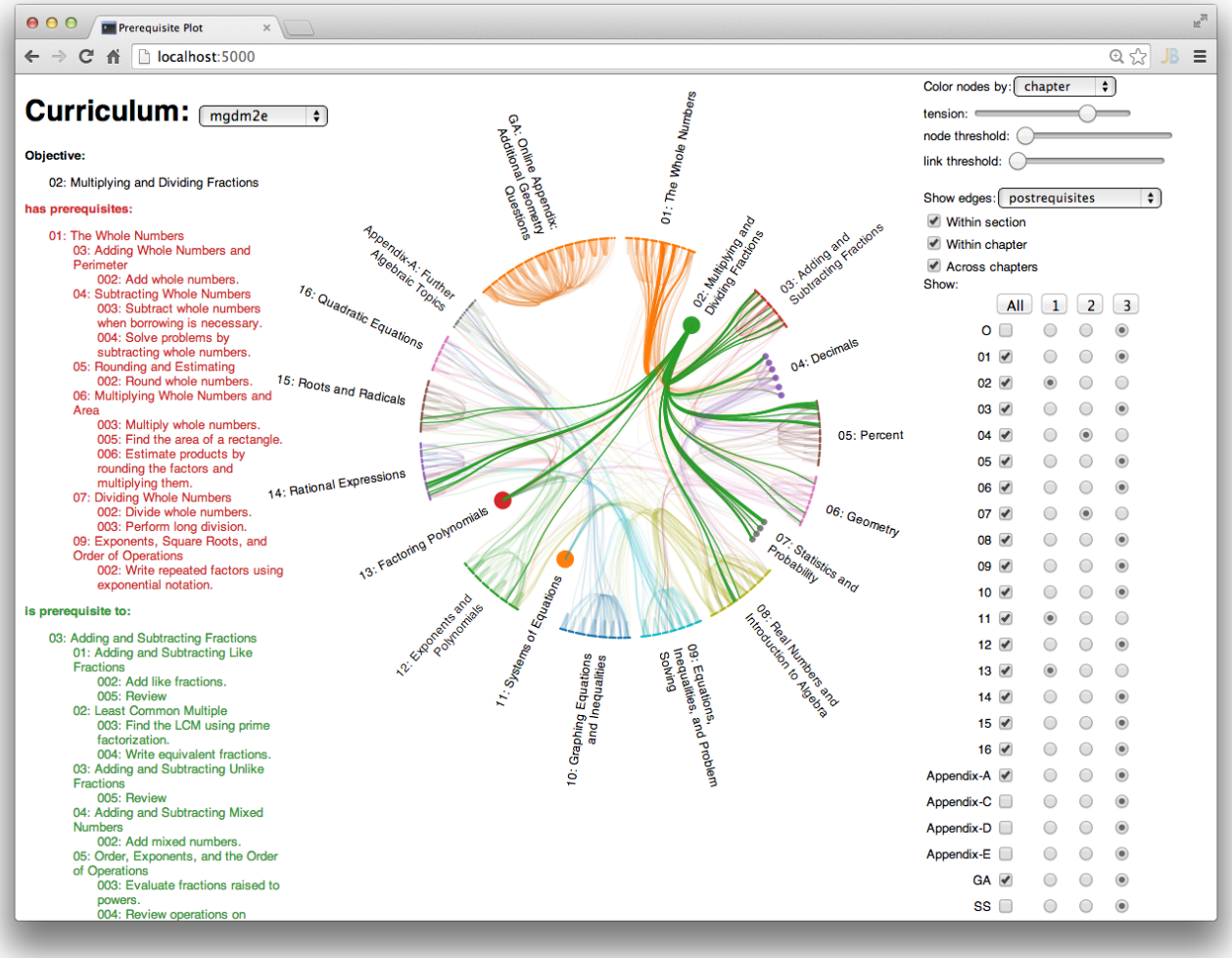


Figure 1: Screen shot of the interactive, hierarchical edge bundling graph.

Evaluation of Logic Proof Problem Difficulty through Student Performance Data

Behrooz Mostafavi
North Carolina State University
Department of Computer Science Raleigh, NC 27695
bzmostaf@ncsu.edu

Tiffany Barnes
North Carolina State University
Department of Computer Science Raleigh, NC 27695
tmbarnes@ncsu.edu

ABSTRACT

The interactions of concepts and problem-solving techniques needed to solve open-ended proof problems are varied, making it difficult to select problems that improve individual student performance. We have developed a system of data-driven ordered problem selection for Deep Thought, a logic proof tutor. The problem selection system presents problem sets of expert-determined higher or lower difficulty to students based on their measured proof solving proficiency in the tutor. Initial results indicate the system improves student-tutor scores; however, we wish to evaluate problem set difficulty through analysis of student performance to validate the expert-authored problem sets.

Keywords

Problem Difficulty, Logic Proof, Data-driven Problem Selection

1. INTRODUCTION

Effective intelligent tutoring systems present problems to students in their zone of proximal development through scaffolding of major concepts [3]. In domains such as deductive logic, where the problem space is open-ended and requires multiple steps and knowledge of different rules, it is difficult to choose problems for individual students that are appropriate for their proof-solving ability. We have developed a system that uses the data-driven knowledge tracing (DKT) of domain concepts in existing student-tutor performance data to regularly evaluate current student proficiency of the subject matter and select successive structured problem sets of expert-determined higher or lower difficulty.

We used an existing proof-solving tool called Deep Thought to test the DKT problem selection system. The system was integrated into Deep Thought and tested on a class of undergraduate philosophy students who used the tutor as assigned homework over a 15-week semester. Performance data from this experiment were compared to data from previous use of Deep Thought without the DKT problem selection system. The results of the comparison indicate that the DKT problem selection system is effective in improving student-tutor performance. However, we wish to evaluate the difficulty of presented problems using student performance data to validate the difficulty of expert-determined problem sets, and improve the system for future students.

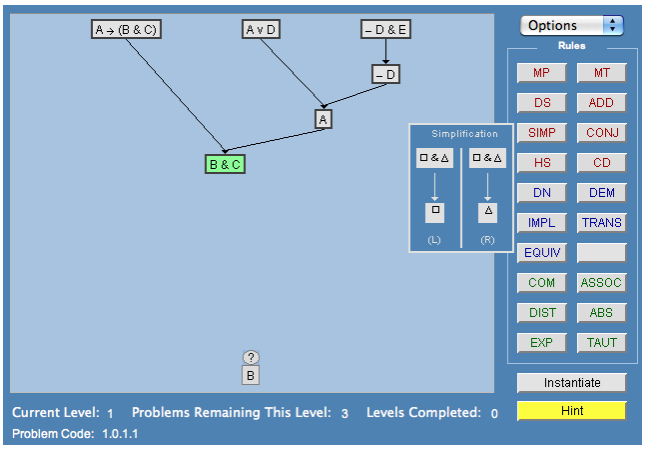


Figure 1: A screen capture of the Deep Thought tutor, showing given premises at the top, conclusion at the bottom, and rules for application on the right.

2. DEEP THOUGHT

Fig. 1 shows the interface for Deep Thought, a web-based proof construction tool created by Croy as a tool for proof construction assignments [1]. Deep Thought displays logical premises, buttons for logical rules, and a logical conclusion to be derived. For example, the proof in Fig. 1 provides premises $A \rightarrow (B \wedge C)$; $A \vee D$; and $\neg D \wedge E$, from which the user is asked to derive conclusion B using the rules on the right side of the display window.

Deep Thought keeps track of student performance for the purpose of proficiency evaluation and post-hoc analysis. As a student works through a problem, each step is logged in a database that records: the current problem; the current state of progress in the proof; any rule applied to selected premises; any premises deleted; errors made (such as illegal rule applications); completion of the problem; time taken

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

per step; elapsed problem time; knowledge tracing scores for each logic rule in the tutor.

2.1 Problem Selection

The problem selection system in Deep Thought presents ordered problem sets to ensure consistent, directed practice using increasingly related and difficult concepts. The system presents set of problems at different degrees of difficulty, determined through evaluation of current student performance in the tutor.

Evaluation of student performance is performed at the beginning of each level of problems. Level 1 of Deep Thought contains three problems common to all students who use the tutor, and provides initial performance data to the problem selection model. Levels 2–6 of Deep Thought are each split into two distinct sets of problems, labeled higher and lower proficiency. The problems in the different proficiency sets are conceptually identical to each other, prioritizing rules important for solving the problems in that level. To prevent students from getting stuck on a specific proof problem, Deep Thought allows students to temporarily skip problems within a level. A unique case occurs if a student skips a problem more than once in a higher proficiency problem set; the student will be dropped to the lower proficiency problem set in the same level, under the assumption that the student was improperly assigned the higher proficiency set (See Fig. 2).

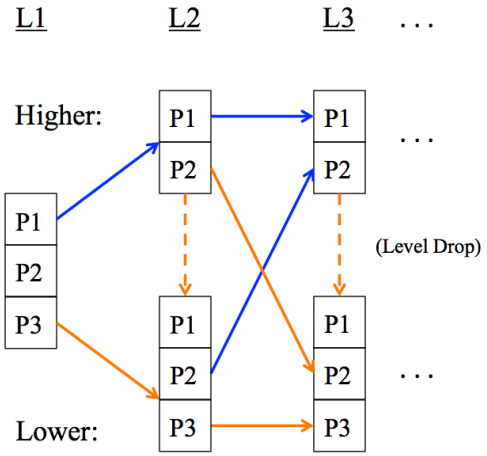


Figure 2: DT2 path progression. At each level, students are evaluated and provided either the higher or lower proficiency problem sets. Students can also be switched from the higher to lower proficiency set within a level.

2.2 Logic Proof Problems

The degree of problem solving difficulty between proficiency sets is different, as determined by domain experts. The problems in the low proficiency set require fewer numbers of steps for completion, lower complexity of logical expressions, and lower degree of rule application than problems in the high proficiency set (See Table 1).

3. DATA GRAPH REPRESENTATION

Table 1: An example of lower and higher proficiency set problems from Deep Thought requiring the same concepts: Level 4 Problem 3 from the lower proficiency set (top); Level 4 Problem 2 from the higher proficiency set (bottom). The prioritized rules required for these problems are Conjunction and Constructive Dilemma.

#	*Premise*	*Derivation*
1	$(A \rightarrow B) \wedge (\neg D \rightarrow F)$	Given
2	$A \vee \neg D$	Given
3	$\neg A \rightarrow (D \vee G)$	Given
4	$\neg A$	Given
5	$B \vee F$	1,2/Constructive Dilemma
6	$\neg D$	2,4/Disjunctive Syllogism
7	$D \vee G$	3,4/*Modus Ponens*
8	G	6,7/Disjunctive Syllogism
9	$(B \vee F) \wedge G$	5,8/Conjunction

#	*Premise*	*Derivation*
1	$Z \rightarrow (\neg Y \rightarrow X)$	Given
2	$Z \wedge \neg W$	Given
3	$W \vee (T \rightarrow S)$	Given
4	$\neg Y \vee T$	Given
5	Z	2/Simplification
6	$\neg W$	2/Simplification
7	$\neg Y \rightarrow X$	1,5/*Modus Ponens*
8	$T \rightarrow S$	3,6/Disjunctive Syllogism
9	$(\neg Y \rightarrow X) \wedge (T \rightarrow S)$	7,8/Conjunction
10	$X \vee S$	4,9/Constructive Dilemma

Deep Thought was used as a mandatory homework assignment by students in a philosophy deductive logic course ($n = 47$). Students were allowed to work through the problem sets at their own pace for the entire 15-week semester. Problem Levels 1–6 were assigned for full completion of the tutor, totaling 13–18 (out of the total tutor-set of 43) problems depending on proficiency path progression.

For the purpose of problem difficulty evaluation, progress through the tutor can be expressed as a directed graph for each individual student, with nodes in the graph each corresponding to a single problem. The node set for the graph represents the problem space for the tutor, and is the same for every student. Each problem node has the following properties:

1. Tutor Level (1–6)
2. Proficiency (High or Low)
3. Problem Number (1–3)
4. Problem Complete (True or False)
5. Expert-Authored
   (a) Required Rules
   (b) Minimal Solution
6. Corresponding Step Logs (See Section 2)

Directed edges between nodes correspond to movement between problems by the individual student, and are assigned a numerical value, ordered by increasing time stamp. The nodes and directed edges together give a map of the student's progression through the tutor. Connected nodes with false Problem Complete status represent a skipped problem, and the node adjacent to the highest numbered edge represents the student terminus point in the tutor. Isolated nodes represent non-visited problems, and are therefore un-useable for problem difficulty evaluation.

Logic proofs can also be represented as directed graphs, with each node containing a proof premise, and each directed edge indicating a node parent-child relationship, along with an applied logic rule. For example, the top proof shown in Table 1 can be represented as a graph with the premise in each line as a node, with the directed edges into that node corresponding to the derivation of that premise from parent nodes. A proof premise can either be a variable (i.e. A), a negated variable or expression (i.e. $\neg A$, or $\neg(A \wedge B)$), or an operational expression in (variable/nested expression)-operand-(variable/nested expression) form (i.e. $A \vee B$, or $(A \wedge B) \vee (A \rightarrow B)$). Nested expressions can be represented in high level form. Therefore, node premises can be categorized by their operand (conjunction, disjunction, negation, implication, equivalence), the complexity of the expression (single variable, simple expression, complex [nested] expression), and the rule used for derivation.

4. PROBLEM DIFFICULTY EVALUATION

The question at hand is how to best use the recorded data to determine proof problem difficulty through student performance. We wish to find both a classification of problem difficulty between proficiency sets in the same level, and difficulty of all problems in the tutor, compared to expert-determined classifications.

Because students follow different problem-solving paths, no student can solve all available problems in the tutor, nor are students likely to solve problems in both proficiency sets within the same level. This makes student performance comparison over multiple problems difficult. We plan to use a combination proof-problem properties weighted by student performance metrics to evaluate problem difficulty; however, we have not determined which combination of methods to use. We are currently looking into weighted cluster-based classification methods to apply to the problems. The hypothesis presumed before applying one of these methods would be that problems of similar difficulty would be placed into the same clusters. Student performance metrics for each problem could be used to determine distance, since it's assumed that students would react most similarly to problems of similar difficulty. Eagle et al. applied network community mining to this student log data in order to form interaction networks [2]; a modified version could be applied here on a student-per-problem level in order to determine prominent similar behaviors that are correlated with problem performance.

This would determine which problems are of similar difficulty, but not necessarily which problems (or groups of problems) are more or less difficult. That determination could be made by analyzing student rule scores across problems, or even the difference in scores at the start and end of a problem. In particular, analyzing the difference in rule scores would both standardize the scores (to account for the scores being calculated at different points in the tutor) and give a measure of forward or backward progress (a student's rule scores should not decrease after solving an easy problem).

Problem properties we feel are valuable to take into consideration when evaluating problem difficulty per student include:

- Classification of problems by operand/expressions
- Deviation of student solutions from expert solutions
  - Number of steps taken
  - Number and frequency of rules used

Student performance metrics that we feel are valuable to take into consideration include:

- Path progression through the tutor, including
  - Order of assigned proficiency sets
  - Number and path location of skipped problems
  - Terminus point in tutor
  - Final tutor grade
- Knowledge tracing scores for each rule, prioritized by problem requirements
- Step and elapsed time
- Type and number of errors committed

We would appreciate any literature recommendations, as well as suggestions for how to use the data from our experiment to measure and compare problem difficulty through student performance.

5. ACKNOWLEDGEMENTS

This material is based on work supported by the National Science Foundation under Grant No. 0845997.

6. REFERENCES

[1] M. J. Croy, T. Barnes, and J. Stamper. Towards an Intelligent Tutoring System for Propositional Proof Construction. In *Current Issues in Computing and Philosophy*, pages 145 – 155. 2008.

[2] M. Eagle, M. Johnson, and T. Barnes. Interaction Networks: Generating High Level Hints Based on Network Community Clusterings. In *Proceedings of the 5th International Conference on Educational Data Mining (EDM 2012)*, pages 164–167, 2012.

[3] T. Murray and I. Arroyo. Toward Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (EDM 2002)*, pages 289 – 294, 2002.

InVis: An EDM Tool For Graphical Rendering And Analysis Of Student Interaction Data

Vinay Sheshadri
North Carolina State University
Raleigh, NC
vshesha@ncsu.edu

Collin Lynch
North Carolina State University
Raleigh, NC
collin@pitt.edu

Dr. Tiffany Barnes
North Carolina State University
Raleigh, NC
tmbarnes@ncsu.edu

ABSTRACT

InVis is a novel visualization tool that was developed to explore, navigate and catalog student interaction data. InVis processes datasets collected from interactive educational systems such as intelligent tutoring systems and homework helpers and visualizes the student data as graphs. This visual representation of data provides an interactive environment with additional insights into the dataset and thus enhances our understanding of students' learning activities. Here, we demonstrate the issues encountered during the analysis of large EDM data sets, the progressive features offered by the InVis tool in order to address these issues and finally establish the effectiveness of the tool with suitable examples.

Keywords

EDM, visualization, graphs, student interaction data

1. INTRODUCTION

One of the central goals of Educational Datamining (EDM) is to translate raw student data into useful pedagogical insights. That is, educational dataminers seek to analyze student interaction data such as user-system logs with the goal of identifying: common errors, typical solutions and key conceptual challenges among other things. This research is of interest to learners, educators, administrators and researchers [17]. In recent years, the increased adoption of web-based tutoring systems, learning management tools and other interactive systems has resulted in an exponential increase in available data and increased demand for novel analytical tools. The Pittsburgh Science of Learning Center's DataShop, for example, currently stores over 188 datasets, encompassing 42 million student actions and 150,000 student hours [19]. With the increase in available data has come a corresponding increase in the insights EDM can provide and in making analytical tools available to expert instructors.

EDM researchers have generally relied on statistical analyses (see [14, 2, 1], formal rule induction (e.g. [12]), or other modeling methods to extract these insights. While these analytical methods are robust and have led to great progress in model development and evaluation, the increased interest in EDM by non-statisticians and practitioners has accentuated the need for "good visualization facilities to make their results meaningful to educators and e-learning designers" [16].

InVis was initially developed by Johnson, Eagle and Barnes [11]. The present version has been expanded to include changes to the visual editing system, export functions and other features. An example graph is shown in Figure 1. The graphical structure of InVis is designed to facilitate direct exploration of student datasets and easy comparison of individual solution paths. InVis can render individual student solutions or display the work of an entire class thus enabling educators to identify and draw insights from common student strategies and repeated mistakes [11]. InVis was inspired by the work of Barnes and Stamper [3] on the use of graphical representations for logic problems. Similar work has been done by Chiritoiu, Mihaescu and Burdescu who developed the EDM Visualization tool. This tool generates the student clustering models using k-means clustering algorithm [5]. However unlike InVis, the resulting visualization is non-interactive and non-graphical.

EDM researchers generally seek to answer questions such as: What actions can predict student success? Which strategy or solution path is more or less efficient and educationally effective? What decisions indicate student progress? And what are the features of a learning environment that promote learning? (see [15]). In a programming tutor, for example, students might be given the task of implementing an array-sorting algorithm for a large vector of integers. The particular choice of algorithm and the implementation details are left to the students to formulate using a variety of existing tools. This resulting code will proceed in several stages including reading data from disk, sorting the contents in memory, and returning the result. Our goal as researchers is to classify the successful students, identify the most commonly-chosen algorithms and flag individuals who faced difficulties or failed to complete the assignment. In a logic tutor such as Deep Thought [7] or a Physics tutor such as Andes [20] we would like to make similar determinations by focusing on the solutions chosen by the students and the individually-critical steps.

Published in CEUR-WS:
G-EDM workshop (Lynch and Barnes)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

The graph representation provided by InVis allows us to answer these questions by constructing and exploring interactive visualizations of the student dataset. By rendering a graph of a class or key subgroup (e.g. low-performing students), we can visually identify garden-path solutions over long isolated chains, identify critical states through which most students traversed and so on. These visualizations can also be used to guide, or evaluate the output of automatic analysis such as MDP models or path-detection algorithms. In the remainder of this paper we will discuss the tool, describe key features of it in detail and illustrate the type of insights it can provide.

2. DATA

We will illustrate the operation of InVis on a typical dataset. For the purposes of the present paper we will use student data collected from the Deep Thought tutor [6, 7]. Deep Thought is a graph-based tutor for first-order logic. Students using the system are presented with a problem defined by a set of given components (e.g. "$A \wedge \neg B \wedge C \Rightarrow B$") and are tasked with proving some goal state (e.g. $\neg C$). Problem solving proceeds through forward or backward-chaining with students applying rules such as Modus Ponens or Modus Tolens to draw new conclusions. For example, given the conclusion B, the student could propose that B was derived using Modus Ponens (MP) on two new, unjustified propositions: $A \rightarrow B, A$. This is like a conditional proof in that, if the student can justify $A \rightarrow B$ and A, then the proof is complete. At any time, the student can work backwards from any unjustified components, or forwards from any derived statements or the premises [8].

The DT data thus has a number of key characteristics that make it amenable to graphical display. The data is grouped into fixed problems covered by many students. Each problem is defined by a static set of given information and a clear goal. And the solutions are constructed via iterative rule applications drawn from a fixed library. As a consequence it is possible to define a fixed, albeit large, space of solution states and to efficiently map the traversal between them. While this seems restrictive this set of criteria applies to data collected from many if not most Intelligent Tutoring Systems. Andes, for example, defines problems by a set of given values (e.g. "$M_{car} = 2kg$") sets fixed variable goals (e.g. "S_{car-t0}": speed of the car at t_0) and groups student actions into a fixed set of rule applications. Similar state representations have also been applied to other datasets such as code-states in the SNAP programming tutor [4].

The figures shown below are drawn from two InVis datasets. We will focus in detail on a small dataset comparing the work of three students on a single problem with a fixed set of givens and two alternate goals. Such a small dataset is designed to allow for efficient illustration but is not an upper limit for analysis. We will also present some qualitative discussion of larger scale analysis with a larger DT dataset as shown in Figure 3.

3. FEATURES OF INVIS

InVis was developed with the *Java Netbeans Framework* and employs the *JUNG* libraries for the rendering of the graphs [13]. It provides an assortment of features that allow the end user to interact with the visualizations and draw observations from the data set. The *Network Display, Network Viewer, Visual Editor and Export Dot Data* are some of the prominent features of InVis which will be illustrated with examples in the upcoming sections. InVis also supports *MDP calculation, between-ness calculation and frequency reduction* which currently are under development and test phases.

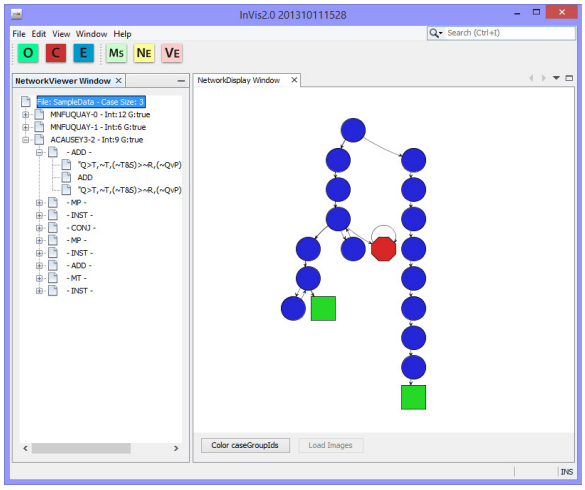


Figure 1: Network Display and Viewer

3.1 Network Display and Viewer

The front-end of InVis is the The *Network Display* component. It displays the interaction network generated by the engine in a graphical format. The user is presented with a cumulative overview of the processed input data. The various logic states of the DT tutor are represented by nodes and the applied propositional logic transformations are represented by edges of the graph. Intermediate states are represented by blue circular nodes while the goal states are represented by green square nodes. Error states in the DT dataset are defined by logical fallacies and are represented by red octagons for easy identification. The sample display shown in Figure 1 contains 16 intermediate nodes arrayed from the top to bottom of the network, one error state located in the center, and two goal states at the bottom.

The *Network Viewer* component represents the InVis input data in the form of a tree structure known as case-set. Each primary node in the case-set represents a student and each sub-node under it represents a transition state executed by the student sequentially. Selecting a student in the Network Viewer window highlights the corresponding path in the Network Display window. Selecting a sub-node highlights the corresponding nodes and edges that were involved in the transformation. Expanding a sub-node will cause the system to display the pre-state and post-state information from the nodes involved in that transition.

The path taken by a student to solve the given problem can be detected by selecting the appropriate student in the Network Viewer window. This will fade the non-path nodes to bring the chosen path to the foreground. An example of this highlighting is shown in Figure 2 where we have selected a single student path within the demo dataset.

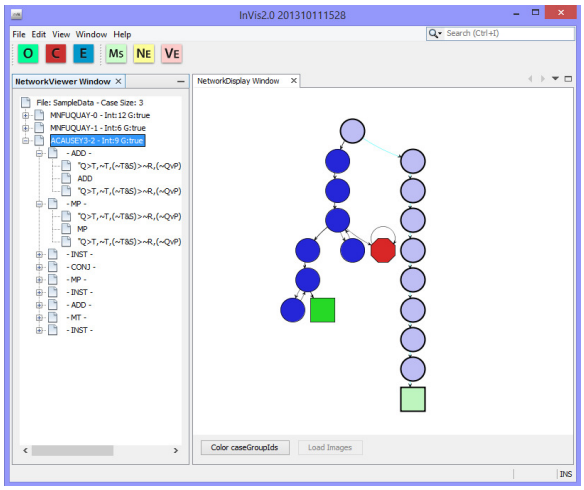


Figure 2: Tracing the path of a student

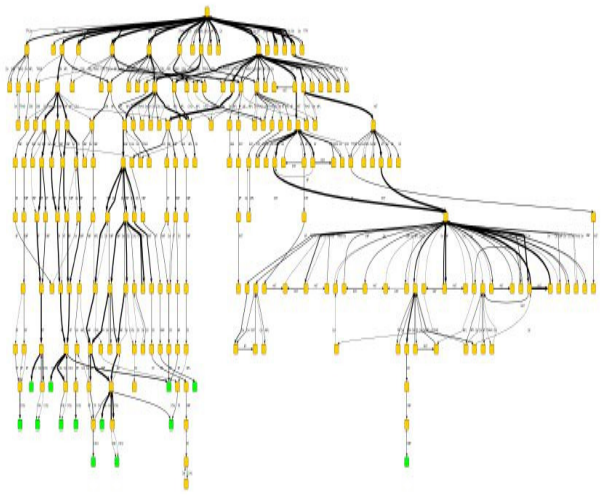


Figure 3: InVis and large data sets

One common use of InVis is to identify frequently-occurring error states. The system can also be used to analyze the different paths taken by students in order to achieve a common goal and isolate the areas where the students face difficulties in solving the given problem or took a garden path. A garden path is an inefficient path from one target state to another with many nonessential intermediate states. From Figure 1, in the current data set, for example, one student performed 11 transitions to achieve the goal, due in part to cycles, whereas a separate student reached the goal with 5 transitions. Each transition is marked by an arc from one state to another in the graph. Thus the Network Display provides an instructor with a cumulative analysis of the input data and aids the instructor in identifying areas of difficulty faced by students during the course of problem solving.

Figure 3 shows the visualization generated by InVis for a sample large dataset. The bold edges indicate the common paths employed by the students in order to solve a given problem. The graph also highlights the garden paths and the succeeding action taken by students towards achieving the goal states. From the rendered visualization it is clear that the cloud space comprises of students who achieved the goal, indicated in green and students who failed to reach the final goal states. InVis can thus be employed to congregate useful observations on large EDM datasets.

3.2 Visual Editor

The *Visual Editor* component of InVis controls the various visual aspects of the graph displayed in the Network Display window. The visual editor provides options for displaying the node and edge data of the graph. InVis renders graphs with the DAG tree layout as the default layout. The visual editor provides options for rendering the graph in different layouts. An ISOM layout of the originally generated graph is shown in Figure 4.

The Visual Editor also provides an option for normalizing the edge widths based on the case frequencies. Case frequencies are defined by the number of students who used the same transition between the given set of states. When the *Normalize Width* option is selected, InVis reloads the graph with width of edges proportional to the case frequency. This feature helps instructors in identifying the logic states and transitions which are most used by the students.

The Visual Editor can be launched by clicking on the Visual Editor icon in the toolbar. Options are provided in the Visual Editor window to control the display of node and edge labels. A notable option provided by the visual editor is the option to normalize edge widths. Normalizing edge widths results in the modification of the edge widths of the graph in proportion to the case frequencies.

Figure 5 displays the zoomed in version of the graph with normalized edges. Edges with case frequency of 2 have thicker connecting lines compared to the edges with case frequency of 1. Thus the thickness of the edge offers a visual cue to the instructor in identifying the most commonly traversed paths by students when achieving the given goal.

3.3 Exporting InVis Data

Graphviz is a heterogeneous collection of graph drawing tools [9]. The software is available under open source license. The input to the Graphviz tool is a description of the required graph in a simple text language such as DOT. The tool processes the input and renders output graphs in useful formats, such as images and SVG for web pages; PDF or Postscript for inclusion in other documents; or display in an interactive graph browser [10]. Graphviz has many useful features for concrete diagrams, options for colors, fonts, tabular node layouts, line styles, hyperlinks, and custom shapes.

In order to leverage the graph design features offered by Graphviz, InVis now features a new export option which

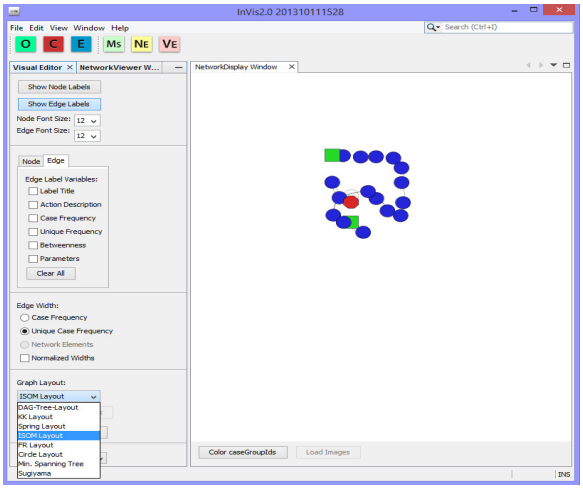


Figure 4: Different graph layouts

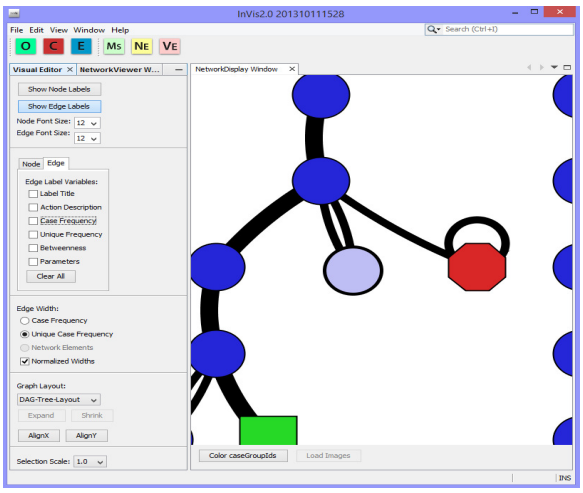


Figure 5: Normalized width - Zoomed in

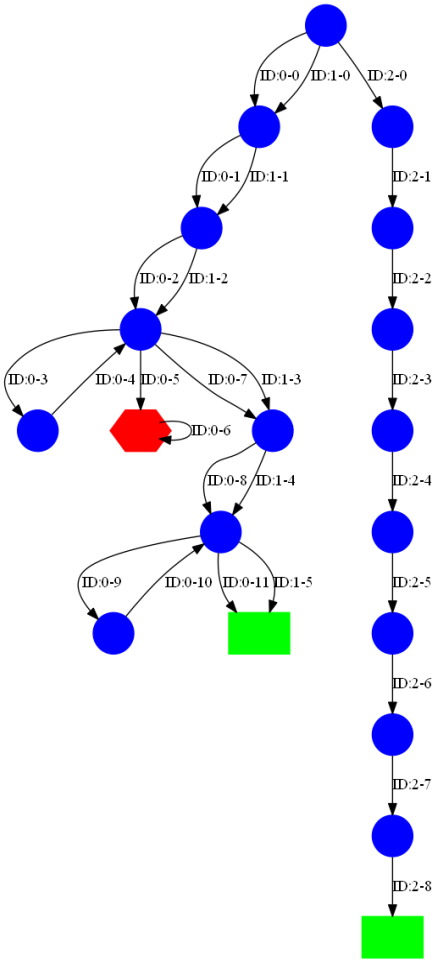


Figure 6: Exported data loaded in Graphviz

renders the input Deep thought data into a DOT format file. The DOT file can be directly imported by Graphviz to generate static images such as PNG, JPEG or interactive formats such as SVG. These visualizations will match those generated by the Network Display tool. Figure 6 shows a graph generated by Graphviz using exported InVis data. Here the arcs are annotated via a static ID number that helps in manually identifying the states and transition information. This data is captured as part of the export process.

4. DISCUSSION

The graphical rendering of EDM data via InVis can yield unique insights into the student interaction data. Romero and Ventura classified EDM objectives depending on the viewpoint of the final user as learner, educator, administrator and researcher [17]. InVis supports learners by providing visual feedback and recommendations to improve performance. Students can compare their approach with that of other students graphically. This can promote real time self-assessment and adoption of better approaches to problem solving.

Educators can use the tool to identify good and poor student solutions and to better understand the students' learning processes which can, in turn, reflect on their own teaching methods. The graphical summary presented by InVis gives an overview, and allows for detailed exploration of, the paths taken by students in achieving a solution to a given problem.

The presence of garden paths, loops and error states illustrate areas where the students have encountered difficulties in deriving a solution to a given problem. This empowers researchers with visual data to model suitable hint generation techniques that can deploy automatic corrective actions [18]. InVis can assist administrators to reorganize institutional resources based on visual evaluation of the effectiveness of a teaching method adopted in a particular course.

In the case of the sorting example introduced in the earlier section, by normalizing the edge width, we can identify the most commonly used sorting algorithm. We can also identify the optimal solution to the given problem comparing the number of transition states between the start and end goal

for each student. Finally the presence of error states, garden paths can be visually identified and corrective actions can be taken to aid students in achieving the goal. Thus the visualizations help in the generation of real time feedback and provides hints for modeling of dynamic hint generation strategies.

InVis is currently limited to the analysis of deep thought tutor data. We are actively working on InVis to extend its capabilities to analyze data sets generated from fields such as: state based games, feedback back based hint generation and others. We are also actively improving the efficiency, user interface, and automatic analysis features of the tool. The InVis project provides the EDM community with a visualization tool for enhanced and accelerated understanding of education based systems. New features will be added to InVis in future to support and sustain this goal. We solicit the EDM community to provide us with additional suggestions for, the InVis tool and help us to enhance the functionality and usability of InVis for EDM applications.

Acknowledgments

This work was supported by NSF-IIS 0845997 "CAREER: Educational Data Mining for Student Support in Interactive Learning Environments" Dr. Tiffany Barnes PI.

5. REFERENCES

[1] R. Baker, A. Corbett, I. Roll, and K. Koedinger. Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction*, 18(3):287–314, 2008.

[2] R. Baker and K. Yacef. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Datamining*, 1(1):3–17, 2009.

[3] T. Barnes and J. Stamper. Toward the extraction of production rules for solving logic proofs. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop*, AIED2007, pages 11–20, 2007.

[4] A. H. Barry Peddycord III and T. Barnes, editors. *Generating Hints for Programming Problems Using Intermediate Output*. International Educational Datamining Society IEDMS, 2014. In Press.

[5] M. S. Chirioiu, M. C. Mihaescu, and D. D. Burdescu, editors. *Students Activity Visualization Tool*. International Educational Datamining Society IEDMS, 2013.

[6] M. J. Croy. Graphic interface design and deductive proof construction. *Journal of Computers in Mathematics and Science Teaching*, 18(4):371–385, 1999.

[7] M. J. Croy. Problem solving, working backwards, and graphic proof representation. *Teaching Philosophy*, 2(23):169 – 187, 2000.

[8] M. J. Eagle and T. Barnes. Evaluation of automatically generated hint feedback. *EDM 2013*, 2013.

[9] J. Ellson, E. Gansner, L. Koutsofios, S. North, and G. Woodhull. Graphviz - open source graph drawing tools. In P. Mutzel, M. JÃijnger, and S. Leipert, editors, *Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*, pages 483–484. Springer Berlin Heidelberg, 2002.

[10] E. Gansner, E. Koutsofios, and S. North. Drawing graphs with dot. Technical report, Technical report, AT&T Research. URL http://www. graphviz. org/Documentation/dotguide. pdf, 2006.

[11] M. W. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks. *Proc. EDM 2013*, 65, 2013.

[12] C. F. Lynch, K. D. Ashley, N. Pinkwart, and V. Aleven. Argument graph classification with genetic programming and c4.5. In R. S. J. de Baker, T. Barnes, and J. E. Beck, editors, *EDM*, pages 137–146. www.educationaldatamining.org, 2008.

[13] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y.-B. Boey. Analysis and visualization of network data using jung. *Journal of Statistical Software*, 10(2):1–35, 2005.

[14] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. In V. Dimitrova, R. Mizoguchi, B. du Boulay, and A. C. Graesser, editors, *AIED*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.

[15] A. Pena-Ayala. *Educational Data Mining: Applications and Trends*. Springer, 2014.

[16] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Syst. Appl.*, 33(1):135–146, July 2007.

[17] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[18] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.

[19] J. Stamper, K. Koedinger, R. S. J. d. Baker, A. Skogsholm, B. Leber, J. Rankin, and S. Demi. Pslc datashop: A data analysis service for the learning science community. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part II*, ITS'10, pages 455–455, Berlin, Heidelberg, 2010. Springer-Verlag.

[20] K. Vanlehn, C. Lynch, K. Schulze, J. A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, and M. Wintersgill. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.

Workshop on Non-Cognitive Factors & Personalization for Adaptive Learning (NCFPAL)

Many computer-based learning environments adapt to individual learners based on cognitive factors like skill mastery, but recently research has been increasingly directed at improving personalization and adaptation in such systems by harnessing non-cognitive factors such as learner affect, motivation, preferences, self-efficacy, self-regulation, and grit. This workshop brings together researchers studying non-cognitive factors in a variety of environments and contexts, using various experimental, measurement, and/or data mining and statistical methods. In addition to presenting on-going research on specific non-cognitive factors and their impact of learning outcomes, speakers at the workshop will present various creative approaches to address methodological issues endemic to research on non-cognitive factors.

Of one invited paper and five accepted papers, three papers explore non-cognitive factors in intelligent tutoring systems (ITSs) used in K-12 schools. Walkington and collaborators, in an invited paper, provide an account of various text-based features of mathematics word problems that are associated with learner performance in ITSs (specifically, Carnegie Learning's Cognitive Tutor). While explanations that point to both cognitive and non-cognitive factors may account for this association, Bernacki and Walkington follow up this observational study by exploring an intervention in the same ITS wherein word problems are personalized based on learners' out-of-school interests in areas like sports and music and find that personalization has benefits for both learner interest and measures of learning. A third study by Ostrow and colleagues considers an intervention in the ASSISTments system in which learners were presented with different types of "growth mindset" motivational messages (e.g., animations, audio, etc.). The impact of these messages on measures like persistence and learning are considered.

The next three papers consider data from college-level courses and learners. Ezen-Can and Boyer present an unsupervised method for classifying dialogue acts (e.g., ask a question, give a command) when learners interact with (human) tutors in a text-based dialogue environment; their method leverages gender and learner self-efficacy as noncognitive factors along which sub-populations of learners can be identified so that dialogue acts can be better classified. Next, Moretti and colleagues mine data about university computer science courses that are publicly available on the web to determine factors (e.g., choice of programming language and grading criteria) that are associated with learner feedback and other aspects of instruction. Finally, Gray and colleagues provide an analysis, using both classification and regression methods, of various psychometric measures of non-cognitive factors as predictors of whether students are "at risk" or likely to fail in their university courses.

The papers that comprise these proceedings represent a diverse set of measurement and analytical approaches and of student populations and learning platforms to which they are applied. We take this as a sign of developments to come, especially as researchers and developers in the learning sciences, educational data mining, and learning analytics increasingly turn to non-cognitive factors as possible "levers" to adapt and personalize learning experiences in more and more sophisticated technology-enhanced learning platforms and environments.

We gratefully acknowledge the following members of the workshop program committee:

Vincent Aleven, Carnegie Mellon University
Ryan S.J.d. Baker, Columbia University
Matt Bernacki, University of Nevada, Las Vegas
Alan Drimmer, Apollo Group, Inc.
Andrew Krumm, SRI International
Timothy Nokes-Malach, University of Pittsburgh
John Stamper, Carnegie Mellon University
Candace Walkington, Southern Methodist University
Michael Yudelson, Carnegie Learning, Inc.

The NCFPAL workshop organizers
Steven Ritter
Stephen E. Fancsali

Table of Contents NCFPAL

The Impact of Cognitive and Non-Cognitive Text-Based Factors on Solving Mathematics Story Problems

Candace Walkington
Southern Methodist University
3011 University Blvd. Ste. 345
Dallas, TX, 75205
1-214-768-3072
cwalkington@smu.edu

Virginia Clinton
University of Wisconsin - Madison
1025 West Johnson Street
Madison, WI 53706
1-608-890-4259
vclinton@wisc.edu

Steven Ritter
Carnegie Learning, Inc.
437 Grant Street, Suite 918
Pittsburgh, PA 15219
1-888-851-7094 x122
sritter@carnegielearning.com

Mitchell Nathan
University of Wisconsin - Madison
1025 West Johnson Street
Madison, WI 53706
1-608-262-0831
mnathan@wisc.edu

Stephen E. Fancsali
Carnegie Learning, Inc.
437 Grant Street, Suite 918
Pittsburgh, PA 15219
1-888-851-7094 x219
sfancsali@carnegielearning.com

ABSTRACT

Intelligent tutoring systems (ITSs) that personalize instruction to individual learner background and preferences have emerged in K-16 classroom settings all over the world. In mathematics instruction, ITSs may be especially important for tracking mathematical skill development over time. However, recent research has pointed to the importance of text-based measures when solving mathematics word problems, suggesting that in order to accurately model the student it is important to understand how they respond to text characteristics. We investigate the impact of text-based factors (readability and problem topic) on the solving of mathematics story problems using a corpus of N = 3394 students working through an ITS for algebra, Cognitive Tutor Algebra. We leverage recent advances in computerized text-mining to automate fine-grained text analyses of many different word problems. We find that several elements of the text of mathematics word problems matter for performance – including the concreteness of the problem's topic, the length and conciseness of the story's text, and the words and phrases used.

Keywords

Intelligent tutoring system, readability, mathematics, word problems, personalization

1. INTRODUCTION

Since the 1980s, Intelligent Tutoring Systems (ITSs) have risen as an important instructional tool to support student learning in classrooms, especially in middle and high school. ITSs typically consist of at least three components: (1) the *domain model* of the appropriate steps needed to correctly solve each problem, (2) the *student model*, which captures the evolution of an individual student's cognitive states as they relate to the domain model, and (3) the *tutoring model* which selects tutor actions based on the domain model and student model [1]. It is through the construction of the student model and its contribution to the tutoring model that ITSs can enact *personalization* where they adapt to the needs and backgrounds of individual learners. Here we explore cognitive and non-cognitive factors related to how students react to and understand the text of mathematics story problems. We argue that these non-mathematical factors may be an important element to consider for an ITS in secondary mathematics. In particular, we provide evidence suggesting that both the students' reading level (a cognitive factor) and the students' interests, preferences, and motivational outlooks (non-cognitive factors) have the potential to influence how they respond to text-based mathematics problems situated in "real world" contexts.

Cognitive Tutor Algebra (CTA; [2]) is a prominent mathematics ITS used in many schools across the United States. CTA uses *model-tracing approaches* to relate student actions to the domain model and provides individualized error feedback. CTA also uses *knowledge-tracing approaches* to track students' learning from one problem to the next, using this information to identify the students' strengths and weakness in terms of production rules (i.e., knowledge components or skills). The software then uses this analysis to individualize the selection of problem tasks. However, missing from this tutoring model is a consideration of other non-mathematical characteristics of the story problem texts – including the *reading difficulty* of the text respective to students' reading ability and preferences, and the *real-world topic* of the text respective to students' interests and preferences.

For example, a learner presented with a mathematics word problem that is difficult to read – with high-level vocabulary, complex sentence structure, etc. - may lack the reading ability to appropriately comprehend that problem. This cognitive element of the problem's difficulty is not typically monitored by ITSs for mathematics learning. In

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

addition, such a problem may inhibit the students' motivation – a non-cognitive factor. In particular, even if the learner is technically able to read the problem, they may be intimidated by the problem text, and request a hint instead of putting forth the effort of understanding the text of the problem. ITSs also do not typically monitor the learner motivation for reading and understanding text-based problems.

Another non-mathematical element of the text of mathematics story problems is the real world topic – whether the story is about working at a part-time job or harvesting a field of grain. The way in which students react to the topic of the story problem is also based on both cognitive and non-cognitive factors. Students may be unfamiliar with elements of the context that are important for fully comprehending the problem – for example, in a banking context, they may not know what "break even" means. In this way, they may lack the prior knowledge needed to interpret the story. Similarly, different real world topics may differ in the motivation they elicit from students – students may experience greater motivation when solving a problem about a familiar, interesting context than about a context they find boring or unfamiliar.

We next provide a theoretical framework that provides an explanation of how students comprehend story problems and how cognitive and non-cognitive factors may interact as they solve story problems.

2. THEORETICAL FRAMEWORK

2.1 Cognitive Factors

Nathan and colleagues [3] proposed a model of mathematics story problem solving where students navigate three levels of representation as they comprehend and solve story texts: (1) a *textbase* containing the propositional statements made in the story problem, (2) a *situation model*, a qualitative representation of the actions and events in the story, and (3) a *problem model*, containing the formal mathematical equations, variables, and operands. Because mathematics word problems are stated in verbal language (rather than mathematics notation), we hypothesize that the reading difficulty and topic of the problem matters for the construction of the situation model and its successful coordination with the problem model.

Various aspects of the reading difficulty, including *readability measures*, may be important in situation model construction. Readability measures often include the kinds of words used, the length of the story, and the structure of the sentences. These elements of the text's structure may make it more difficult to comprehend, especially for students with weaker reading skills.

Another aspect of reading difficulty is the *topic* of the problem – whether it is about, for example, farming or banking. Walkington and colleagues [4] proposed that story contexts that are related to topics that are familiar and accessible to students are easier for them to solve because these contexts can facilitate situation model construction because of their relatedness to learner prior knowledge. In related work [5], they also identified the prevalence of issues with verbal interpretation of mathematics story problems, finding that even high school students struggle to understand difficult vocabulary words and construct an accurate propositional textbase and situation model from a story problem's text.

2.2 Non-Cognitive Factors

An important precursor to students' motivation is their level of *interest* – defined as the state of engaging and the predisposition to re-engage with particular topics, ideas, or activities [6]. Two types of interest have been described in the literature. First, *situational interest* is an immediate, temporary state of heightened attention and affective engagement that stems from elements of a learning environment that are surprising, salient, evocative, challenging, personally relevant, etc. Situational interest can be *triggered* in response to a stimuli within a learning environment, and then may or may not become *maintained* over time [6]. A second type of interest is *individual interest* – learners' enduring predispositions to engage with certain activities or topics over time.

Elements of a story problem's text have the potential to both trigger and maintain situational interest. In particular, story problems that are accessible, easy to read, and situated within the topics and contexts that a particular learner finds relevant and interesting may trigger and maintain interest. In the other hand, difficult reading passages disconnected from a learner's experiences and interests may not trigger interest and may cause disengagement if interest has previously been triggered.

2.3 Research Purpose

If text-based measures like readability and problem topic matter for student performance, these might be important elements to add to future systems for personalized learning in mathematics. For example, an ITS might present weak readers with problems with simplified verbal language as these learners are initially mastering a new mathematical skill. As the student gains expertise with the mathematics by mastering skills, additional levels of verbal difficulty could be layered on by the ITS. Similarly, learners that lack motivation may be presented with story problems that are less intimidating to read and situated within their interests, with this support faded out over time. By neglecting to model this aspect of the user's experience in the ITS, the system may be generating inferences about learner knowledge states that are inaccurate.

3. LITERATURE REVIEW

3.1 The Impact of Reading Difficulty on Solving Mathematics Story Problems

Recent research has found that reading ability is especially important as students solve mathematics word problems [7]. Studies examining the association of reading difficulty of mathematics word problems and U.S. student

performance on large-scale assessments has found that problems that use words with multiple meanings, complex verbs, and mathematics vocabulary words are more difficult [8]; the effect is especially pronounced for students who speak English as a second language [9]. A small study of students working in CTA found that extraneous text that provided a real world context for the problem, as well as references to concrete people, places, and things, were associated with less concentration and more confusion in the tutor [10]. However, a similar study found that the extraneous text was also associated with fewer unproductive "gaming the system" behaviors in the tutor [11]. Converging evidence suggests text characteristics relating to reading difficulty are important when solving mathematics word problems, but studies are needed that address which elements of reading difficulty are most important.

3.2 The Impact of Problem Topic on Solving Mathematics Story Problems

The topic of mathematics story problems also has an important relationship to students' prior knowledge and motivation. A study of high school students solving either standard story problems or story problems personalized to topics they were interested in (e.g., sports, video games, social networking) within one unit of CTA found that personalized stories were associated with higher performance. This performance gain was present in two tasks – labeling independent and dependent quantities given in algebra story problems, and writing algebraic expressions from the story scenarios [12]. It was hypothesized that during these two tasks, students are working closely with the problem text, constructing their situation model and coordinating it with a problem model. This study also found that students receiving problems in the context of their out-of-school interests were less likely to game the system – to exploit regularities in hints and feedback provided by CTA in order to avoid productive learning behaviors. Further, students who received personalization had stronger performance in future units where the problems were no longer personalized.

In a recent follow-up study [13], story problems in four units of CTA were personalized to topics students were interested in, and students solving personalized problems were compared to a control group solving normal problems. Results showed that personalized problems both triggered students' situational interest and enhanced students' individual interest for learning algebra. Personalization was associated with greater learning gains than a control condition only when the personalization was matched to deep features of the students' interest area. This was contrasted with personalization that was only matched surface features of the learners' interests – i.e., modifications to the problems that simply involved inserting familiar pop-culture words rather than considering how learners might actually use relationships between quantities in their everyday activities. Thus converging evidence points to the importance of considering the real world topic of mathematics story problems and its relationship to students' interests and experiences. However, more research is needed to determine which topics may be more or less likely to trigger and maintain students' interest.

3.3 Research Questions

In the present study, we investigate the relationship between readability and topic measures and student performance on mathematics story problems. We examine these issues within an ITS for Algebra I, Cognitive Tutor Algebra (CTA), that tracks student hint requests in addition to whether they get problems correct or incorrect. We investigate two research questions: (1) How are readability and topic measures associated with correct answers and hint requests when students label independent and dependent quantities in stories in CTA? (2) How are readability and topic measures associated with correct answers and hint requests when students write algebraic expressions from stories in CTA? Answers to these questions could inform the design of future ITSs for personalized instruction.

4. METHOD

Data from N = 3394 students with active CTA accounts were collected from 9 high schools and 1 middle school that were diverse in terms of their socio-economic, racial, and achievement background (Table 1). Data were collected for students solving 151 distinct word problems accross the first 8 units of CTA; later units were not included because many students did not advance beyond these units. We collapsed for all analyses (i.e., treat as identical) problems containing an identical story but using slightly different numbers. On average, each problem had been solved by 742 students (SD = 495). Each problem included a story scenario that outlined one or more linear functions within a real world situation (Figure 1). The student was asked to complete steps in which they identified the independent and dependent quantities in the story, wrote a linear algebraic expression for the story, and solved their expression for different x and y values; we consider only the first two skills.

CTA log data from students in the selected schools were uploaded to DataShop (pslcdatashop.web.cmu.edu), an online repository of detailed student interaction data. These logs contained information on whether the student got each problem correct, incorrect, or requested a hint on their first attempt; because requesting a hint is a distinct outcome, correct and incorrect are not completely repetitive measures. Thus, for each problem, we compiled the percentage of students who had gotten the problem correct on the first attempt, incorrect, or requested a hint. This percentage was our dependent measure in three distinct regression models.We analyzed the text of the introduction to each story problem (i.e., the initial text that gives the linear rate of change and intercept; see Figure 1) with the *Coh-Metrix* and *LIWC* text-mining programs. Coh-Metrix

[14] measures a large number of aspects of text readability, including the amount semantic overlap between sentences, the number of verbs, use of concrete versus abstract words, the average sentence length, and others.

Table 1. Demographic characteristics of schools in study

ID	Math Prof %	State Prof %	School Enrollment	School Type
1	88%	70%	797	Middle
2	81%	47%	1,482	High
3	95%	84%	2,163	High
4	55%	46%	708	High
5	27%	NA	1,875	High
6	68%	59%	986	High
7	2%	31%	602	High
8	76%	84%	1,333	High
9	19%	39%	397	High
10	68%	79%	800	High

ID	White	Black	Hispanic	F/R Lunch
1	72%	7%	15%	21%
2	90%	4%	2%	4%
3	84%	10%	3%	6%
4	99%	1%	1%	41%
5	20%	4%	72%	77%
6	9%	2%	88%	41%
7	1%	99%	1%	82%
8	36%	60%	2%	48%
9	100%	0%	0%	45%
10	38%	51%	11%	62%

Because some of our story introductions had only one sentence, measures that pre-supposed multiple sentences were ommitted. LIWC [15] was used to determine the topic of the story problems – this program counts how many words in the story fall into various word categories, including social processes (family, friends, people), affective processes (positive emotions and negative emotions), biological processes (body, health, ingestion), cognitive processes (insight, causation, discrepancy, tentativeness, certainty, inhibition, inclusive/exclusiveness), perceptual processes (see, hear, feel), relativity processes (motion, space, time), and personal concerns (work, achievement, leisure, home, money, religion). If a story contained any words that fell into one of these topic categories, that story was coded as a 1 for that category; otherwise it was coded as a 0.

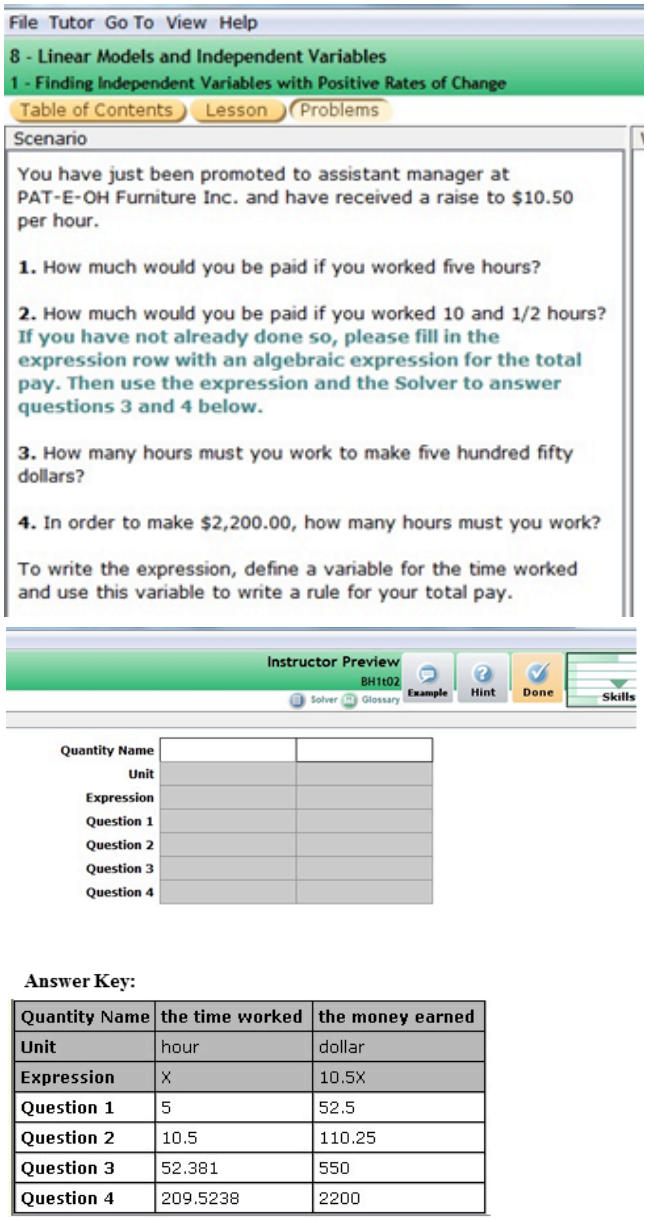


Figure 1. Screenshot of algebra story problem in CTA with answer key superimposed

For each category in Coh-Metrix and LIWC, the correlation was computed between the list of each problem's score on that category, and the percentage of students who got each problem correct, incorrect, or requested a hint. Correlations that were significantly different from 0 were tested for inclusion as fixed effects in regression models predicting the performance measures (hints, corrects, incorrects). These models included random effects that described various aspects of the problem's mathematical structure, including the unit and section it came from in CTA, and the numbers it used. Models were initially fit using the *lmer()* command in *R* including all potential fixed and random effects. Then we used the *step()* command in *R* to perform backwards elimination on fixed

and random effects, leaving a model with only the effects that significantly improved the fit of the model. These analyses were carried out separately for a dataset that included only instances of students labeling independent and dependent quantities, and a dataset that included only instances of students writing algebraic expressions.

5. RESULTS

5.1 Labeling Independent and Dependent Variables

Regression results showing the relationship between performance measures (% incorrect, hint, and correct) and readability and topic measures for labeling quantities in story problems are provided in Table 2. Table 2 shows that problems that use adverbial phrases (*DRAP*) were associated with fewer incorrect answers. Adverbial phrases are phrases that add on to verbs, answering the questions where, when, or how? In the present data set, adverbial phrases mostly answered when the action occured, and often included words like *currently*, *already*, *next*, *first*, *every day/week*, and *not yet*. However, some of these adverbs also answered the how question, relying information about quantities that might be useful to cue students to the constraints of the problem – examples of words used in this manner included *only*, *completely*, and *evenly*. These words may have given important details about how the quantities involved in the story were changing as the action in the story proceeded.

Table 2. Regression tables relating performance measures on labeling quantities to readability/topic categories

	Estimate	Std. Err	t value	Pr(>|t|)	
% Incorrect					
(Intercept)	0.182	0.032	5.63	0.00018	***
DRAP	-0.0008	0.0003	-2.34	0.02104	*
motion	0.036	0.0137	2.65	0.00899	**
% Hint					
(Intercept)	0.045	0.014	3.21	0.01407	*
inhibition	0.023	0.008	2.83	0.00543	**
% Correct					
(Intercept)	0.784	0.044	17.87	0.00000	***
motion	-0.042	0.0182	-2.29	0.02370	*

Stories that involve *motion words* (e.g., *go*, *move*, *ran*, *arrive*, *come*, *enter*, *threw*) are associated with more incorrect answers and fewer correct answers. These stories often incldued contexts where people were walking, biking, hot-air-balooning, driving, or actively constructing something. In terms of the quantities used, there was often a rate of change (e.g., per hour, per minute, a day) that involved this motion, and students had to identify the two quantities that made up this rate of change. Using more abstract physics quantities – like distance and speed – may have been more difficult for students than using quantities relating to specific concrete objects (e.g., accumulating cards, toys, or money). Finally, *inhibition words* were associated with more hint requests. Inhibition words were often included in story problems that discussed safety issues or saving money. Students may have persieved these less concrete, finance- or safety-oriented contexts as less accessible, making them more likely to request a hint rather than attempt to write the labels. These problems often involved money as the dependent variable, but the label for this variable may have been complex because the actor in the story might have already saved or spent some money when the story started. Thus a label of simply *money* may not be appropriate, and the student would have to enter a label that captured that it was *total money* or *net money* saved or spent.

5.2 Writing the Algebraic Expression

Regression results showing the relationship between performance measures and readability and topic measures for writing the expression are shown in Table 3. We again see that *inhibition words* – often associated with financial contexts – are more difficult for students – they are associated with more incorrect answers, more hint requests, and fewer correct answers. The conceptual difficulty of this topic area might become especially important as students move from formulating their situation model to coordinating their situation model with a problem model.

Table 3. Regression tables relating performance measures on writing expressions to readability/topic categories

	Estimate	Std. Err	t value	Pr(>|t|)	
% Incorrect					
(Intercept)	0.195	0.060	3.26	0.00167	**
WRDPOLc	0.0494	0.013	3.91	0.00014	***
inhibition	0.086	0.034	2.52	0.01286	*
% Hint					
(Intercept)	0.055	0.014	3.95	0.00050	***
One sentence	(ref.)				
Two sentences	-0.045	0.016	-2.82	0.00548	**
Three Sentences	-0.057	0.017	-3.48	0.00067	***
4 + Sentences	-0.033	0.019	-1.77	0.07868	
RDL2	0.002	0.001	3.51	0.00061	***
family	0.030	0.015	2.05	0.04282	*
inhibition	0.052	0.011	4.74	0.00001	***
motion	0.025	0.009	2.77	0.00637	**
% Correct					
(Intercept)	0.334	0.17478	1.91	0.05778	
LDTTRc	0.428	0.169	2.53	0.01242	*
WRDPOLc	-0.041	0.01469	-2.78	0.00609	**

inhibition	-0.128	0.03909	-3.28	0.00132	**

Another factor that stands out in the regression results is word polysemy (*WRDPOLc*) – or the number of different meanings that a word has (for example, in English, *mine* can be something you own or an explosive device). The results show that stories that contain words with more potential meanings are associated with more incorrect answers and fewer correct answers. Polysemous words have been found to make mathematics word problems more difficult to interpret accross other studies [8-9].

Results also showed that higher type-token ratios (*LDTTRc*) are associated with more correct answers. As type-token ratio increases, more unqiue words are being used in the story problem, and fewer words are being repeated. These results suggest that students have an easier time writing the expression in a story that is relatively concise with little reptition of ideas. While it makes sense that this type of story may be more amenable to translation into mathematics notation, this result contrasts with research in text comprehension in reading tasks [14] which generally finds that repitition and lower type-token ratios facilitate reading comprehension. However, the story problems with high levels of word repetition frequently discuss complex topics of which students may lack familiarity, including operating capital, business inventory, and wholesale prices. In this way, a high type-token ratio may be indicative of a complex topic rather than increased readability in these story problems.

Students‘ tendency to seek hints when writing the algebraic expression is associated with a number of different readability factors. First, we see an effect for the length of the story text; students are more likely to seek hints for *one sentence story problems*, compared to problems that have two or more sentences. Having only one single sentence in a story problem might not be enough to ground or fully describe a linear rate of change as it arises in a real-world situation, and these overly-sparse stories might consequently inhibit performance.

In addition to greater difficulty of inhibition words, stories with *family words* and *motion words* were associated with greater hint-seeking. Only 13 of the problems involved family words, and these were often complex scenarios where multiple actors (e.g., a main character and his brother) were each contributing to the algebraic rate of change in their own way (e.g., saving/earning/splitting money together). Motion words often involved physics contexts (e.g., traveling in a car or plane) in which students had to track distance, rate, and time. This suggests that keeping track of multiple individuals engaging in mathematical actions and solving problems with physical distances and rates may be significant difficulty factors when writing expressions.

Finally, the regression results showed that scoring higher on Coh-Metrix’s second language readability measure (*RDL2*) was associated with greater hint-seeking when writing expressions. This measure is calculated through measures of word frequency (with words that occur more frequently in the English language yielding higher scores), sentence syntax similarity (with sentences that have similar grammatical structures yielding higher scores), and word overlap (with words that share semantic meaning yielding higher scores; [16]). Given that a higher second language readability score is typically associated with greater ease in comprehending the text [17], it is suprising that stories that score higher on this measure would be associated with students seeking more hints. The explanation of this finding may be similar to that for our finding with type-token ratio; story problems that use similar words and sentence structures often use a lot of reptition as a way to present complex ideas. Stories that are simple and concise may be easier for students to solve.

6. DISCUSSION

Results indicate that readability and topic measures have important associations with students‘ performance when solving mathematics word problems in an ITS. In particular, it was more difficult for students to name the independent and dependent quanitities in problems relating to motion (physics) and inhibition (saving and safety), while adverbial cues facilitated this skill. When writing algebraic expressions, we again see that motion and inhibition topics are difficult, but also find other important readability measures that matter. Words with multiple meanings make story problems more difficult, which corresponds to previous findings in both mathematics and reading education.

However, mathematics stories that use concise language with little repitition, which in terms of their readability level makes them technically *less* readable, are actually easier for students to solve. Thus measures of readability that stem from research on reading comprehension may need to be considered differently when working with mathematics problems. Results also suggest that while a story problem that includes only a single sentence is concise, it might present difficulty for students by not providing necessary context and information for them to feel they can respond without needing a hint.

Overall, our results suggest that mathematics story problems that have story texts that are more accessible to students have several characteristics: (1) they are concise with little repetition, but not a single sentence only, (2) they use only a single actor performing actions, (3) they use simple words with clear meanings, (4) they avoid more abstract physics or financial contexts, instead focusing on familiar contexts involving accumulation or loss of concrete physical objects, and (5) they make use of adverbial cues. Story problems with these characteristics may allow students to more easily construct a situation model from a propositional textbase. They may promote situation-model construction by both increasing students‘

ability to comprehend the semantics of the problem, and by increasing students' interest in working on the problem.

7. CONCLUSION

Future adaptive ITSs will be designed to model student characteristics at an extremely fine-grained level, as technology for personalized learning continues to advance. Here we argue that an important element of these future adaptive systems will be a consideration of the non-mathematical text-based characteristics of the problem tasks they present to students. Making inferences about students' current level of mathematical knowledge or motivation without considering these characteristics may lead to misspecifications.

Readability and topic measures may be an important consideration for ITSs to model in a variety of domains, including when considering tasks from history, social studies, and science. Future research should focus on the readability and topic measures that are most important for students of different age groups in different subject domains, and narrow down which characteristics are most critical to include in student and domain models as we build future ITSs. In current work, we are analyzing the mathematics problems on the National Assessment of Educational Progress (NAEP) and Trends in International Mathematics and Science Study (TIMSS) to examine how readability and topic measures impact the performance of 4th and 8th graders in the United States, and how these factors interact with cognitive and non-cognitive student background characteristics.

8. REFERENCES

[1] Padayachee, I. 2002. Intelligent tutoring systems: Architecture and characteristics. University of Natal, Durban, Information Systems & Technology, School of Accounting & Finance.

[2] Ritter, S., Anderson, J. R., Koedinger, K. R., Corbett, A. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2), 249-255.

[3] Nathan, M. J., Kintsch, W., Young, E.: A theory of algebra-word-problem comprehension and its implications for the design of learning environments. *Cognition and Instruction*, 9(4), 329-389 (1992)

[4] Walkington, C., Petrosino, A., Sherman, M. 2013. Supporting algebraic reasoning through personalized story scenarios: How situational understanding mediates performance and strategies. *Mathematical Thinking and Learning*, 15(2), 89-120.

[5] Walkington, C., Sherman, M., & Petrosino, A. 2012. 'Playing the game' of story problems: Coordinating situation-based reasoning with algebraic representation. *Journal of Mathematical Behavior*, 31(2), 174-195.

[6] Hidi, S., & Renninger, K. 2006. The four-phase model of interest development. *Educational Psychologist*, *41*(2), 111-127.

[7] Vilenius-Tuohimaa, P. M., Aunola, K., Nurmi, J. E. 2008. The association between mathematical word problems and reading comprehension. *Educational Psychology*, 28(4), 409-426.

[8] Shaftel, J., Belton-Kocher, E., Glasnapp, D., Poggio, J. 2006. The impact of language characteristics in mathematics test items on the performance of English language learners and students with disabilities. *Educational Assessment*, 11(2), 105-126.

[9] Wolf, M. K., Leon, S. 2009. An investigation of the language demands in content assessments for English language learners. *Educational Assessment*, 14(3-4), 139-159.

[10] Doddannara, L. S., Gowda, S. M., Baker, R. S., Gowda, S. M., De Carvalho, A. M. 2011. Exploring the relationships between design, students' affective states, and disengaged behaviors within an ITS. In: *Proceedings of the 16th International Conference on Artificial Intelligence and Education*, pp. 31-40.

[11] Baker, R. S., de Carvalho, A. M. J. A., Raspat, J., Aleven, V., Corbett, A. T., Koedinger, K. R. 2009. Educational software features that encourage and discourage "gaming the system." In: *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 475-482.

[12] Walkington, C. 2013. Using learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, 105(4), 932-945.

[13] Bernacki, M. & Walkington, C. 2014. The Impact of a Personalization Intervention for Mathematics on Learning and Non-Cognitive Factors. Submitted to the *2014 International Conference of Educational Data Mining*, London.

[14] Graesser, A. C., McNamara, D. S., Louwerse, M. M., Cai, Z.: Coh-Metrix: Analysis of text on cohesion and language. Behavior Research Methods, Instruments, & Computers, 36(2), 193-202 (2004)

[15] Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A., Booth, R. J.: The development and psychometric properties of LIWC2007. Austin, TX, LIWC. Net. (2007)

[16] Crossley, S., Allen, D., McNamara, D. 2011. Text readability and intuitive simplification: A comparison of readability formulas. Reading in a Foreign Language, 23(1), 84-101.

[17] Crossley, S. A., Greenfield, J., McNamara, D. S. 2008. Assessing text readability using cognitively based indices. TESOL Quarterly, 42(3), 475-493.

The Impact of a Personalization Intervention for Mathematics on Learning and Non-Cognitive Factors

Matthew Bernacki
University of Nevada, Las Vegas
4505 S. Maryland Parkway
Las Vegas, NV 89012, USA
1-702-895-4013
matt.bernacki@unlv.edu

Candace Walkington
Southern Methodist University
3011 University Blvd. Ste. 345
Dallas, TX, 75205, USA
1-214-768-3072
cwalkington@smu.edu

ABSTRACT
Personalization of learning environments to the background characteristics of learners, including non-cognitive factors, has become increasingly popular with the rise of advanced technology systems. We discuss an intervention within the Cognitive Tutor ITS where mathematics problems were personalized to the out-of-school interests of students in topic areas such as sports, music, and movies. We found that relative to a control group receiving normal problems, personalization had benefits for interest and learning measures. However, personalization that included deeper connections to students' interests seemed to be more effective than surface-level personalization.

Keywords
Personalization; interest; mathematics; intelligent tutoring systems

1. INTRODUCTION
The question of how to enhance the interest and motivation of adolescents has gained increasing prominence [1] especially in secondary mathematics [2]. Students often find mathematics, especially the math in middle and high school, to be disconnected from their interests, everyday lives, and typical ways of thinking about relationships and quantities [3]. At the same time, young people are using increasingly sophisticated and technology-driven ways to pursue and learn about their non-academic interests, and have become accustomed to a high level of customization, interaction, and control when seeking knowledge [4].

As a result, the idea of designing and advancing highly *personalized* systems for student learning has become a central focus for educational stakeholders [5]. Technology systems that enact personalized learning in the classroom have the potential to intelligently adapt to students' prior knowledge, interests, preferences, and goals [4]. In mathematics, these systems can make explicit connections between the interests students pursue outside of school – like sports, video games, or social networking – and the academic concepts they are learning. Algebra in particular is a rich space for such connections to be made [6] – students experience mathematical concepts like rate of change as they gain points in their favorite video game, track their pace in cross country, or accumulate followers on Instagram. As Algebra is often considered to be a gatekeeper to higher-level mathematics [7], and a subject that adolescents struggle to see as relevant [3], it may be a particularly important area for the development of interventions for personalized learning. We posit that 1) using a technology-based system for personalization that grounds algebra problems in students' out-of-school interests has the potential to elicit students' interest in the mathematics content to be learned, and 2) that personalization to well-developed individual interests can have a long-term effect on students' learning of algebraic concepts and their motivation to learn mathematics.

2. THEORETICAL FRAMEWORK
Interest has been defined as being both the state of engaging and the predisposition to re-engage with particular activities, events, and ideas over time [8]. Researchers have defined two types of interest. *Situational interest* is a state of heightened attention and increased engagement elicited by elements of an environment that are surprising, salient, evocative, or personally relevant. Situational interest can be *triggered* in response to stimuli, and becomes *maintained* over time as a learner engages further with the stimuli [8]. *Individual interest* is an enduring preference for certain objects or activities that persists over time and involves knowledge, value, and enjoyment; individual interest can be *emerging* or *well-developed*.

Situational interest can also be subdivided into interest based on *enjoyment* of the activity and interest based on *valuing* of the activity with respect to other things the learner values. Value-based situational interest has also been referred to as utility value – a learner's awareness of the usefulness of a topic to their life and goals [9]. Interventions that are intended to trigger students' situational interest are sometimes called "catch" interventions – the idea is to immediately grab students' attention through salient, evocative, relevant, or surprising characteristics of the instructional materials. Interventions that are designed to promote maintained situational interested as sometimes called "hold" interventions – they often reveal the value of the content to students' lives and goals, seeking to empower students [10-12]. For example, Mitchell [4] proposed that activities involving group work, computers, and puzzles function as "catch" mechanisms in the secondary mathematics classroom, while meaningfulness and involvement "hold" situational interest. Research has shown that when individuals are interested in a task or activities, they engage in more productive learning behaviors and have improved learning outcomes [e.g., 13].

An important question, then, is how to elicit and develop learners' interests for academic content areas. *Personalization* is a particular kind of intervention that can be used in learning environments to accomplish this goal. Personalization interventions identify topics for which learners have emerging or well-developed individual interest, and then connect these topics to academic content topics they are learning about in school (like algebra), for which they may have a lower level of interest. For example, consider a student who has a well-developed individual interest in music, but is not interested in Algebra. In their Algebra I class, they may engage with a variety of problems and projects that explore the mathematics behind musical pieces. Over time, the connection between these two areas might support her in developing situational interest based on her enjoyment of the incorporation of music as a context and the value perceived for music-themed problems, ultimately leading to the development of individual interest in Algebra [14]. By making explicit

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

connections to students' interests, personalization interventions are hypothesized to trigger situational interest in the academic content being learned, which can be maintained over time and eventually develop into individual interest in that content area. Personalization can increase students' engagement in the math task, improve their performance on personalized math tasks and future math tasks that are not personalized [15], and may even increase students' interest in the math they now see as relevant to their personal interests. However, little research has investigated the mechanisms by which personalization promotes these learning outcomes. In this study, we test this *situational interest hypothesis* by monitoring students' interest in math units via embedded self-report surveys and examining whether personalization induces higher levels of situational interest, and whether this situational interest transforms into individual interest. Thus we test whether increased situational interest is an important mechanism through which personalization may gain its effect.

In addition to possessing enjoyment and value components, Renninger, Ewen, and Lasher [16] accentuate that interest also involves knowledge. Learners tend to possess useful prior knowledge related to their areas of interest, but this knowledge may be intuitive and informal with respect to underlying principles, making connections to concepts being learned in school (like algebra) difficult to acknowledge or articulate. In addition to possessing the potential to spur enjoyment and value-driven reactions to an academic content area, personalization is advantageously positioned to formalize students' intuitive prior knowledge about their interests by explicitly connecting it to a concept learned in school. For example, a learner with substantial knowledge of musical composition may have implicit understandings of the mathematical or numerical underpinnings of music, and this knowledge can potentially act as a support when they are learning formal algebra. In mathematics education, this follows a "funds of knowledge" perspective [17], which accentuates that students bring with them to the classroom powerful quantitative ways of reasoning from their home and community lives. These informal, interest-based funds of knowledge are potential strengths that can be leveraged through thoughtful instructional approaches like personalization to develop students' algebraic knowledge. In this study, we test the *funds of knowledge hypothesis* by examining whether solving personalized problems that incorporate deeper features of one's interest (e.g., mechanics of a popular video game) elicit stronger effects on learning than problems personalized based on shallower features of a learner's interest (e.g. passing reference to a game title in a problem about snacking) or non-personalized problems. Thus we test whether increased activation of prior knowledge is an important mechanism through which personalization gains its effect.

Whereas outside interests can be leveraged by personalization, initial interest in mathematics may moderate the effectiveness of personalization interventions. Durik and Harackiewicz [10] found that an intervention designed to "catch" (i.e., trigger [8]) student interest (adding colorful, vivid decorations to instructional materials) was most effective for learners with low individual interest in mathematics (IIM), but hampered learners with high IIM. Conversely, they found that an intervention designed to "hold" (i.e., maintain based on value [8]) student interest (informing students of the value of the content being learned) was beneficial for high IIM students, and detrimental for low IIM students.

In order for personalized instructional materials to successfully activate knowledge, trigger interest, and enhance perceptions of value, Walkington and Bernacki [14] identified three key features designers must consider. First is the *depth* of the intervention – whether the personalization draws upon surface level aspects of a learners' interest (e.g., simply inserting familiar objects or names into an already-designed task), or whether the personalization involves deep, authentic connections to actual experiences the learner has pursuing an interest like music. Second is the *grain size* of the intervention – whether the personalization is targeted to the specific experiences of an individual, or to the generic experiences of an entire group. When considering grain size, it is important to remember that some topics will tend to tap into the interests of larger groups of students more than others – for example, a problem about the specifics of football may match the fine-grained interests of more ninth graders than a problem about field hockey. Use of these topics that relate to many students' experiences may be a productive way to allow materials to be personalized at a finer grain size. Third is the *ownership* of the personalization – whether the students themselves take a role in generating the connections between the academic content area and their interests, or if teachers or curriculum developers control the personalization. In this study, we examined students' interest in mathematics and algebra learning when exposed to a personalization intervention of medium grain size (i.e., personalized for local users based on interest interviews conducted at the same school in a prior year) versus a standard set of problems (i.e., broad grain size written by curriculum developers for all Algebra I students who use the curriculum). In the fourth unit of the intervention, we also varied the depth of problems by personalizing on surface or deep features of the problem to examine the effects of depth on interest and learning (i.e. the funds of knowledge hypothesis). No manipulation of problem ownership was conducted.

In the present study, we pursue the following research questions by implementing a personalization intervention for Algebra I:

1) What is the immediate impact of a personalization intervention on students' situational interest in algebra instructional units?
2) What long-term effect does personalization have on students' individual interest in algebra?
3) What is the impact of a personalization intervention on students' learning of algebra concepts?
4) How does depth influence the impact of personalization on interest and learning?

Based on prior work examining the effects of personalization on learning [15] and theoretical assumptions about the development of interest [8] including the situational interest hypothesis, we hypothesize that 1) Personalized problems should trigger greater situational interest in algebra units than standard problems; 2) Students completing personalized problems that incorporate out of school interests will report greater individual interest in algebra; and 3) Students who complete personalized problem solving units will achieve greater increases in their algebra performance than students completing standard problem solving units. In accordance with the funds of knowledge hypothesis, we expect 4) that students who complete problems that are personalized based on deeper features of their interest area should outperform those completing problems personalized on surface features of the problems and standard problems.

3. METHODS

3.1 Participants and Environment

Total participants included $N = 152$ ninth grade Algebra I students in the classes of two Algebra I teachers. Students attended a rural

Northeastern school that was 96% Caucasian with 21% of students eligible for free or reduced price lunch. In 2012, 71% of students passed the state standardized test in Mathematics, which is administered in the 11th grade. The sample was 51% female. Because one teacher at the school site did not administer the pretest before students began using the Cognitive Tutor, eighty-three students completed pretest, posttest and all questionnaires delivered in the CTA software and compose the primary sample for this study.

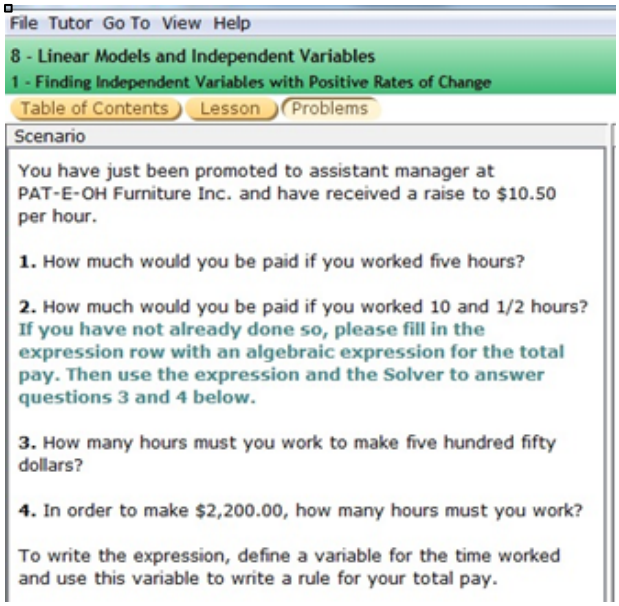
The school at which the study took place used the Cognitive Tutor Algebra (CTA) curriculum [18]. CTA is an intelligent tutoring system for Algebra I that uses *model-tracing approaches* to relate the students' actions back to the domain model to provide individualized error feedback. CTA also uses *knowledge-tracing approaches* to track learning from one problem to the next, using this information to identify strengths and weakness in terms of production rules. CTA presents learners with algebra story problems where they must navigate tabular, graphical, and symbolic representations of functions (Figure 1). Students in schools that use CTA typically use the software 2 days per week.

4. Personalization Intervention

Before entering the first unit in CTA (Unit 1), all participants were given an interests survey where they would rate their level of interest in 10 topic areas – music, art, cell phones, food, computers, games, stores, TV, movies, and sports. Participants were then assigned to one of two main conditions: (1) a Control Condition that received the standard algebra story problems in all units in CTA including Units 1, 3, 7, and 9 covering linear equations, (2) an Experimental Condition that received versions of these same problems with the same underlying structure that were matched to the interests they indicated on the interests survey for Units 1, 3, 7, and 9 (i.e. Personalization Condition). In unit 9, we tested the funds of knowledge hypothesis by further subdividing learners in the Personalization condition to (A) a Deep Personalization condition where they received personalized problems with greater depth – i.e., the personalized problems the Deep Personalization group received in Unit 9 were written to better correspond to ways that adolescents might actually use linear functions when pursuing their interests, and were intended to draw upon "funds of knowledge" more explicitly. The remaining students were assigned to (B) a Surface Personalization Condition where they received problems that contained stories with only superficial references to their identified interests. These problems should elicit situational interest, but not draw upon knowledge about one's interests.

In the first sample Control problem in Table 1, students must identify the relationship between dosage and weight. This relationship is grounded in a story that provides a context that likely to be of limited relevance to the student. In the Surface Personalization problem the structure of the problem remains consistent, but a topic that corresponds to the learners' personal interests has been applied. In the Deep Personalization version, the personal interest is applied more intentionally. Like the surface-level personalization problem, The Clash of Clans problem matches students' reported interest in games. However it is also intended to draw upon the learner's knowledge of the game's architecture to frame the underlying algebraic relationship to be learned in a deeply relevant context (i.e. it is actually useful to keep track of the relationship between elapsed time and how goals are accomplished, and this quantity is explicitly tracked and displayed for the player within the game interface). We consider this to be a deeper level of personalization compared to the Surface Personalization condition, as it seems less likely that despite an interest in games, a teen would care about or track exactly how frequently they consume snacks during play. Personalized problems were written based on surveys (N = 45) and interviews (N = 23) with Algebra I students at the school where they discussed their out-of-school interests.

Deep Personalization problems were written to more closely correspond to quantitative information given by students in the interviews and open-ended surveys about their out-of-school interests, including interviews with Algebra I students at the school where the study was conducted. In these interviews, students discussed how they consider rate of change as they play video games, participate in sports, track their rate of texting and battery usage on their cell phone, engage in cooking, work at part-time jobs, activities, and so on. (see [6] for a full analysis of student interviews).



Answer Key:

Quantity Name	the time worked	the money earned
Unit	hour	dollar
Expression	X	10.5X
Question 1	5	52.5
Question 2	10.5	110.25
Question 3	52.381	550
Question 4	209.5238	2200

Figure 1. Screenshot of Cognitive Tutor Algebra environment with answer key superimposed

Table 1. Study Conditions

	Control	Surface Personalization	Deep Personalization
GAMES	The correct dosage of a certain medicine is two milligrams per 25 pounds of body weight.	While playing cards a person typically eats two snacks for every 25 minutes of playing time in a card game.	When playing Clash of Clans a player can build two barracks for every 25 minutes of playing time.
SPORTS	Three out of every five people in a recent survey supported the President's Health Plan.	Three out of five people have attended a Pittsburgh Steelers game in their lifetime.	Three out of five free throws are successful for NBA players.
FOOD	Directions for a swimming pool chemical that controls the growth of algae state that you should use six fluid ounces of chemical for every 500 gallons of water.	Looking through a collection of online recipes, there are six recipes that require powdered sugar for every 500 recipes that you find online.	In a family recipe you use six drops of hot pepper oil for every 500 ounces of chili that is being cooked.

Problems across the 3 conditions were written to hold constant factors like order of information given, numbers, sentence structure and length, mathematical vocabulary, readability, pronoun use, and distractor information. The personalized problems did *not* require that students have additional knowledge of specific numerical mathematical information in their interest area (e.g., knowing how many points a field goal is worth) – all information given was matched across problem types.

All instructional units involved in the study involved linear functions. Of the core sample comprising most of our analyses, 31 participants were assigned to the Control, 34 were assigned to Surface Personalization, and 27 were assigned to Deep Personalization.

4.1 Measures

We collected the following measures from all participants:

4.1.1 Paper-Based Pre/Post Assessments

At the beginning of the school year, prior to entering the tutor, all students completed a paper-based pre-test on linear functions. The test contained 4 story problems where a linear function was described that either had a slope and intercept (2 problems) or had only a slope (2 problems). Participants first were given an x value in the linear function and asked to solve for y, then they were given a y value in the linear function and asked to solve for x. Finally, they were asked to write the linear function using algebra symbols. A post-test was administered to all students around the midterm of their ninth grade year (i.e., four months later). The post-test contained 4 matched items containing slightly different wording and numbers. Students' responses to each part of each problem were scored as correct or incorrect.

4.1.2 Domain-Level Motivational Surveys

Prior to entering Unit 1 (pre-) and Unit 10 (post-) in CTA, the software presented students with a survey asking them to rate their attitudes about algebra. Specifically, they rated their individual interest in mathematics (IIM), as well as their maintained situational interest–enjoyment and maintained situational interest-value for mathematics. Subscales were adopted from a larger set of scales from Linnenbrink-Garcia et al. [19]. Sample items for each scale appear in Table 2.

4.1.3 Unit-Level Motivational Surveys

After each unit impacted by the personalization intervention (Figure 2; Units 1, 3, 7, and 9), participants were also given a unit-level motivational survey that assessed the degree to which that unit triggered their situational interest and maintained their situational interest in the CTA unit. These scales were adapted based on measures from Linnenbrink-Garcia et al. [19] with the math unit as the referent. Sample items for each scale appear in Table 2, as do Cronbach's alphas for the initial administration of each survey. An overview of the survey measures and CTA units completed by participants in this study is provided in Figure 2.

Table 2. Interest Measures

Interest Measure	Sample item	α
Individual Interest in Mathematics	Thinking mathematically is an important part of who I am.	.92
Maintained Situational Interest in Math- Value	What we are studying in math class is useful for me to know.	.92
Maintained Situational Interest in Math- Enjoyment	I really enjoy the math we do in this class.	.89
Triggered Situational Interest in Math	The topics in this unit grabbed my attention.	.84
Maintained Situational Interest in Unit - Value	The math in this unit is useful for me to know.	.90
Maintained Situational Interest in Unit - Enjoyment	In this unit, I really enjoyed the math.	.84

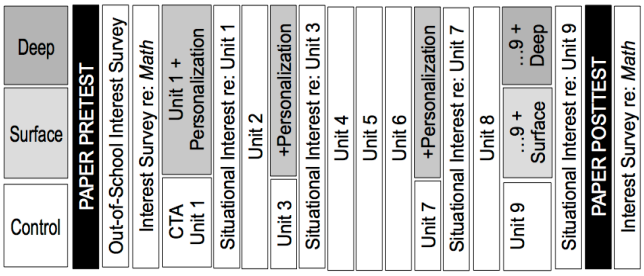


Figure 2. Measures

5. RESULTS

We report results as they address the first three research questions in section 2. We do not provide a separate section for research question 4 (impact of depth of personalization), and instead discuss the results for depth of personalization within each of the other three sections.

5.1 What is the impact of personalization on students' situational interest in algebra units?

To assess the effect of the personalization interventions on students' situational interest, we conducted a series of analyses of covariance examining students' reported triggered and maintained interest in CTA units. All students were given unit-level surveys assessing their level of interest in the instructional unit after each of the units impacted by the personalization treatment (Units 1, 3, 7, and 9). We controlled for initial individual interest in mathematics (IIM) as indicated on the domain survey before Unit 1 (Figure 2).

Students in the two Personalization conditions (i.e., Surface Personalization and Deep Personalization are identical in Units 1, 3, and 7) consistently reported significantly higher levels of triggered situational interest than students assigned to the Control condition (Table 3; Unit 1 $F(1,80) = 5.19$, MSe = .96, $p = .03$, Unit 3 $F(1,80) = 5.31$, MSe = .98, $p = .02$; Unit 7 $F(1,80) = 3.82$, MSe = .91, $p = .05$).

Significant differences between any of the 3 groups in triggered situational interest were not obtained in Unit 9. The level of triggered situational interest reported by the Deep Personalization was consistent with prior units with the triggered interest for the Surface Personalization group was slightly lower. The Control group, however, reported greater triggered situational interest, and the inclusion of three groups (two with smaller Ns) further diminished the statistical power available to detect effects.

No significant differences in maintained situational interest were found between groups on any of the four units observed, Fs < 3.73, ps = ns. Directionally, measures of maintained situational interest generally favored the personalization groups.

5.2 What effect does personalization have on students' individual interest in algebra?

All students were given domain-level surveys assessing their interest towards learning algebra prior to the intervention and after the final personalized unit (i.e., Unit 9). A repeated measures analysis of variance examining change in Individual Interest in Mathematics (i.e., Post-Pre) between the two Personalization conditions (i.e., Deep & Surface) versus Control was conducted to examine the main effect of Time and Interaction between Time X Condition. Results indicated a significant main effect of Time, F (1, 81) = 5.39, MSe = 1.75, p = .023. Overall, students' individual interest in mathematics declined from pretest to posttest. Analyses also indicated a marginally significant interaction between Time and Condition, F (1, 81) = 3.73, $p = .057$. Students in the control group significantly *reduced* their rating of individual interest in algebra an average of 0.37 points over the 10-unit span (Table 3; $t(29) = 3.21$, $p < .01$), while students in the Deep and Surface Personalization groups maintained their individual interest in algebra ($M = 0.04$ decline). Thus personalization had a positive effect in that it preserved students' individual interest in algebra. Within the Personalization condition, no differences were found between students who received Surface versus Deep Personalization.

5.3 What is the impact of personalization on students' learning of Algebra I concepts?

The pre- and post- test scores on the algebra learning measures for each of the three conditions is shown in Table 4. A linear regression model predicting amount of absolute gain from pre- to post-test (i.e., post-test score minus pre-test score) was fit to the data, with students' class period as a random effect. Adding a predictor for Condition significantly improved the fit of the model ($\chi^2(2) = 6.39$, $p = 0.04$), as did a control variable for students' initial level of individual interest in mathematics (IIM) prior to the intervention ($\chi^2(1) = 4.07$, $p = 0.04$). The interaction of Condition and IIM also significantly improved the fit of the model ($\chi^2(2) = 14.43$, $p < .001$).

Table 3. Estimated Marginal Means Controlling for Individual Interest in Math

Variable			Personalization[a]		Control[b]		
	Unit		EMM	SE	EMM	SE	
Triggered Situational Interest	1		2.86	0.13	2.33	0.19	*
	3		2.82	0.13	2.27	0.19	*
	7		2.69	0.13	2.25	0.18	*
	9	D[c]	2.82	0.18	2.55	0.19	
		S[d]	2.56	0.20			
Maintained Situational Interest - Value	1		2.95	0.13	2.77	0.19	
	3		3.07	0.13	2.74	0.18	
	7		2.76	0.13	2.76	0.18	
	9	D	2.84	0.19	2.82	0.18	
		S	2.70	0.17			
Maintained Situational Interest - Enjoyment	1		2.76	0.12	2.46	0.17	
	3		2.81	0.13	2.40	0.18	
	7		2.66	0.12	2.35	0.17	
	9	D	2.62	0.19	2.50	0.18	
		S	2.33	0.17			
Individual Interest in Math	Pre		2.87	.14	3.34	.20	
	Post		2.83	.16	2.94	.22	

Notes. *- $p < .05$, EMM = Estimated Marginal Mean, SE = Standard Error, D = Deep personalization, S = Surface Personalization, [a] - N = 55, [b] - N = 28, [c] - N = 24, [d] - N = 31

Table 4. Scores on Knowledge tests by Condition

		Pretest		Posttest	
Condition	N	M	SD	M	SD
Control	32	0.68	0.2	0.83	0.12
Surface Personalization	29	0.73	0.15	0.82	0.15
Deep personalization	32	0.63	0.22	0.84	0.18

The regression output is shown in Table 5. The reference category is the Control Group, and we interpret all significant simple effects regardless of whether they are displayed in the table. The IIM control measure was dichotomized to separate students with high IIM (average rating of 3 or more) from low IIM (average rating less than 3) to aid interpretability and to be consistent with prior work [e.g., 14]. As can be seen from Table 5, for students with low individual interest in math, Deep Personalization was significantly more effective than Control ($p < 0.05$). Additional contrasts not shown in the table compared Surface Personalization to Deep Personalization, and found that for students with low IIM,

Deep Personalization was significantly more effective than Surface Personalization (B = 0.24, *SE* (B) = 0.07, $p < 0.001$). Finally, within the Deep Personalization condition, students with high IIM gained significantly less than students with low IIM (B = .17, *SE*(B) = .07, $p = .01$).

Table 5. Regression Output for Pre/Post Learning Gains

	B	SE (B)	*t*	*p*
(Intercept)	.13	.07	1.81	.07
Control	(ref.)			
Surface Personalization	-.10	.08	-1.33	.18
Deep Personalization	.14	.07	1.97	.05
Low IIM	(ref.)			
High IIM	.00	.07	-.07	.94
Surface Personalization × High Initial Individual Interest	.08	.10	.82	.41
Deep Personalization × High Initial Individual Interest	-.17	.10	-1.71	.09

6. DISCUSSION & CONCLUSION

This study examined whether personalizing algebra problems to students' out-of-school interests would increase their situational interest in CTA algebra problems, increase their interest in mathematics, and improve their acquisition of algebra knowledge (i.e., the situational interest hypothesis). It additionally tested whether solving problems that incorporated deep features of an interest into problems would produce greater benefits that solving problems that incorporated interests superficially or standard problems (i.e. the funds of knowledge hypothesis). Students who received problems personalized to their out-of school interests reported significantly higher triggered situational interest for CTA math units. Compared to a Control group that experienced a drop in their individual interest in mathematics, Personalization also had a preserving effect on students' interest in mathematics. After accounting for students' initial individual interest in mathematics, significant differences in learning gains were found between groups of students in the Deep Personalization, Surface Personalization and Control Conditions. These findings are next discussed in light of prior theory and research.

6.1 Personalization and Situational Interest

Students who completed algebra problems personalized to their interests reported greater triggered situational interest compared to students who completed standard CTA problems, however students who solved personalized problems did not report significantly greater maintained interest resulting from enjoyment or perceptions of value. The finding that personalization was effective in triggering situational interest is encouraging as we consider the Control condition to be a considerably strong control. That is, the standard problems included in tutor units might be considered to be personalized to student interests at a very broad grain size [11] – they were generally written by teachers and curriculum writers with this student population in mind (i.e., adolescent algebra learners). The personalized problems in the intervention, on the other hand, had a medium grain size – they were written for and provided to subsets of the student population that had particular topic interests (e.g., sports, video games). The change from a large to a medium grain size was sufficient to elicit changes in triggered situational interest, though additional effort may be necessary to elicit sufficient enjoyment or perception of value to maintain students' situational interest. Indeed, in another personalization study [20], we found that a personalization intervention with a much smaller grain size where students wrote and solved problems that incorporated features of their personal interests produced increases in students' maintained situational interest associated with perceived value. This intervention also involved a higher level of ownership of the personalization on the part of the students [14], which suggests that personalization at a medium grain size may successfully trigger situational interest, but a personalization at a smaller grain size with some level of ownership may be necessary to achieve more enduring situational interest in math units. This type of intervention may be especially important given that it takes the burden of generating fine-grained instructional materials away from teachers and curriculum developers and places it on students.

6.2 Personalization and Individual Interest

Despite a failure to elicit maintained situational interest, the Personalization intervention did have a significant effect on students' individual interest in mathematics. Importantly, the individual interest items assessed how students felt about the domain of mathematics as a whole, rather than how they felt about the particular math class they were enrolled in or the particular units they were working on. This preservation of individual interest in algebra over half a year of high school coursework is a desirable outcome, given research that documents declines in interest in math over adolescence [21, 22]. In sum, the findings from the first two research questions support the situational interest hypothesis. We consider this finding in light of theory on interest development in section 6.4.

6.3 Deep Personalization and Algebra Learning

Walkington [12] found that a one-unit personalization intervention improved students' long-term learning of algebra concepts within the CTA environment, relative to a control condition. This study extends that work and indicates that, when personalization incorporates deep features of students' out-of-school interests, it can also induce learning gains that transfer outside of an intelligent tutoring environment (i.e. to delayed, paper-based tests). However, these effects are moderated by students' initial level of individual interest in mathematics, with Deep Personalization being beneficial mainly for low IIM students. Walkington [15] did not collect such interest measures in her study, but did find that personalization was most effective for students who were making slower progress through CTA– a variable known to track closely with interest in math [23]. We consider these findings in light of proposed hypotheses that personalization may obtain effects on learning by activating students' funds of knowledge in their out-of-school interest, and that personalization may trigger greater situational interest in math tasks. The current study showed that Deep Personalization was significantly less effective for learners with high IIM, compared to learners with low IIM. This, along with the results that personalization triggers but does not maintain situational interest, suggests that even Deep Personalization may achieve its effects on learning as a "catch" intervention, immediately eliciting triggered situational interest. That is, solving personalized problems triggered students' interests, but did not maintain them. This provides some promise as prior research has shown catch interventions that trigger interest to be beneficial primarily for learners with low IIM [10]. This is contrasted with a "hold" intervention that maintains situational interest, often by communicating the value of the content being learned. In this study personalization did not increase students' perceptions that

algebra problems had value, but additional interventions aimed at boosting perceived value and relevance [11, 12] could potentially be incorporated to ITSs to also obtain this effect and its benefits for learning.

Although we termed our Condition "Deep" Personalization, the connections made to learners' actual experiences may not have been uniformly deep depending on students more specific interests within a topic area, and thus may not have elicited value-based reactions from some students. This stems from issues with the grain size of the intervention – students merely indicated their level of interest in a broad topic (e.g., "sports"), and were then given problems that could cover the entire space of activities that fell within that topic (e.g., basketball, hockey, football), without considering students more specific interest in a subtopic (e.g., just hockey). Although attempts were made to use the "high-leverage" interest sub-topics that many students would have specific knowledge of (i.e., football rather than field hockey) this approach likely allowed for the personalization to have highly variable level of correspondence to students' exact interests. The level of correspondence depended on the overlap between a student's interest and the commonly reported interests by peers in surveys and interviews prior to problem development. Walkington and Bernacki [20] found significant increases in maintained situational interest (value) for students who authored problems about their specific interests, suggesting that the smaller grain size and increased ownership of the personalization intervention in that study allowed it to function more as a "hold" intervention.

Finally, the current study showed that Deep Personalization was significantly more effective than Surface Personalization for students with low IIM. This suggested that personalization may need to have at least a moderate level of depth for it to be effective at all for supporting learning outcomes for any subgroup of students. Indeed, a number of recent personalization interventions that employed relatively surface-level personalization have reported null findings [24, 25]. Thus we conclude from all of these analyses that a personalization intervention with a moderate depth and grain size can potentially have long-term effects on student learning for students who begin with limited interest in mathematics. However, increasing depth and personalizing at an even smaller grain size may have more powerful effects, especially for students with higher IIM for whom value-based connections may be most critical.

Although learning gains were produced for low IIM students who received Deep Personalization (rather than Surface Personalization), these students did not show differences in situational or individual interest measures within Unit 9 compared to the Surface Personalization group. There were also no differences between Surface and Deep in individual interest over the course of the entire intervention. This suggests that Deep Personalization may gain its effectiveness over Surface Personalization by connecting to students' prior knowledge (funds of knowledge hypothesis) rather than triggering and maintaining differing levels of situational interest (situational interest hypothesis). However, ultimately comparisons between these two groups are of limited usefulness given the relatively small sample sizes. Thus we find limited but promising support for the funds of knowledge hypothesis.

6.4 Theoretical Implications

When viewed through the lens of interest development theory [8], the findings regarding personalization and interest development are somewhat puzzling. Per Hidi and Renninger's [8] theory, interest is 1) triggered by environmental stimuli and 2) maintained when engagement in the environment is enjoyable or confers value through consistent or repeated situational interest. This supports 3) the emergence of an individual interest, which 4) becomes well developed over time. In this study, analyses reveal a triggering of situational interest among students in the Surface and Deep Personalization conditions, no reported maintenance of situational interest via enjoyment or value, but a significant effect of Personalization on individual interest. Thus individual interest developed without being maintained during learning; this requires that we consider alternate explanations by which such effects on individual interest may have been obtained.

One potential explanation is that the way instructors used Cognitive Tutor in the math classes may have reproduced some of the behaviors expected when students' situational interest is maintained. In their model, Hidi and Renninger [8] describe that those who maintain interest in a topic tend to repeatedly engage with content involving the topic (e.g., a student who is interest in dolphins may seek more opportunities to learn about them by reading books about them in school or choose "dolphins" as a topic for school assignments). While students' did not report that personalized Cognitive Tutor Algebra units maintained their interest to a degree that we would expect them to voluntarily seek out opportunities to learn using Cognitive Tutor, the compulsory use of the Cognitive Tutor in math class twice a week for many months effectively ensured repeated engagement in (personalized) problem solving via CTA use. Thus we could conclude that the continued exposure to math content personalized to one's out-of-school interests approximated behavioral outcomes of maintained situational interest and created an alternate pathway by which individual interest was preserved in Personalization conditions (i.e., no drop in interest), but not in the Control condition where there was no initially triggered interest. Much like the typical adolescent whose interest in math declines over time, students in the Control condition were required to complete math units that did not trigger situational interest and subsequently reported declines in their interest in mathematics.

6.5 Conclusion

The results obtained in this study provide important insight about the ways depth and grain size of personalization may impact the development of students' interests in their math course, the domain of mathematics, and ultimately their long-term learning of algebra concepts. In future analyses, we will analyze additional data from students participating in this study, and look for difference in in behavior and performance within intervention and subsequent CTA units, including analyses of learning behaviors using log-files and automated detectors.

7. ACKNOWLEDGMENTS

Both authors contributed equally to this manuscript. The authors thank Steve Ritter, Susan Berman, Tristan Nixon and Steve Fancsali (Carnegie Learning), Gail Kusbit (Carnegie Mellon University & LearnLab) and participating teachers. Funding for the study was provided by a subgrant of National Science Foundation Award # SBE-0354420. Additional funding as provided by IES Award # R305B100007.

8. REFERENCES

[1] Hidi, S., & Harackiewicz, J. (2000). Motivating the academically unmotivated: A critical issue for the 21st century. Review of Educational Research, 70, 151–179.

[2] Mitchell, M. (1993). Situational interest: Its multifaceted structure in the secondary school mathematics classroom. *Journal of Educational Psychology*, *85*, 424–436.

[3] McCoy L. P. (2005). Effect of demographic and personal variables on achievement in eighth-grade algebra. *Journal of Educational Research*, *98*(3), 131–135.

[4] Collins, A., & Halverson, R. (2009). *Rethinking Education in the Age of Technology: The Digital Revolution and Schooling in America*. New York: Teachers College Press.

[5] U.S. Department of Education, Office of Educational Technology, Transforming American Education: Learning Powered by Technology, Washington, D.C., 2010. http://www.ed.gov/sites/default/files/netp2010.pdf

[6] Walkington, C., Sherman, M., & Howell, E. (in press). Connecting Algebra to sports, video games, and social networking: How personalized learning makes ideas "stick." *Mathematics Teacher*.

[7] Moses, R., & Cobb, C. (2001). *Radical Equations: Math Literacy and Civil Rights*. Boston: Beacon Press.

[8] Hidi, S., & Renninger, K. (2006). The four-phase model of interest development. *Educational Psychologist*, *41*(2), 111-127.

[9] Eccles, J. (1983). Expectancies, values and academic behaviors. In R. C. Atkinson, G. Lindzey, & R. F. Thompson (Eds.), *Achievement and achievement motives: Psychological and sociological approaches* (pp. 75-146). San Francisco: W.H. Freeman & Co

[10] Durik, A., & Harackiewicz, J. (2007). Different strokes for different folks: How individual interest moderates effects of situational factors on task interest. *Journal of Educational Psychology*, *99*(3), 597-610.

[11] Hulleman, C. S., Godes, O., Hendricks, B. L., & Harackiewicz, J. M. (2010). Enhancing interest and performance with a utility value intervention. J*ournal of Educational Psychology*, *102*, 880-895.

[12] Hulleman, C. S., & Harackiewicz, J. M. (2009). Promoting interest and performance in high school science classes. *Science*, *326*(5958), 1410–1412.

[13] Harackiewicz, J., Durik, A., Barron, K. Linnenbrink, E., & Tauer, J. (2008). The role of achievement goals in the development of interest: Reciprocal relations between achievement goals, interest, and performance. *Journal of Educational Psychology*, *100*(1), 105-122.

[14] Walkington, C., & Bernacki, M. (in press). Motivating students by "personalizing" learning around individual interests: A consideration of theory, design, and implementation issues. In S. Karabenick & T. Urdan (eds.) *Advances in Motivation and Achievement*, Emerald Group Publishing.

[15] Walkington, C. (2013). Using learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, *105*(4), 932-945.

[16] Renninger, K., Ewen, L., & Lasher, A. (2002). Individual interest as context is expository text and mathematical word problems. *Learning and Instruction*, *12*, 467-491.

[17] Civil, M. (2007). Building on community knowledge: An avenue to equity in mathematics education. In N. Nassir. and P. Cobb (Eds.) *Improving access to mathematics: Diversity and equity in the classroom* (pp. 105-117). Teachers College Press.

[18] Carnegie Learning (2013). Cognitive Tutor Algebra [software]. Carnegie Learning, Inc. Pittsburgh, PA, USA.

[19] Linnenbrink-Garcia, L., Durik, A., Conley, A., Barron, K., Tauer, J., Karabenick, S., & Harackiewicz, J. (2010). Measuring situational interest in academic domains. *Educational Psychological Measurement*, *70*, 647-671.

[20] Walkington, C., & Bernacki, M. (2014). Students authoring personalized "algebra stories": Problem-posing in the context of out-of-school interests. Presentation at the 2014 Annual Meeting of American Educational Research Association.

[21] Fredricks, J. A., & Eccles, J. (2002). Children's competence and value beliefs from childhood through adolescence: Growth trajectories in two male-sex-typed domains. *Developmental Psychology*, *38*, 519–533.

[22] Frenzel, A. C., Goetz, T., Pekrun, R., & Watt, H. M. G. (2010). Development of mathematics interest in adolescence: Influences of gender, family, and school context. *Journal of Research on Adolescence*, *20*, 507–537.

[23] Bernacki, M. L., Nokes-Malach, T.J., Aleven, V., & Glick, J. (2014). Intelligent tutoring systems promote achievement in middle school mathematics, especially for students with low interest. Presentation at the 2014 Annual Meeting of the American Educational Research Association.

[24] Bates, E., & Wiest, L. (2004). The impact of personalization of mathematical word problems on student performance. *The Mathematics Educator*, *14*(2), 17-26.

[25] Caker, O., & Simsek, N. (2010). A comparative analysis of computer and paper-based personalization on student achievement. *Computers & Education*, *55*, 1524-1531.

Promoting Growth Mindset Within Intelligent Tutoring Systems

Korinn S. Ostrow
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
ksostrow@wpi.edu

Sarah E. Schultz
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
seschultz@wpi.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
iarroyo@wpi.edu

ABSTRACT

When designing adaptive tutoring systems, a myriad of psychological theories must be taken into account. Popular notion follows cognitive theory in supporting multi-channel processing, while working under assumptions that pedagogical agents and affect detection are of the utmost significance. However, motivation and affect are complex human characteristics that can muddle human-computer interactions. The following study considers the promotion of the growth mindset, as defined by Carol Dweck, within middle school students using an intelligent tutoring system. A randomized controlled trial comprised of six conditions is used to assess various delivery mediums of growth mindset oriented motivational messages. Student persistence and mastery speed are examined across multiple math domains, and self-response items are used to gauge student mindset, enjoyment, and perception of system helpfulness upon completion of the assignment. Findings, design limitation, and suggestions for future analysis are discussed.

Keywords

Motivational messages, growth mindset, pedagogical agents, multi-media learning principles, e-learning design.

1. INTRODUCTION

The optimal design of adaptive tutoring systems is a continuous debate for researchers in the Learning Sciences. Decisions when authoring content can be immense, including not only the user interface and tutor material, but also the presence of adaptive feedback strategies such as hints or scaffolding, the use of affect detectors, and in growing popularity, the use of pedagogical agents. While many adaptive tutors share designs rooted in cognitive theory, creators should also incorporate elements that improve student motivation, engagement, persistence, metacognition, and self-regulation skills. These elements aid in the promotion of active learning, an experience that has been shown to heighten the creation of mental connections [10]. However, successful adaptive tutoring systems are not just a random conglomeration of these learning goals. All too often, adaptive tutors are designed under the assumption that students are ideal learners, driven and motivated, ready to employ a full range of self-regulation skills coupled with technological prowess [1]. Thus, researchers have recently undertaken a more thorough examination of how to universally encourage and motivate students while still promoting self-regulated learning skills and optimizing system design [3, 8].

Human motivation has historically been explained and argued by an array of theories, as intrinsic or as extrinsic, as static or as the constant flow of needs, emotions, and cognitions [13]. In a somewhat similar sense, recent research promoting affect detection within educational technology suggests that affect plays a primary role in learning success [2]. How can researchers incorporate deeply rooted human characteristics like motivation and affect into the design of an adaptive tutoring system? A renowned leader in the field of psychology, Carol Dweck has helped to establish theories of intelligence that marry these complex constructs within the confines of learning studies [5]. Her research has shown that students approach learning tasks largely with one of two 'mindsets.' The *fixed mindset* is characterized by the notion that intelligence is somehow innate or immutable. Students who live within this fixed realm generally emit lower learning and performance outcomes as well as higher attrition rates based in the notion that effort will not lead to intellectual advancement [6]. Much of American society is rooted in this view; strong emphasis is placed on standardized testing and zero sum competition, with the goal of comparing student intelligence rather than promoting learning. Alternatively, students with a *growth mindset* believe that intelligence is malleable and that effort and persistence can lead to success. While Dweck [7] argues that neither mindset is necessarily 'correct,' she promotes the notion that mindset can be altered, and explains the growth mindset as offering a healthier mental lifestyle. Altering mindset is best achieved by varying the type of praise students receive and by realigning their definition of successful learning. By highlighting the learning process rather than the student's intelligence or performance, 'process praise' and the promotion of malleable intelligence has led to positive, long-term learning gains [5]. Students trained in the growth mindset show increased enjoyment in difficult learning tasks as well as higher overall achievement and performance [6].

An expert in his own right, Richard Mayer has devoted much of his career to promoting a series of multi-media learning principles that enhance e-learning design. These principles call for learning environments to be driven by active learning processes while considering the cognitive load and working memory of users [4]. As such, those authoring adaptive tutors

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

should utilize audio, animation, graphics, video, and other hypermedia elements to appease multiple sensory channels and thereby reduce the user's overall cognitive load. It is important to note that powerful design requires a fine balance of these resources, as exorbitance may serve to distract or disrupt learners. The evolution of pedagogical agents and learning companions within adaptive tutoring systems has served as a primary way to incorporate both multi-media elements and non-cognitive support. As guidelines for the design of human-computer interaction have followed those set forth by human-human interaction, the art of appropriating the cognitive and affective responses of pedagogical agents has been of major concern [9]. Agents are typically designed with the premise that they should respond happily to student successes and with a shared disappointment upon failures [9].

Considering the optimal design of adaptive tutoring systems and the incorporation of hypermedia and pedagogical agents to engage students in active learning, the current study seeks to analyze the promotion of Dweck's growth mindset theory within ASSISTments, an adaptive mathematics tutor. The following research questions were derived from themes relevant to Dweck's [6] work, in combination with adaptive tutoring structures unique to ASSISTments:

1. Does the addition of motivational messaging within the tutoring system affect the likelihood of student persistence or attrition?
2. Does the presence of motivational messaging within the tutoring system affect mastery speed as defined by how many items, on average, it takes for students to complete the problem set?
3. Can specific elements within message delivery be pinpointed as significantly powerful? That is, can researchers isolate an element (e.g., the presence of a pedagogical agent, the audio component, static images, or a combination of these elements) that is responsible for the majority of variance in persistence and learning efficiency?

It is hypothesized that students randomly assigned to a messaging condition will be more likely to show continued, persistent effort than those in the control condition. Similarly, regardless of the delivery medium, researchers expect students who receive mindset messages to show improved mastery speed, with fewer items, on average, required to complete a problem set. In the assessment of message delivery, it is hypothesized that motivational messages delivered using an animated version of Jane, a learning companion that originates from partnering tutor Wayang Outpost, will have a stronger effect on student persistence and learning efficiency than alternative message mediums.

2. METHODS

To determine appropriate math content for this study, the tutor's database was queried to compile a historical record of usage data for a variety of problem sets that fit within Common Core State Standards across various grade levels. All observed problem sets were of a style unique to the ASSISTments tutor, requiring students to answer three consecutive questions correctly in the same day in order to complete the assignment. If the student were to reach a preset 'daily limit' (i.e., ten problems) while attempting to solve three consecutive questions, they are prompted to consult with their teacher and try again tomorrow.

Five problem sets were chosen based on high usage, with math content spanning grades four through seven. The skill topics assessed by these problem sets included finding missing values using percent on a circle graph, equivalent fractions, multiplying decimals, rounding, and order of operations. The goal in designing multiple problem sets was three-fold: to increase data collection, to determine any significant effect for student skill level, and to determine if content was linked to student motivation, perhaps due to difficulty level. Six conditions were then established for each problem set, as defined in Table 1. These conditions were designed following the principles set forth by Mayer [4], to test matched content messages across a variety of processing channels.

Table 1. Motivational messaging conditions.

Control	ASSISTments as usual; no messages added
Animation	Jane, a female pedagogical agent, delivers messages with motion and sound
Static Image with Text	The agent is presented as a static image, with a speech bubble to deliver motivational text messages
Static Image with Audio	The agent is presented as a static image, supplemented by audio files to deliver motivational messages
Word Art	A speech cloud shows motivational text messages, with no agent involvement
Audio	The agent's voice delivers motivational messages with no graphical changes to tutor content

The student experience for each problem set was formatted in the same manner. An introductory 'question' explained the format of the problem set and alerted the student to turn on their computer volume and to use headphones if necessary. The second 'question' tested whether or not the student was able to see and hear the pedagogical agent Jane as she introduced herself as a problem-solving partner. This question was included to test the compatibility of the HTML files that supported the pedagogical agent's animation and sound conditions, thus serving as confirmation of fair random assignment. Researchers then relied on a randomization feature unique to ASSISTments that randomly assigned students to one of the six conditions depicted in Table 1. Math content was isomorphic across conditions, and was thus considered comparable in difficulty. A test drive of the student experience for each problem set can be found at [12].

Motivational message content, as depicted in Table 2, was matched across conditions to reduce confounding. These messages were validated in and derived from [1]. Each problem set was designed to randomly select questions from a pool of approximately 100 problems, containing two types of motivational message delivery: *general attributions*, in which the motivational message was presented with the primary question, and *incorrect attributions*, in which the motivational message was presented alongside content feedback if the student responded incorrectly or employed a tutoring strategy. Following this design structure, students saw general attributions on approximately half of the questions, with the remaining half displaying incorrect attributions only to students who answered a problem correctly. Therefore, each student's experience of motivational messaging may have differed slightly, even within each condition. This design was established to reduce persistent message delivery and to avoid inundating students with messages on each question, with the goal of optimizing the effects of motivational messages while retaining a primary focus on math content. All visual motivational messages appeared within the tutor and remained until the student completed the problem; audio messages were played once upon loading the problem or tutoring strategy.

Table 2. Motivational message item content.

General Attributions
1. Did you know that when we learn something new our brain actually changes? It forms new connections inside that help us solve problems in the future. Pretty amazing, huh?
2. Did you know that when we practice to learn new math skills our brain grows and gets stronger? That is so cool!
3. Hey, I found out that people have myths about math… like that only some people are "good" at math. The truth is we can all be successful in math if we give it a try.
4. I think the most important thing is to have an open mind and believe that one can actually do math!
5. I think that more important than getting the problem right is putting in the effort and keeping in mind the fact that we can all be good at math if we try.
Incorrect Attributions
1. Making a mistake is not a bad thing. It's what learning is all about!
2. When we realize we don't know why that was not the right answer, it helps us understand better what we need to practice.
3. We may need to practice a lot, but our brains will develop with what we learn.

At the end of each problem set, students were asked to partake in a series of four survey questions developed based on previously validated content from [11], to assess student mindset, goal orientation, and perceptions of enjoyment and system helpfulness. All students received these questions regardless of condition. All survey content can be accessed at [12].

3. PROCEDURE

Teachers in the state of Massachusetts who frequently use ASSISTments with their students were approached with a brief presentation explaining the study and providing examples of the conditions, motivational messages, and math content. Teachers assigned one or more of the problem sets to their students in accordance with the teachers' usual use of the tutoring system (i.e., as either classwork or homework). Material was assigned as current content and/or review, for a total of 765 student assignments. Log data was compiled for each student's performance. Prior to analysis of persistence and mastery speed, students were removed if they had noted experiencing technical difficulties or if they failed to log enough progress to enter one of the six conditions. Additional students were removed prior to survey analysis due to incompletion. Students remaining after each step are examined across problem sets in Table 3.

Table 3. Explanation of Students Remaining After Removals.

Problem Set	*A*[1]	*MA**	*SA***
Percent on a Circle Graph	87	69	62
Equivalent Fractions	255	208	205
Multiplying Decimals	62	48	47
Rounding	253	208	205
Order of Operations	108	88	86
REMAINING	**765**	**621**	**605**

A[1] = Assigned. MA = Math Analysis. SA = Survey Analysis.
*Students were removed prior to math analysis due to technical difficulties or failure to initiate a condition.
**Additional students were removed prior to survey analysis due to incompletion.

An ex post facto judgment of student gender was determined for 570 students within the sample remaining for math content analysis. Due to incompletion rates within this subset of students, gender was determined for 554 students within the sample remaining for survey content analysis.

4. RESULTS

Analyses of student persistence and mastery speed were performed at the condition level for each problem set, as well as for an aggregate of the five sets to serve as a composite analysis of the conditions across math content. To determine if an effect existed within a particular processing channel, similar conditions were compiled based on delivery elements. For example, all conditions utilizing audio were compiled to assess the effect of audio (i.e., audio, animation, static image with audio). Similar analyses were performed to determine the effect of textual messages and the effect of the pedagogical agent's presence. Researchers also compared a compilation of all conditions containing motivational messages to the control condition in order to determine the effectiveness of motivational messages in general. Initial findings suggested that in general, the sample was too advanced for the math content as students were found to be at ceiling across many of the problem sets. Thus, secondary analyses examined gender differences and assessed the aforementioned variables for a subset of students operationally defined as "strugglers," or those requiring more than three questions to complete their assignment.

When considering student persistence, as defined by continuing until reaching completion, ANOVA results suggested null results ($p > .05$) across all problem sets except for multiplying decimals $F(5, 42) = 2.57$, $p < .05$, $\eta^2 = 0.23$. No significant results were observed when the problem sets were compiled or when specific delivery elements were isolated, and there was no significant difference between messaging conditions and the control. For the full sample, gender was found to differ significantly on persistence, $F(1, 568) = 3.84$, $p = 0.051$, $\eta^2 = 0.01$, with girls showing significantly more persistence ($M = 0.99$, $SD = 0.12$) across conditions than boys ($M = 0.96$, $SD = 0.20$). While girls were found to be approaching completion in all conditions ($p < .05$), boys showed lower completion overall, with the lowest performance apparent in the control condition.

When considering mastery speed, as defined by the number of questions required for problem set completion, ANOVA results suggested null results ($p > .05$) across all problem sets analyzed

individually. Further, no significant results were observed when problem sets were compiled or when specific delivery elements were isolated, and there was no significant difference between messaging conditions and control. Although there was no significant difference in mastery speed across genders, trends suggested that girls had faster mastery speed in general, requiring consistently fewer questions to complete problem sets regardless of condition (M = 4.25, SD = 2.65) than boys (M = 4.43, SD = 2.86). Means and standard deviations for the full sample are presented in Table 4.

ANOVA comparisons of the survey measures of mindset, enjoyment, and system helpfulness similarly conveyed null results within the full sample. The "mindset" variable was established from an average of two binary survey questions, with a composite score scaled from 0-2 representing the spectrum from fixed mindset (0) to growth mindset (2). The "enjoyment" variable was based on one question with Likert scale scores from 0-3, representing how much the student enjoyed their assignment. The "helpfulness" variable is represented in the same manner, based on the student's perception of how helpful the tutoring system was in completing their assignment. Null results were found for all three measures across problem sets when analyzed individually, and no significant differences were observed between conditions when problem sets were compiled or when specific delivery elements were isolated. Further, there was no significant difference between all messaging conditions and the control group. Gender was found to have a significant effect on enjoyment, regardless of condition $F(1, 552) = 19.50$, $p < .001$, $\eta^2 = 0.03$, with girls measuring more enjoyment on average (M = 1.84, SD = 0.81) than boys (M = 1.52, SD = 0.90). As shown by Table 4, the Control was found to be the most enjoyable condition, while WordArt was enjoyed significantly less ($p < .10$). Gender was also approaching significance on the mindset measure, $F(1, 552) = 3.31$, $p = 0.069$, $\eta^2 = 0.01$, with boys exhibiting a lower mindset in general (M = 0.93, SD = 0.78) than girls (M = 1.05, SD = 0.77). Gender was not found to have a significant effect on student's perception of tutor helpfulness.

In an attempt to answer our third research question, elements within message delivery were collapsed based on similarity to better understand if a certain processing channel (i.e., audio) was providing the main effect for messaging results. As noted briefly in results for persistence, mastery speed, and survey measures, researchers were not able to isolate any significant differences among delivery elements ($p > .05$).

While few significant findings were observed in the full sample, it became clear that many students were at ceiling in the math content and therefore showing high persistence (completion) in minimum mastery speed (three consecutive correct questions). When we reassessed the sample for students operationally defined as 'struggling,' or those who required more than three questions to complete their assignments, our analysis became a bit more informative. Among 253 student assignments, no significant differences were found among conditions in persistence or mastery speed ($p > .05$). However, findings suggested that it took struggling students less questions on average to reach mastery when in the audio condition (M = 5.59, SD = 2.00) compared to all other conditions, as shown in Table 5.

When considering gender, struggling boys exhibited lower mastery in conditions including audio ($p < .05$) yet were found to persevere more when an image of Jane was present, while girls persevered less with the female presence ($p < .05$). Survey results for struggling students suggested that boys exhibited the lowest mindset measures after experiencing the control condition ($p < .05$), and trends suggested that regardless of condition, girls exhibited the growth mindset more consistently (M = 1.00, SD = 0.79) than boys (M = 0.91, SD = 0.75). As with the primary analysis, trends suggested that boys exhibited the growth mindset after experiencing the animation condition ($p < .10$). It was also found that regardless of condition, girls enjoyed their assignments (M = 1.72, SD = 0.87) significantly more than boys (M = 1.42, SD = 0.92), $p < .05$, and that girls consistently found the tutoring system more helpful in completing their assignment (M = 2.10, SD = 0.83) than did boys (M = 1.92, SD = 0.90).

Table 4. Means and Standard Deviations for Persistence, Mastery Speed, and Survey Measures Across Control and Messaging Conditions for All Students.

	Control (104^a, 99^b)		All Messaging (517^a, 506^b)		Animation (106^a, 103^b)		Static Image with Text (116^a, 113^b)		Static Image with Audio (117^a, 115^b)		Word Art ($90^{a,b}$)		Audio (88^a, 85^b)	
Analysis	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Persistence	0.95	0.21	0.98	0.14	0.97	0.17	0.97	0.16	0.98	0.13	**1.00**	**0.00**	0.97	0.18
Mastery Speed	**4.74**	**3.35**	4.32	2.67	4.24	2.69	4.62	2.83	4.32	2.42	4.28	3.33	4.09	1.91
Mindset	**1.06**	**0.81**	0.96	0.78	1.01	0.80	0.96	0.77	1.02	0.77	1.00	0.79	0.78	0.75
Enjoyment	**1.83**	**0.80**	1.67	0.89	1.74	0.87	1.66	0.90	1.77	0.82	1.49	0.91	1.67	0.96
Helpfulness	1.99	0.85	1.94	0.86	1.86	0.89	**2.01**	**0.89**	**2.01**	**0.77**	1.82	0.79	1.95	0.95

[a]Sample size for Persistence and Mastery Speed.
[b]Sample size for Mindset, Enjoyment, and Helpfulness.
Note. "Mindset" is measured by two questions (0 = Fixed Mindset, 1 = Growth Mindset) and scores are compiled. "Enjoyment" is measured by one question (Likert Scale, 0-3). "Helpfulness" is measured by one question (Likert Scale, 0-3).

Table 5. Means and Standard Deviations for Persistence, Mastery Speed, and Survey Measures Across Control and Messaging Conditions for Struggling Students.

	Control (46^a, 45^b)		All Messaging (207^a, 204^b)		Animation (42^a, 41^b)		Static Image with Text (49^a, 47^b)		Static Image with Audio ($49^{a,b}$)		Word Art ($28^{a,b}$)		Audio ($39^{a,b}$)	
Analysis	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Persistence	0.98	0.15	0.99	0.12	0.98	0.15	0.96	0.20	**1.00**	**0.00**	**1.00**	**0.00**	**1.00**	**0.00**
Mastery Speed	7.07	3.95	6.34	3.32	6.17	3.48	6.84	3.24	6.14	2.88	**7.11**	**4.95**	5.59	2.00
Mindset	0.93	0.75	0.95	0.78	1.00	0.81	0.89	0.73	**1.04**	**0.82**	0.82	0.86	0.92	0.70
Enjoyment	1.60	0.86	1.58	0.94	**1.76**	**0.92**	1.45	1.00	1.71	0.79	1.43	1.07	1.51	0.97
Helpfulness	1.98	0.92	2.01	0.87	1.98	0.94	1.98	0.82	2.04	0.87	2.00	0.82	**2.05**	**0.94**

[a]Sample size for Persistence and Mastery Speed.
[b]Sample size for Mindset, Enjoyment, and Helpfulness.
Note. "Mindset" is measured by two questions (0 = Fixed Mindset, 1 = Growth Mindset) and scores are compiled. "Enjoyment" is measured by one question (Likert Scale, 0-3). "Helpfulness" is measured by one question (Likert Scale, 0-3).

Approximately 60% of students in the full sample exhibited the growth mindset in their survey responses, regardless of condition. Noting Table 4, students in the control condition actually reported the highest levels of growth mindset (M = 1.06, SD = 0.81), with those in the audio condition reporting the lowest levels (M = 0.78, SD = 0.75). Among struggling students, the highest levels of growth mindset were reported by students in the static image with audio condition (M = 1.04, SD = 0.82), while those in the word art condition reported the lowest levels (M = 0.82, SD = 0.86). Responses to measures of enjoyment and helpfulness followed normal distributions, with approximately 60% finding the assignments at least "somewhat" enjoyable, and approximately 78% finding the tutoring system at least "somewhat" helpful.

5. DISCUSSION

Within the current study, the addition of motivational messaging to the ASSISTments tutor did not significantly affect the likelihood of student persistence or mastery speed. Further, there was little evidence that the motivational messages had the intended effect on mindset within the full sample. Trends suggested that those in messaging conditions experienced a slight increase in persistence and a decrease in mastery speed in comparison to those in the control condition. However, students in the messaging conditions also exhibited consistently lower levels for measures of mindset, enjoyment of the assignment, and perception of system helpfulness. A larger student population would be required to discern a truly significant effect within these trends.

Interestingly, struggling students appeared to benefit from the presence of messages, showing an increase in persistence, a decrease in mastery speed, and slightly increased measures of the growth mindset. It can be argued that struggling students, or those facing a challenge, are most in need of motivational interventions, and that they are more likely to respond to messaging, regardless of condition. Motivational messages produced distinctly higher adoption of the growth mindset in struggling students who experienced the static image with audio condition. Thus when designing motivational content for struggling students, current findings promote the addition of audio as an alternative processing channel to assist students. Researchers were not able to pinpoint an optimal processing channel for the delivery of growth mindset messages when targeting the general population.

One participating teacher requested that her students use a feature within the tutoring system to comment on their experience while completing their assignment. Feedback was predominantly negative, with students citing the messages as distracting or confusing. One student specifically questioned why the animated learning companion simply repeated messages rather than helping to solve the problems. This suggests that students are familiar with systems that utilize pedagogical agents, and that they have developed expectations for characters that are associated with learning. This echoes the argument set forth by Kapoor, et al. [9] regarding the necessity for tutors to provide appropriate cognitive and affective responses, and aids in the design of tutoring systems hoping to incorporate learning companions.

This study had a variety of limitations. The ASSISTments math content chosen due to popular usage lead to a high percentage of ceiling effects within the sample. Teachers assigned multiple problem sets to their students, often as review. Thus, many students easily mastered the content intended for lower grades and thereby skewed rates of persistence and mastery speed. Further, the null effects found in the full sample raise important questions regarding the generalizability of mindset interventions outside of struggling student populations. Within the context of an adaptive mathematics tutor, students who appear to be at ceiling in math content may not require motivational messaging, and it may become detrimental to the learning process.

We also note that approximately 18.8% of students reported having technical difficulties and were removed prior to analysis. The incompatibility of simple HTML files serves as a reminder that many classrooms struggle to maintain up-to-date technological resources. Students are often required to share computers or iPads that come equipped with outdated software and generally slow internet connections. Future research should incorporate allowance for these issues within the experimental design, as incompatibilities may lead to selection bias.

It is also difficult to justify whether or not students consistently attended to the motivational messages. As students were simply presented the messages and were not asked to respond in any manner, the levels of message internalization may be broad. We also note that the duration of the intervention may have been too short to observe reliable differences among messaging conditions. In much of her work, Dweck has provided longer interventions upfront, coupled with 'reminders' such as the messages used in the current study [7]. Further, her studies often run longitudinally across the course of a school year or more. Still, regardless of condition, the majority of students in our sample exhibited the growth mindset. Future research should include a pretest mindset survey to determine if these results can be credited solely to the motivational messages provided throughout the learning experience.

Finally, it should be noted that researchers relied on the tutoring system to perform random assignment. While prior research has suggested that this practice is sound, assignment for this study appears to have favored the static image with audio condition. Future research using ASSISTments should take this bias into consideration.

Future iterations of this study should focus on struggling students, or those undertaking challenging academic tasks. Future research should also seek to assess these conditions in an even more adaptive environment. It seems as though students were not reaping the benefits of the "persona effect" found in prior research [1], due to a lack of bonding with the agent. A truly adaptive agent, one consistently present and building rapport, may be more effective in message delivery. Rather than repeating the same select set of general and incorrect attributions, struggling students may require motivational messages linked with the tutor content and their progress. Perhaps just as a pedagogical agent, these messages must be fine-tuned to a student's cognitive and affective states. Alternative message delivery methods, including video feedback with human tutors used as hints, scaffolding, and misconception messages, should also be considered in future research.

6. ACKNOWLEDGEMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's "STEM Grand Challenges," and IES (#R305A120125, R305C100024). Thanks to S. & L.O.

7. REFERENCES

[1] Arroyo, I., Burleson, W., Tai, M., Muldner, K., Woolf, B.P. 2013. Gender differences in the use and benefit of advanced learning technologies for mathematics. Journal of Educational Psychology. 105, 4, 957-969.

[2] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A. & Koedinger, K. 2008. Why students engage in "Gaming the System" behavior in interactive learning environments. *Journal of Interactive Learning Research*. 19, 2, 185-224.

[3] Bernacki, M. L., Nokes-Malach, T. J., & Aleven, V. 2013. Fine-grained assessment of motivation over long periods of learning with an intelligent tutoring system: Methodology, advantages, and preliminary results. In *International handbook of metacognition and learning technologies*. Springer New York. 629-644.

[4] Clark, R.C. & Mayer, R. E. 2003. e-Learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning. San Francisco, CA: Pfeiffer.

[5] Dweck, C.S. 2002. Messages that motivate: how praise molds students' beliefs, motivation, and performance (in surprising ways). Improving Academic Achievement: Impact of Psychological Factors in Education. Ed. Joshua Aronson. New York.

[6] Dweck, C.S. 2006. Mindset: The new psychology of success. Random House.

[7] Dweck, C.S. 2013. Mindsets: Helping Students Fulfill Their Potential. Smith College Lecture Series, North Hampton, MA. September 19.

[8] Graesser, A., Chipman, P., King, B., McDaniel, B., & D'Mello, S. 2007. Emotions and learning with autotutor. *Frontiers in Artificial Intelligence Applications*, *158*, 569.

[9] Kapoor, A., Burleson, W., & Picard, R. W. 2007. Automatic prediction of frustration. International Journal of Human-Computer Studies. 65, 724-736.

[10] Mayer, R.E. 2014. Incorporating motivation into multimedia learning. Learning and Instruction. Volume 29, 171-173.

[11] Mueller, C. & Dweck, C. 1998. Praise for Intelligence Can Undermine Children's Motivation and Performance. Journal of Personality and Social Psychology, Vol. 75, No. 1, 33-52.

[12] Ostrow, K.S. 2013. Motivational Message Study. Accessed 12/12/2013. Student Experience, RCT & All Data: https://sites.google.com/site/korinnostrow/research

[13] Reeve, J. 2009. Understanding motivation and emotion. (5th ed.). Hoboken, NJ: Wiley.

Toward Adaptive Unsupervised Dialogue Act Classification in Tutoring by Gender and Self-Efficacy

Aysu Ezen-Can
Department of Computer Science
North Carolina State University
aezen@ncsu.edu

Kristy Elizabeth Boyer
Department of Computer Science
North Carolina State University
keboyer@ncsu.edu

ABSTRACT

For tutorial dialogue systems, classifying the dialogue act (such as questions, requests for feedback, and statements) of student natural language utterances is a central challenge. Recently, unsupervised machine learning approaches are showing great promise; however, these models still have much room for improvement in terms of accuracy. To address this challenge, this paper presents a new unsupervised dialogue act modeling approach that leverages non-cognitive factors of gender and self-efficacy to better model students' utterances during tutorial dialogue. The experimental findings show that for females, leveraging learner characteristics within dialogue act classification significantly improves performance of the models, producing better accuracy. This line of investigation will inform the design of next-generation tutorial dialogue systems, which leverage machine-learned models to adapt to their users with the help of non-cognitive factors.

Keywords

Tutorial dialogue, learner characteristics, dialogue act classification, unsupervised machine learning, adaptive learning.

1. INTRODUCTION

Tutorial dialogue is a highly effective form of instruction, and much of its benefit is thought to be gained from the rich natural language dialogue exchanged between tutor and student [7, 17, 36]. In order to model tutorial dialogue for the purposes of building tutorial systems or for studying human tutoring, *dialogue acts*, which capture both cognitive and non-cognitive aspects of dialogue utterances, provide a valuable level of representation. Dialogue acts represent the underlying intention of utterances (for example, to ask a question, agree or disagree, or to give a command) [3, 32]. Within the computational linguistics and dialogue systems literature, automatically classifying dialogue acts has been a focus of research for several decades [6, 14, 35]. For tutorial dialogue systems, dialogue act classification is crucial to understanding students' utterances and developing tutorial strategies [8, 24].

Today's tutorial dialogue systems utilize a variety of dialogue act classification strategies, some rule-based and some statistical [13]. Historically when machine learning has been used to devise tutorial dialogue classifiers, these have been *supervised* classifiers, which require training on a manually labeled corpus. The same is true within the broader dialogue systems research community: dialogue act classifiers have historically either been handcrafted and rule-based, or learned with supervised machine learning techniques [11, 14, 22, 29]. However, supervised techniques face substantial limitations in that they are labor-intensive due to the manual annotation and handcrafted dialogue act taxonomies that are usually domain-specific. To overcome these challenges, unsupervised dialogue act modeling techniques including hidden Markov models [20, 21, 30], Dirichlet Process clustering [12, 23], *k*-means clustering [31], and query-likelihood clustering [15] have been investigated in recent years.

Despite this growing focus on developing unsupervised dialogue act classifiers, these models still underperform compared to supervised approaches in their accuracy for classifying according to manual tags. However, while unsupervised models to date have considered such things as lexical features (the words found in the utterance) and syntactic features (the structure of the sentence), they have not considered non-cognitive factors, such as gender and self-efficacy, which are believed to influence the structure of tutorial dialogue [10]. Cognitive factors such as skill mastery has been widely studied in learning environments. However, there is a smaller body of work on adaptive learning environments using non-cognitive factors. A variety of learner characteristics, including non-cognitive factors, play an influential role in learning, not only in tutoring but in classroom settings [1], and in web-based courses [19]. Prior work on learner characteristics has focused on building adaptive systems based on different user groups [16], tutorial feedback selection [9] and identifying students that need remedial support [27]. Identifying clusters of student characteristics is also an active area of research [4, 25–27].

This paper investigates whether the performance of an unsupervised dialogue act classifier can be improved by taking these factors into account. Because non-cognitive factors are shown to affect language, we believe that training dialogue act classifiers tailored to specific learner characteristics can help tutorial dialogue systems to understand students better. We utilize two learner characteristics: gender, as self-reported by students on a survey and domain-specific self-efficacy, as measured by a validated instrument for determining a student's confidence in her own abilities. Specifically, we train unsupervised dialogue act models that are tailored to students of specific gender and self-efficacy level, and we compare those models to corresponding ones trained without restricting by that learner characteristic. This unsupervised training is conducted entirely without the use of manual tags. We then test all of the models on held-out test sets within leave-one-student-out cross validation, and compare the resulting classification accuracy according to their previously applied manual tags. The results show that for female students, utilizing learner characteristics statistically significantly improves dialogue act classification models. For self-efficacy groups, improvement is observed but not at a statistically reliable level. This paper constitutes the first research toward incorporating non-cognitive factors into unsupervised dialogue act classifiers for

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

tutorial dialogue with the overarching goal of providing personalized learning for students. We first administered a survey to collect these characteristics via self-report, and then learned a dialogue act classifier tailored to those characteristics. These results can inform the way that next-generation tutorial dialogue systems conduct their real-time dialogue act classification and language adaptation.

2. RELATED WORK

Dialogue act modeling is an important level of representation within dialogue systems. Following theories proposed several decades ago within philosophy and linguistics [3, 32], dialogue act classification aims to capture the intention of an utterance; for example, in tutoring some dialogue acts involve asking questions or giving or requesting feedback. While a long-standing line of investigation has focused on handcrafted or supervised machine learning techniques for dialogue act classification [11, 14, 22, 29], only recently is a body of work emerging on unsupervised approaches to this problem. Most of this work has been done outside of educational domains, with a proposed hidden Markov model in the domains of Twitter posts [30] and emails [21], Dirichlet Process Mixture Models for a train fare dialogue domain [12] and for navigating buildings [23], and a Chinese Restaurant Process approach for spoken Japanese [20].

Another important difference between the current work and prior research is in the features used, namely the non-cognitive characteristics of gender and self-efficacy. Prior work has used a variety of features for performing supervised dialogue act classification, including prosodic and acoustic features which involve the profile of the sound signal itself [35], lexical features such as words and sequences of words [34], syntactic features including part-of-speech tags [6, 24], dialogue structure features such as taking the initiative and the previous dialogue act [33] as well as task/subtask features in tutorial dialogue [8, 18]. Within unsupervised dialogue act classification a subset of these features have also been used such as words [12], state transition probabilities in Markov models [23], topic words [30], function words [15], a smaller subset of words containing beginning portions of utterances [31], part-of-speech tags and dependency trees [21]. While a variety of experiments have demonstrated the utility of these features in several domains, no prior work has reported on an attempt to include the factors considered here, in order to improve the performance of an unsupervised dialogue act classifier. To investigate this, we build dialogue act classifiers that learn from utterances of specific learner groups and predict dialogue acts of students according to their learner characteristics.

3. CORPUS

The corpus used in this study consists of student-tutor interactions in an introductory computer science programming task [18]. Throughout the data collection, freshman engineering students and tutors communicated through a textual dialogue-based learning environment while working on Java programming. The ethnicity of students participated in this study is distributed as follows: 26 white, 9 Asian, 3 Latino, 2 African American, 1 Middle Eastern and 1 Asian American. An excerpt from the corpus is shown in Table 1.

Students were given a pre-survey that included survey items on computer science self-efficacy, such as 'I am sure I can learn programming'. This self-efficacy scale was adapted directly from the Domain-specific Self-Efficacy Scale [5], with five items measured on a Likert scale from 1-5 (1 being lowest self-efficacy, 5 being highest). Students also completed a demographic questionnaire from which gender was obtained. For self-efficacy, students were divided into classes based on the median score across all students on that scale. Along with gender, this produces two partitions of the 42 students: females (12) and males (30), low (24) and high self-efficacy students (18).

Table 1: Excerpt of dialogue with a *male* student in the *low self-efficacy* group

Role	Utterance	Dialogue Act
Tutor	You'll need to end every Java statement with a semi colon	*S*
Student	Got it!	*ACK*
Tutor	This is to let Java know where each statement ends	*S*
Tutor	Ah no prompt!	*S*
Tutor	Why do you think that is?	*Q*
Student	I wish I knew...	*A*
Student	I don't think I spelled anything wrong	*S*
Tutor	Ah it's actually pretty easy	*S*
Tutor	The order of the lines matters	*S*

The corpus containing 1640 student utterances was manually annotated with dialogue act tags in previous work [18] (Table 2). These dialogue act tags are not available during model training, but we use them for evaluation purposes to calculate accuracy on a held-out testing set.

Table 2: Student dialogue acts and distributions

Student Dialogue Act	Example	Distribution
A (answer)	*yeah I'm ready!*	39.95%
ACK (acknowledgement)	*Alright*	21.31%
S (statement)	*i am taking basic fortran right now never seen literal before*	21.20%
Q (question)	*what does that mean?*	15.15%
RF (request feedback)	*better?*	0.98%
C (clarification)	**html messing*	0.79%
O (other)	*haha*	0.61%

4. DIALOGUE ACT MODELING BASED ON LEARNER CHARACTERISTICS

We hypothesize that dialogue act models built using unsupervised machine learning will perform substantially better when customized to specific learner groups. Specifically, we investigate whether by training a model only on students of a particular learner characteristic, that model would perform significantly better at predicting the dialogue acts of unseen students with the same learner characteristic compared to a model that was trained on students of all learner characteristics.

We note that because the same corpus is being partitioned in two different ways, the same student will occur in one of the gender groups and in one of the self-efficacy groups. This choice to partition in 2-way splits rather than *2n*-way splits where *n* is the number of learner characteristics is because of issues that arise with sparsity. This interdependence between partitions is a limitation to note; however, as discussed in Section 5, this

interdependence can be taken into account for making decisions within a tutorial dialogue system by employing a suite of classifiers within a voting scheme.

4.1 Experimental Design

For gender and self-efficacy, we will test whether an unsupervised dialogue act classifier trained only on students with that characteristic outperforms a classifier that is not specialized by this characteristic. In order to gather accuracy data across these characteristics, we conduct leave-one-student-out training and testing folds. The testing set for each of the *n* folds (where *n* varies depending on which learner group is being considered) consists of all of a single student's dialogue utterances and the model is trained on the remaining *n-1* students. The average number of utterances per student in the corpus is 36.8 (σ=12.07; min=16; max=64). These are therefore the average, minimum, and maximum number of utterances across the leave-one-student-out test sets.

We compute the average test set performance of the model across all folds for each non-cognitive characteristic partition. The performance metric utilized in this study is *accuracy* compared to the manually labeled dialogue acts described in the previous section, where accuracy is computed as the number of utterances in the test set that were classified according to their manual label, divided by the number of utterances total in the test set. As described in 4.2, the process of labeling via unsupervised classification involves taking the majority vote within each cluster.

For constructing the folds, we take an approach to balance the sample size available to model training. This balancing approach is needed to ensure that each model is trained on a similar size of data. Consider, for example, the partition of gender. Without a balanced sampling approach the leave-one-student-out testing folds for the un-specialized classifier for female students would include n_{female}=12 test folds but the available data for each training fold would be n_{total}-1 = 41. In contrast, the specialized classifier trained only on female students would still include n_{female}=12 test points but the available data for each training fold would be n_{female}-1 = 11. Therefore, each un-specialized classifier was trained on a randomly selected subset of the corpus. In the case of females, each of the 12 testing folds will utilize a model trained on 11 data points. The specialized classifier will use 11 female data points, and the un-specialized classifier will use 11 randomly selected data points. In this way, we investigate how well a model predicts dialogue acts of a student with and without utilizing learner characteristic information.

4.2 Unsupervised Dialogue Act Models

Our unsupervised dialogue act classification approach leverages the *k*-medoids clustering technique [28]. This approach groups similar utterances together, and is similar to the more familiar *k*-means algorithm except that in *k*-medoids, the centroid of each cluster must be an actual data point within the corpus rather than a potentially artificial data point computed as the mean of distances. Our experiments with *k*-medoids have demonstrated that it outperforms a variety of other unsupervised machine learning approaches for the task of dialogue act classification in tutorial dialogue, although the results of such experiments are beyond the scope of this paper since our goal is to investigate the *differential benefit* of adding learner characteristic features to the model, not to compare different unsupervised approaches.

The *k*-medoids algorithm requires seeding clusters at the beginning of each training fold and then proceeds by distributing data points to clusters according to their closest centroids until convergence upon the model. In the standard *k*-medoids algorithm, the seeds are randomly selected. However, we employ a greedy seed selection approach intended to mitigate the effects of the unbalanced distribution of dialogue acts in the corpus [2]. Within this greedy seed selection, an initial seed is randomly selected and then each of the subsequent seeds are selected by choosing the point that maximizes its distance from the already-selected seeds. The goal in using this approach is to select the seeds from diverse utterances so the algorithm produces better clusters, and our initial experiments indicated that it substantially improves the model.

In addition to its seeding approach, the *k*-medoids approach requires the number of clusters *k* to be set prior to model training. To discover the number of clusters, we experimented with *X*-Means and Expectation Maximization clustering, both of which attempt to identify the optimal number of clusters. Both of these algorithms converged at four clusters as the optimal choice, so we proceed with *k*=4. However, perhaps in part due to the benefit of the greedy seed selection made possible by *k*-medoids, these models performed with substantially worse overall accuracy than *k*-medoids.

The utterances were represented as vectors with each column matching a token (punctuation and words) in the corpus and each row matching an utterance. There were a total of 877 distinct tokens.

With these parameters in place, first the clusters were formed using each training set, and then for each utterance of the student held out within the leave-one-student-out fold, we computed the closest cluster to that utterance as indicated by average cosine distance to each point in the cluster. The closest cluster was selected as the cluster to which the test utterance belongs, and the majority vote of the cluster was assigned to the test utterance as its dialogue act label. For each leave-one-student-out testing fold, the accuracy was computed by comparing these cluster-assigned labels to the manual dialogue act tags.

4.3 Experimental Results

This section presents experimental results for unsupervised dialogue act classification based on learner characteristics. We compare each model built separately by gender and self-efficacy level to the models that are built using utterances from randomly selected students, *i.e.* not utilizing learner characteristic information. Each comparison in this section is conducted with a one-tailed *t*-test with a post-hoc Bonferroni correction. The threshold for statistical reliability after the correction has been taken as α=0.05.

Gender. As shown in Figure 1, the average leave-one-student-out cross-validation accuracy for the model built using female students' utterances (n_{female}=12) is higher than the model built on randomly selected students. In each test run, all of one female's utterances were left out to be used as the test set, and the dialogue act model was built on the remaining eleven female students' utterances. This process was repeated for each female student. Note that for each of the eleven students, all utterances from that student were considered. Average test set accuracy for the model with randomly selected students was 0.41 (σ=0.2), whereas the average test set accuracy for the dialogue act classification model that was built utilizing female students' utterances only was 0.56 (σ=0.19). After a Bonferroni correction this difference was statistically significant (p_{Bonf}<0.05).

For male students (n_{male}=30), the average accuracy is only slightly higher with the models tailored to males 0.43 (σ =0.13) than the models learned for randomly selected students 0.40 (σ =0.12), and this difference is not statistically significant (Figure 1). Looking more closely at the results, we find that for eight of the thirty males within the corpus, a tailored model outperformed the random model (with five of these seeing more than 10% increase in accuracy), while twenty-two of the cases saw no difference in accuracy between the random and tailored conditions. Two of the males saw a decrease in accuracy for the tailored condition.

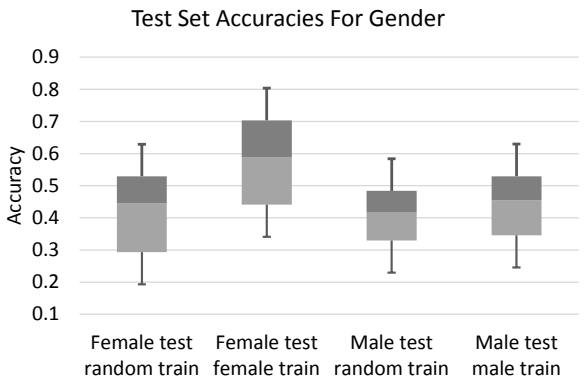


Figure 1: Leave-one-student-out test set accuracies for models by gender

Self-efficacy. Models built using the self-efficacy learner characteristic predict the unseen utterances' dialogue acts marginally more successfully than models that do not use this information, though these differences are not statistically reliable. For students with low self-efficacy (n_{lowEff}=24) the average test set accuracy for dialogue act models that selected students randomly is 0.38 (σ=0.16) and it increases to 0.43 (σ=0.17) with dialogue act models that learn only from low-self-efficacy students' utterances (Figure 2). In fifteen out of twenty-four cases the dialogue act models tailored to low self-efficacy groups outperform models that are trained on randomly selected students (eight of the cases with more than a 10% increase), while in seven of the cases the performance is decreased by utilizing the learner characteristic (five of them by more than a 5%) and in two of the cases the accuracy remains the same.

The improvement obtained by utilizing learner characteristics in dialogue act classification task is also marginal for high-self-efficacy students, where $n_{highEff}$=18. The average performance for the random model is 0.41 (σ=0.14) whereas the model achieves 0.47 (σ=0.11) accuracy when trained only on utterances of high-self-efficacy students. This improvement was statistically significant before Bonferroni correction but not afterward. In seven out of eighteen cases, models trained on utterances of high self-efficacy students improved test set accuracy (five of them above 15% improvement) and in two of the cases the learner characteristic decreases the performance (both of them below 5% decrease). Nine of the cases remained unaffected in their dialogue act classification accuracy.

The average accuracies over the leave-one-student-out cross-validation folds can be found in Table 3. Models tailored to learner groups uniformly outperform their counterpart, and the improvement is statistically significant for females.

Figure 2: Leave-one-student-out test set accuracies for models by self-efficacy

Table 3: Average test set accuracies for each learner characteristic (p<0.05 after Bonferroni correction)**

Learner characteristic group	Model restricted by learner characteristic	Model built on randomly selected students
Females	**0.56****	0.41
Males	**0.43**	0.40
Low self-efficacy	**0.43**	0.38
High self-efficacy	**0.47**	0.41

5. DISCUSSION

Dialogue act classification is a central task for tutorial dialogue systems. Without accurate dialogue act classification, systems cannot adapt and respond appropriately. Unsupervised machine learning approaches to dialogue act classification are a highly promising new area of study, and we have presented the first unsupervised dialogue act classifier tailored to learner characteristics. The experimental results demonstrated that dialogue act classifiers that leverage the non-cognitive factors of gender and self-efficacy outperform those that do not, and in the case of female students the improvement was statistically significant. This section presents some examples of the learned dialogue act clusters and discusses the implications of this work for tutorial dialogue systems.

First, we examine clusters from the gender-tailored unsupervised dialogue act classifier. Table 4 displays a selection of utterances that were clustered together during the unsupervised training of the model, and afterward the clusters were labeled for testing purposes using the manual tags that comprise the majority of each cluster. For those in Table 4 the clusters were labeled as Acknowledgments and Questions. By examining the structure of these clusters we gain some intuition as to the types of regularities that help the tailored models to perform significantly better. We see females in this study tended to use acknowledgment phrases such as, "oh I see" and "makes sense," while males tended to use the phrasing, "got it" more frequently. Within the cluster labeled as questions, we observe that females tended to request more feedback, an observation that also emerged in prior work within a different corpus in the same domain collected approximately six years earlier [10]. On the other hand, male students tended to ask more general questions.

In addition, we observe some example clusters from the models based on self-efficacy in Table 5. Students with high self-efficacy tend to use more confident utterances such as "absolutely" compared to "ok" used by low-self efficacy students. We note that questions in the low self-efficacy group often make an implicit

request for reassurance within their task-based questions, such as, "and that is it?". In contrast, students in the high self-efficacy group more often ask contentful questions.

Table 4: Selected utterances from clusters tailored to gender

	Females	Males
Acknowledgements	- oh I see - make sense - yup - aha! -hahaha its ok	- got it - ok i got it - alright i got it - gotcha alrigth - cool - sure thing
Questions	-is this right? -does that work? -should I run it? -was i supposed to put that before something? -so for line number could i have typed system out println monopoly instead of println x if i wanted to?	-so will testing always be related to running the program -so it is kinda like saying x number or something in algebra? -why does not it stop on the next line in this case

Table 5: Selected utterances from clusters tailored to self-efficacy

	Low Self-Efficacy	High Self-Efficacy
Acknowledgements	- ok - yes there were a lot of things i felt like i had to switch around - that makes sense now	-cool! -oh ok that works - yep got that - absolutely
Questions	-so what exactly am i supposed to be doing? - is there something specific i need to call my game - i finished reading should i click compile again? -and that is it?	-what is the best way to do that? - ok so tell me if this makes sense string declares the variable and then line number tells me what that variable is value is?

Limitations. The present work has several notable limitations. First, as mentioned previously, the partitions of the corpus are not independent; that is, the same student, and associated utterances, are present within one gender group and one self-efficacy group. Because these partitions are not independent, care must be taken when interpreting the findings. Furthermore, it is possible that the self-efficacy of students can change in the course of tutoring, which would not be handled by a classifier built using a one-time self-report. However, we believe that the current approach holds great promise for real-time tutorial dialogue classification. By building separate classifiers by learner characteristic, a suite of classifiers (each smaller and faster than one built on the entire corpus) can be run in parallel and can vote for the classification of a given students' utterance. However, as is the case with the work presented here, splitting the corpus results in a substantially reduced sample size on which to train, which partially explains the lack of statistically reliable results observed here. Our work has begun to explore the use of intrinsic metrics for accuracy (rather than relying on manual tags), which has the potential to dramatically increase the available data to any dialogue act classifier and mitigate issues of sparsity that arise when splitting by learner characteristics.

6. CONCLUSION AND FUTURE WORK

More accurately understanding student natural language within intelligent tutoring systems is a critical line of investigation for tutorial dialogue systems researchers. The field has only begun to explore unsupervised approaches and to investigate the range of features that are beneficial within this paradigm. We have presented a first attempt to leverage non-cognitive factors within such a dialogue act classification model, achieving statistically significant improvements in dialogue act modeling for female students, and increasing the models' performance by small margins for the self-efficacy groups.

Building upon these first steps, there are several promising future directions. First, while sample size prohibited exploring some other learner characteristics here, other characteristics are likely highly influential and should be investigated. These may include ethnicity, personality, and other non-cognitive factors. Additionally, while the current work focused on analyzing dialogue, another aspect of the tutorial interaction that presents challenges in understanding is the task model. Models that aim to understand students' problem-solving activities and infer their goals or plans may benefit substantially from leveraging learner characteristics. It is hoped that the research community can continue to build richer models of natural language understanding for students of all learner characteristics in order to improve the student experience and enhance learning by adaptation.

ACKNOWLEDGMENTS

The authors wish to thank the members of the Center for Educational Informatics at North Carolina State University for their helpful input. This work is supported in part by the National Science Foundation through Grant DRL-1007962 and the STARS Alliance, CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Ames, C. and Archer, J. 1988. Achievement Goals in the Classroom: Students' Learning Strategies and Motivation Processes. *Journal of Educational Psychology*. 80, 3, 260–267.

[2] Arthur, D. and Vassilvitskii, S. 2007. k-means++: The Advantages of Careful Seeding. In *Proceedings Of The Eighteenth Annual ACM-SIAM Symposium On Discrete Algorithms*. 1027–1035.

[3] Austin, J.L. 1962. *How To Do Things With Words*. Oxford University Press.

[4] Azarnoush, B., Bekki, J.M. and Bernstein, B.L. 2013. Toward a Framework for Learner Segmentation. *JEDM*. 5, 2, 102–126.

[5] Bandura, A. 2006. Guide for Constructing Self-Efficacy Scales. *Self-efficacy Beliefs Of Adolescents*. 5, 307–337.

[6] Bangalore, S., Di Fabbrizio, G. and Stent, A. 2008. Learning the Structure of Task-Driven Human-Human Dialogs. *IEEE Transactions on Audio, Speech and Language Processing*. 16, 7, 1249–1259.

[7] Bloom, B.S. 1984. Sigma of Problem: The Methods Instruction One-to-One Tutoring. *Educational Researcher*. 4–16.

[8] Boyer, K.E., Ha, E.Y., Phillips, R., Wallis, M.D., Vouk, M.A. and Lester, J.C. 2010. Dialogue Act Modeling in a Complex Task-Oriented Domain. In *Proceedings of SIGDIAL*. 297–305.

[9] Boyer, K.E., Phillips, R., Wallis, M., Vouk, M. and Lester, J. 2008. Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue. In *Proceedings of ITS*, 239–249.

[10] Boyer, K.E., Vouk, M.A. and Lester, J.C. 2007. The Influence of Learner Characteristics on Task-Oriented Tutorial Dialogue. In *Proceedings of AIED,* 365–372.

[11] Buckley, M. and Wolska, M. 2008. A Classification of Dialogue Actions in Tutorial Dialogue. In *Proceedings of the 22nd International Conference on Computational Linguistics*. 1, 73–80.

[12] Crook, N., Granell, R. and Pulman, S. 2009. Unsupervised Classification of Dialogue Acts Using a Dirichlet Process Mixture Model. In *Proceedings of SIGDIAL*. 341–348.

[13] Dzikovska, M.O., Farrow, E. and Moore, J.D. 2013. Combining Semantic Interpretation and Statistical Classification for Improved Explanation Processing in a Tutorial Dialogue System. In *Proceedings of AIED*. 279–288.

[14] Eugenio, B. Di, Xie, Z. and Serafin, R. 2010. Dialogue Act Classification , Higher Order Dialogue Structure, and Instance-Based Learning. *Dialogue & Discourse*. 1, 2, 1–24.

[15] Ezen-Can, A. and Boyer, K.E. 2013. Unsupervised Classification of Student Dialogue Acts With Query-likelihood Clustering. In *Proceedings of EDM*, 20–27.

[16] Forbes-Riley, K. and Litman, D.J. 2009. A User Modeling-Based Performance Analysis Of A Wizarded Uncertainty-Adaptive Dialogue System Corpus. In *Proceedings of INTERSPEECH*, 2467–2470.

[17] Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P. and Kreuz, R. 1999. AutoTutor: A Simulation Of A Human Tutor. *Cognitive Systems Research*. 1, 1, 35–51.

[18] Ha, E.Y., Grafsgaard, J.F., Mitchell, C.M., Boyer, K.E. and Lester, J.C. 2012. Combining Verbal and Nonverbal Features to Overcome the 'Information Gap' in Task-Oriented Dialogue. In *Proceedings of SIGDIAL*, 247–256.

[19] Hershkovitz, A. and Nachmias, R. 2011. Online Persistence In Higher Education Web-Supported Courses. *The Internet and Higher Education*. 14, 2, 98–106.

[20] Higashinaka, R., Kawamae, N., Sadamitsu, K., Minami, Y., Meguro, T., Dohsaka, K. and Inagaki, H. 2011. Unsupervised Clustering of Utterances Using Non-Parametric Bayesian Methods. In *Proceedings of INTERSPEECH*, 2081–2084.

[21] Joty, S., Carenini, G. and Lin, C.-Y. 2011. Unsupervised Modeling Of Dialog Acts In Asynchronous Conversations. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. 1807–1813.

[22] Keizer, S., Akker, R. and Nijholt, A. 2002. Dialogue Act Recognition with Bayesian Networks for Dutch Dialogues. In *Proceedings of the SIGDIAL Workshop*, 88–94.

[23] Lee, D., Jeong, M., Kim, K., Ryu, S. and Geunbae, G. 2013. Unsupervised Spoken Language Understanding for a Multi-Domain Dialog System. *IEEE Transactions On Audio, Speech, and Language Processing*. 21, 11, 2451–2464.

[24] Marineau, J., Wiemer-Hastings, P., Harter, D., Olde, B., Chipman, P., Karnavat, A., Pomeroy, V., Rajan, S. and Graesser, A. 2000. Classification of Speech Acts in Tutorial Dialog. In *Proceedings of the Workshop On Modeling Human Teaching Tactics And Strategies at ITS*. 65–71.

[25] Meece, J.L. and Holt, K. 1993. A Pattern Analysis Of Students' Achievement Goals. *Journal Of Educational Psychology*. 85, 4, 582–590.

[26] Merceron, A. and Yacef, K. 2003. A Web-Based Tutoring Tool With Mining Facilities to Improve Learning and Teaching. In *Proceedings of AIED*, 201–208.

[27] Merceron, A. and Yacef, K. 2005. Clustering Students To Help Evaluate Learning. *Technology Enhanced Learning*. 171, 31–42.

[28] Ng, R.T. and Han, J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, 144–155.

[29] Reithinger, N. and Klesen 1997. Dialogue Act Classification Using Language Models. In *Proceedings of EuroSpeech*, 2235–2238.

[30] Ritter, A., Cherry, C. and Dolan, B. 2010. Unsupervised Modeling of Twitter Conversations. In *Proceedings of the Association for Computational Linguistics*, 172–180.

[31] Rus, V., Moldovan, C., Niraula, N. and Graesser, A.C. 2012. Automated Discovery of Speech Act Categories in Educational Games. In *Proceedings of EDM*, 25–32.

[32] Searle, J.R. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

[33] Serafin, R. and Di Eugenio, B. 2004. FLSA: Extending Latent Semantic Analysis With Features For Dialogue Act Classification. In *Proceedings of the Association for Computational Linguistics*, 692–699.

[34] Sridhar, V.K.R., Bangalore, S. and Narayanan, S.S. 2009. Combining Lexical, Syntactic and Prosodic Cues For Improved Online Dialog Act Tagging. *Computer Speech & Language*. 23, 4, 407–422.

[35] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. Van and Meteer, M. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*. 26, 3, 339–373.

[36] VanLehn, K., Jordan, P.W., Rosé, C.P., Bhembe, D., Bottner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S. and Srivastava, R. 2002. The Architecture Of Why2-Atlas: A Coach For Qualitative Physics Essay Writing. In *Proceedings of ITS*, 158–167.

Mining the Web to Leverage Collective Intelligence and Learn Student Preferences

Antonio Moretti†, José P. González-Brenes*, Katherine McKnight†
†Center for Educator Learning & Effectiveness
*Center for Digital Data, Analytics & Adaptive Learning
Research & Innovation Network, Pearson
{antonio.moretti, jose.gonzalez-brenes, kathy.mcknight}@pearson.com

ABSTRACT

University professors of conventional offline classes are often experts in their research fields, but have little training on educational sciences. Current educational data mining techniques offer little support to them. In this paper we propose a novel algorithm, Analyzing CurrIculum Decisions (ACID), that leverages collective intelligence to model student opinions to help instructors of traditional classes. ACID mines publicly available educational websites, such as student ratings of professors and course information, and learns student opinions within a statistical framework. We demonstrate ACID to discover patterns in learner feedback and factors that affect Computer Science instruction. Specifically, we investigate the choice of a programming language for introductory courses, the grading criteria and the posting of a publicly available online syllabus.

Keywords

offline teacher support, collective intelligence, web mining

1. INTRODUCTION

There are thousands of undergraduates in computer science programs throughout the US, roughly 24% of whom will switch majors to non-computing fields [7]. An essential component of retaining students is the quality of instruction that students receive in introductory courses [7]. While clear instruction and good pedagogy are widely acknowledged as fundamental to retention, supports for instructors to improve their educational practice are often based on old data; the languages used in computer science courses quickly evolve and old surveys are not useful. In this paper, we develop a data mining technique that will help provide insight into learner feedback which can be translated into changes that affect course quality. In general, our approach is similar to large scale surveys that attempt to be representative of student populations. The benefits of our approach are that it is rapid and inexpensive due to its use of publicly available information on the Web.

The field of educational data mining has been cultivating a strong interest in creating technologies to mine data collected from sophisticated online systems such as intelligent tutoring systems, virtual learning environments, and recently from Massive Open Online Courses (MOOC). The merits of these complex online systems have been demonstrated empirically [2, 8] with controlled studies. MOOCs are a powerful resource that allow educators to study student behavior and social learning in a controlled environment, however the scope of the impact of such technologies is limited. For example, a recent survey of active MOOC users in 200 countries and territories revealed that an overwhelmingly majority of students on these courses correspond to the most educated elite of their respective countries [3]. It is clear that improving basic education worldwide is necessary before MOOCs can deliver their promise. Moreover, because most education still happens offline, it is important to provide educational technologies that can utilize the power of internet to understand student behavior and to deliver these technologies to traditional offline classes. It is not clear how existing educational data mining technologies can help bridge this divide.

We discuss the *Analyzing CurrIculum Decisions*(ACID) [11] methodology, which has been presented and applied briefly. In this paper we elaborate on both our methodology and statistical model and expand upon our results. ACID is an algorithm that leverages collective intelligence within a statistical framework. ACID supports the decisions of instructors of traditional offline courses by extracting from the web teaching syllabi data, and using crowd-sourcing to pair it up with students' course ratings, comments and sentiment to analyze the relationship between the two.

This paper reports a case study of using the ACID methodology to explore three questions that instructors of computer science courses face when designing their courses. In addition we discuss ACID's heuristic value within a larger educational framework. We address the following questions:

1. **What course activities and grading rubric correlate with clear instruction?** The question of how to design a grading rubric and weight course activities determines what students focus on within a course. It is important for instructors to optimize course activities and grading criteria with respect to the student experience.

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Algorithm 1 ACID pseucode

n universities to analyze, z reviews to analyze

procedure ACID

  while $|R| < z$ **do**

    $s \leftarrow$ sample of n universities

    $s \leftarrow$ Remove non-English speaking universities

    $R \leftarrow$ Search_The_Web_For_Reviews(s)

    $R \leftarrow$ ratings rated by more than ϵ students

  $Q \leftarrow$ CrowdSource_Questionnaire(R)

  Analyze_Data(Q)

2. **For introductory classes, which programming language(s) correlate with clear instruction?** Academics and industry professionals disagree as to the programming language that is best suited for beginners [16]. For example, some argue that introductory courses should use interpreted languages that allow for a faster understanding of the applications of programming rather than compiled languages that rely heavily on language-specific syntax. Others believe that developing skill with compiled languages is necessary for future work in computer science. The choice of a first programming language likely affects students' decision to continue education within the field of computer science.

3. **Are students more interested in courses with publicly available online syllabi?** The choice to make a syllabus publicly available adds to information available to prospective students on the Web. We hypothesize that the posting of an online syllabus can be used as a proxy for factors including instructor organization and motivation, and that students will both be more interested in and prefer these courses.

The rest of this paper is organized as follows. § 2 explains the ACID methodology; § 3 describes three case studies of evaluating teaching decisions using ACID; § 4 relates to prior work; § 5 concludes.

2. ANALYZING CURRICULUM DECISIONS

Pseudocode for the ACID methodology is presented in Algorithm 1. For a given number of reviews, we sample n universities, remove the non-English speaking universities, scrape and parse the relevant reviews from a ratings website and retain ratings rated by more than a given number of students. We then extract information from these courses using crowd-sourcing, and analyze the data. We describe the process in detail below.

To evaluate the relative impact of different course features, we mine the web for data that reflect:

- **Curriculum decisions** University professors often upload information about their classes. This information is targeted towards prospective or enrolled students. This information includes syllabi with detailed descriptions of course material such as textbooks, projects, home-works and exams. We make use of this data to infer teaching strategies.

- **Student perceptions of the course.** We make use of self-selected student evaluations collected from a third-party website. The validity and usefulness of self-selected online rating systems, have been assessed in the literature [1, 12]. For example, evidence suggests that online ratings do not lead to substantially more biased ratings than those done in a traditional classroom setting [1] and that online ratings are a proxy to measure student learning [12]: student learning can often be modeled as a latent variable that causes patterns of observed faculty ratings. Researchers hypothesize a non-linear or concave relationship between student learning and the perceived difficulty level of a course [12]; students learn most when a course is not too difficult or too easy. Our work relies on self-selected ratings as a metric to study learner opinion.

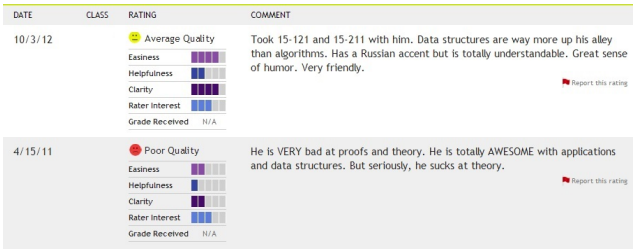


Figure 1: Two Examples from the Ratings Sample

Table 1: Statistics for the Ratings Sample

	Easiness	Helpfulness	Clarity	Interest
Mean	2.84	3.30	3.24	3.35
Std. Dev.	1.33	1.62	1.59	4.00
Median	3.00	4.00	4.00	1.38

We use publicly available self-selected ratings of professors from a third-party website, *Rate My Professor*[1] (RMP). This site allows students to rate the professors of the courses they have taken. The database contains data from over 13 million ratings for 1.5 million professors. They collect ratings on a 1—5 scale (being 1 the lowest possible score, and 5 the highest) under the categories of "easiness", "helpfulness" and "clarity." Additionally students may fill out an "interest" field in which they indicate how appealing the class was before enrolling, and a 350 character summary of their class experience. We focus on perceived clarity because of the direct link between clarity and quality of instruction.

For the purposes of this paper, we focus on Computer Science courses due to our familiarity with the content. Since we do not have access to the ratings database, we develop a process to sample data from the website. For this, we first select a random sample of 50 international universities that teach Computer Science from the Academic Ranking of

[1] `ratemyprofessor.com`

World Universities[2] [14]. From this sample we only consider the 41 universities are English speaking.

We find, scrape and parse the reviews of the ratings data-set for all professors within the computer science departments of the universities in our sample. We remove the ratings from faculty that were rated by fewer than 30 students. More than one professor can teach the same course. For our analysis, we describe one course listing taught by two different professors as two separate courses. Table 1 shows the mean, standard deviation and median of the ratings in our sample. Figure 1 shows two sample ratings for one professor from our sample. The professor name and course names are removed for privacy.

We use Amazon Mechanical Turk, a crowdsourcing platform, to find course features for each of the courses in our ratings sample. We do this by asking respondents to fill out a survey. The survey requests to provide the URL for the online syllabus that corresponds to the course and professor from which we have ratings that is closest to the date of the student review online. Then, using the syllabus, respondents are asked to to provide the programming language(s) used, the textbook(s) used, and the percentage of the grade that was determined by homework, projects, quizzes, exams and whether the course was taught online or in a blended format (both face-to-face and online). However, when we reviewed the responses to the blended format question, it appeared that most syllabi did not provide enough information by which to make an accurate response.

From our original sample of 1,112 courses taught by a unique professor, respondents find an online syllabus matching the professor for 342 courses (∼31%). We hypothesize three explanations for the missing syllabi: (i) the syllabi may be accessed only with a password through a course management system, such as blackboard, (ii) the syllabi may not be available only, or (iii) the respondents are not able to find the syllabi.

3. DATA ANALYSIS: WHAT MAKES A BETTER CLASS?

We report our results of applying the ACID methodology to evaluate teaching decisions. In § 3.1 we assess the quality of the data collected by the crowd sourcing platform. In § 3.2 we discuss the statistical model we use. In § 3.3 we report the results of using ACID.

3.1 Data Quality

We now report the how we attempt to collect high-quality data through the use of crowd-sourcing and how we assess the quality of our data.

Mechanical Turk provides a "master" qualification level to respondents that are more reliable. Masters-level respondents require higher compensation for crowd-sourcing tasks than non-masters level respondents although their "acceptance rate," or proportion of approved tasks is much higher. We ran a preliminary experiment, to decide whether respondents on master level qualification provide better quality data for our purposes. We ask respondents to find the syllabus corresponding to a random sample of 30 courses and to answer a set of questions. Table 2 shows the accuracy and interrater agreement of Masters and non-Masters level respondents.

Table 2: Respondent Validation

	Accuracy	Interrater Agreement
Masters	100%	96.67%
non-Masters	85.56%	6.07%

In the pretest we used a screening question to evaluate the accuracy of respondents' data on each task. We asked respondents to find the URL of the website of a randomly selected faculty member at Carnegie Mellon University from a set of 8, from which we knew the answer. We compared the URL they provided with the correct URL to assess accuracy. Of the 13 responses of non-masters workers that did not provide an exact URL match, five responses left the validation question blank. We found that respondents with master level qualification were significantly more accurate (i.e. answered the validation item correctly) than the non-Masters level respondents (p-value = 0.0002).

Additionally, we tested interrater agreement by asking 3 respondents to carry out the same task, i.e. finding the same URL (for a total of 3x30 or 90 tasks). We used a dummy variable to code whether the three respondents provided the same URL for the course syllabus. Our measure of agreement is calculated by taking the proportion of total responses in which all three respondents provide the same URL. Masters-level respondents agreed (i.e. all three provided the same URL) 100% of the time, whereas the non-Masters level respondents performed much worse – only 6% agreed. As a result of these comparisons, we decided to hire only Masters-level respondents to complete the crowdsourcing experiment.

After collecting the data using Masters level respondents, we performed a post-hoc analysis by examining the responses to the screening question. From the final group of 342 responses that provided a link to an online syllabus, 325 responses (95.03%) provided the correct URL for the faculty website. It should be noted that 13 of the 17 responses that did not provide an exact URL match provided the website for a different faculty member from the set of 8, suggesting that they copied and pasted their previous response without checking to see that the prompt had changed for the new response. Two of the 17 responses provided a link to the directory website for the faculty member rather than the faculty member's personal website. One response provided the correct faculty member's website within the department of Statistics rather than the department of Computer Science (the faculty member is in both departments).

3.2 Model

We describe our general linear mixed model. We provide descriptive statistics and model selection criteria.

[2]Academic Ranking of World Universities is also known as Shanghai Ranking `shanghairanking.com`

Table 3: VPC and ICC Statistics

	University	Professor	Course
VPC	0.0646	0.3365	0.2355
ICC	0.0728	0.3425	0.1982

We explore the relationship between student reviews and features collected from online syllabus data using general linear mixed modeling. Student reviews are organized at three levels: by university, professor and course. It is important to note the non-independence of the student reviews due to the hierarchical or clustered nature of the data. We suspect that student ratings within each course, professor and perhaps university are correlated. We begin by estimating the amount of variance attributed to each of these three levels. The simplest multilevel model does not yet include explanatory variables:

$$y_{i,j} = \beta_0 + u_{0,j} + \epsilon_{i,j} \tag{1}$$

The dependent variable $y_{i,j}$ is the clarity rating that student i gave to level j. The term β_0 represents the intercept or mean student clarity rating across all observations. The term $u_{0,j}$ represents the mean clarity rating for level j. The term $\epsilon_{i,j}$ represents the error attributed to student rating i at level j. For comparison we fit a null or single-level model:

$$y_{i,j} = \beta_0 + \epsilon_{i,j} \tag{2}$$

We calculate the percentage of variation in the data set that is separately attributed to each of the three levels of the data. Conventionally the variance partition coefficient (VPC) and intraclass correlation coefficient (ICC) can be interpreted similarly to an R-squared term and are reported in Table 3.

$$\rho = 1 - \frac{\sigma_e^2}{\sigma_e^2 + \sigma_u^2} \tag{3}$$

The VPC and ICC are denoted by ρ, the residual variance is denoted by σ_e^2 and the variance of the effect is denoted by σ_u^2. The ICC is a statistic that is similar to the VPC. However, since the parameter values of the within and between level variance are estimated using sample data, there may be bias due to sampling variation, particularly when there are fewer observations within a given level. The ICC as described by Bartko [1] corrects for this bias by making a small computational adjustment.[3] Observe that the ICC term appears to give slightly less weight to the course effect. It is clear from both statistics that the main effect is the professor effect.

We examine the professor level-residuals and their associated standard errors to look for variation in clarity ratings across professors. The caterpillar plot displays the professor residuals in rank order together with 95% confidence intervals. Wider intervals occur for professors with more student reviews. Observe that the majority of the intervals do not overlap and thus there are significant differences between professors. The blue circles on the far left represent professors who are rated two standard deviations below the mean clarity rating, whereas those on the far right are 1.5 standard deviations higher than the mean clarity rating. The red horizontal line refers to the "average" professor.

[3]For a description of the computation of the ICC, see the documentation and source code for the R library *lme.*

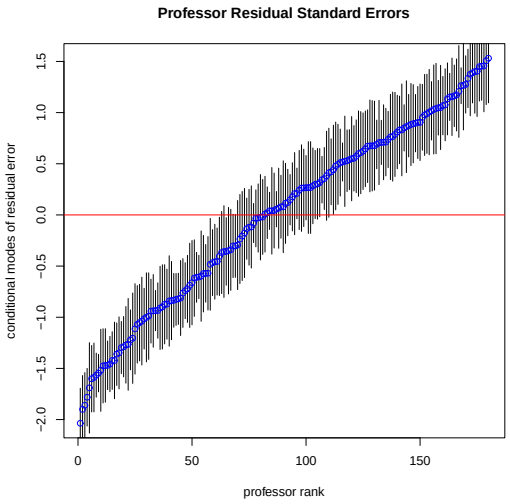


Figure 2: 95% CI for Professor Residual Error

We calculate a Chi-squared likelihood ratio statistic by taking the difference between log likelihood values of two successive models. We begin by comparing the null model and the course level model to compare the significance of including the course effect. We continue by adding each of the additional effects. We do not report the values of the test statistic although all additional levels of complexity are statistically significant. We consider the Bayesian information criterion (BIC) and Akaike information criterion (AIC) as model selection tools to avoid over-fitting the data. The BIC and AIC penalize the log-likelihood of a model for the inclusion of extra parameters. The parameters are estimated using restricted maximum likelihood estimation (REML).

We choose the model with the minimum BIC. A two-level mixed model including course effect and professor effect provides the optimal Bayesian information criterion value. Two and three way interaction effects were considered although they did not decrease the AIC or BIC of any of the models. While the log likelihood value is maximized by including the university effect, a simpler model is preferable because it involves fewer parameter estimates and is more likely to generalize. The model can be written in matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\nu} + \boldsymbol{\epsilon} \tag{4}$$

$\mathbf{Y}$ denotes the response variable observations (student ratings). The matrix $\boldsymbol{\beta}$ represents a vector of fixed-effects parameters with a design matrix $\mathbf{X}$. $\mathbf{Z}$ is a design matrix of indicator variables denoting group membership across random-effect levels and $\boldsymbol{\nu}$ is a vector containing random-effect parameters. $\boldsymbol{\epsilon}$ is a vector of error terms.

3.3 Case Studies

We show the results of using the ACID methodology to answer three course design questions.

Table 4: Programming Language Statistics

	Value	Std.Err	t-value	Pr<|t|	n
C	3.38	0.32	10.58	0.0000	109
C++	3.30	0.31	10.65	0.0000	214
Java	3.62	0.19	19.33	0.0000	353
Python	3.70	0.26	14.50	0.0000	133
Scheme	4.06	0.47	8.61	0.0000	32
Scratch	3.91	0.84	4.67	0.0000	49

3.3.1 *For introductory classes, which programming language do students associate with clear instruction?*

Professors teaching introductory level courses in computer science choose between a number of programming languages and textbooks. We make use of the data collected to provide insights into which programming languages beginning students associate with clear instruction. We filter the data to only include introductory level courses (one which does not require any prerequisite coursework in computer science). Our restricted sample includes 1,024 reviews; 34.58% of all reviews with syllabus data are of introductory courses. We explore the relationship between clarity ratings and programming language with random professor and course effects. Programming languages with less than 30 student reviews are not reported[4]. Table 4 gives the estimates for student ratings of clarity by programming language and their associated p-values. An intercept is not modeled in order to make the results easily interpretable. The mean clarity rating for introductory courses is 3.599.

We found C and C++ had the lowest coefficients (i.e. compiled languages had the lowest perceived clarity ratings). Scheme and Scratch have the highest clarity ratings followed by Python and Java. We note that the standard errors are largest for Scheme and Scratch and smallest for Java and Python. This suggests that results for Java and Python are stronger. Students in our sample associate clearer instruction with interpreted languages rather than compiled languages. Also, both Python and Java are associated with clearer instruction than C or C++.

3.3.2 *What mix of course activities – exams, quizzes, homework and projects – do students associate with clear instruction?*

To assess students' course ratings of clarity based on the percentage of the grade due to exams, quizzes, homework and projects, we created a factor made up of four clusters representing four ways of weighting homework, projects, exams, quizzes and miscellaneous (such as extra credit) for the students' grade. We begin by sorting the data to only include observations in which the grading criteria (percentage of the grade determined by homework, projects, exams, quizzes and miscellaneous) is available and sums to 100. Of the 2,935 observations with syllabus data, there are 2,225 observations with full grading criteria. The difference in these numbers represents 710 ratings for which the respondents were not able to find a complete grade breakdown from the online syllabus.

[4]SQL is a special purpose programming language used only for relational databases and is not reported.

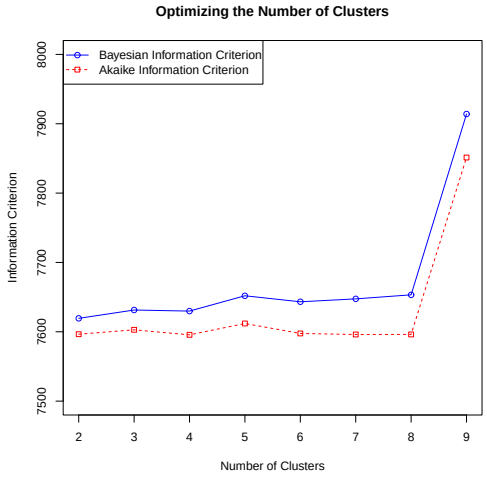


Figure 3: Information Criterion

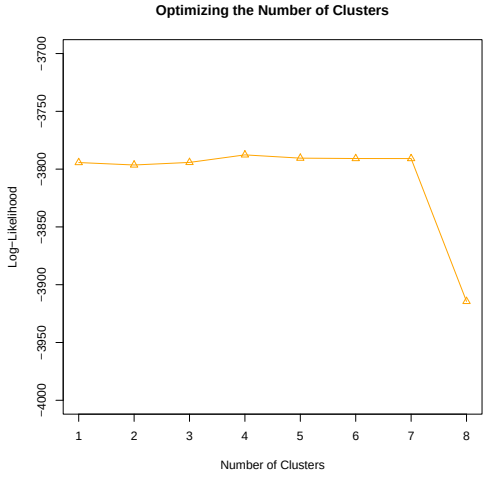


Figure 4: Log Likelihood

Table 5: Cluster Statistics

	HW	Projects	Exams	Quizzes	Other
Cluster1	18.11	2.36	76.66	0.61	2.25
Cluster2	20.59	7.90	48.90	12.46	10.15
Cluster3	7.00	40.18	46.23	3.51	3.08
Cluster4	42.93	0.76	54.61	0.70	2.00

Table 6: Grading Criteria Statistics

	Clarity	Std.Err	t-value	Pr<|t|	n
Exam Heavy	3.23	0.12	26.91	0	726
Equal Mix	3.52	0.14	26.04	0	484
Exam Proj	3.65	0.13	27.76	0	610
Exam HW	3.12	0.13	23.53	0	415

We use k-means clustering to partition the 2,225 observations with complete grading criteria information based on the five aforementioned variables. We optimize k, our number of clusters, by examining how the BIC and AIC of the mixture model change based on the number of clusters selected. Figure 3 displays the information criterion and Figure 4 displays the log-likelihood values for each number of clusters respectively. A solution involving two clusters minimizes the BIC of the model, whereas a four cluster solution minimizes the AIC. The log likelihood is optimized with the four cluster solution. We consider both two and four cluster models as optimal and we find that they lend themselves to similar interpretation. The cluster means for the four cluster solution are presented in table 5.

The first cluster represents courses that are heavily weighted towards exams with a smaller weight towards homework. The second cluster represents a more even weighting of exams, homework, projects and quizzes. The third cluster represents an equal weighting towards exams and projects. The fourth cluster represents courses that are heavily weighted towards exams and homework. The cluster membership is treated as a predictor variable and modeled using equation 4. Table 6 displays the estimated clarity ratings within each group for the four cluster solution.

The exams and projects cluster has the highest estimate of clarity. We find that weighting projects equally with exams is associated with a clearer course experience. The equal mix cluster also is associated with higher clarity estimates. The exam heavy cluster and the exam and homework heavy clusters are associated with lower student clarity ratings. We find that a rubric that weights exams and projects evenly has higher perceived clarity ratings to a rubric which is weighted heavily towards exams and homework. This result extends to both two and four cluster solutions.

3.3.3 *Does the posting of a syllabus online translate into higher ratings?*

We hypothesize the posting of the syllabus online is a proxy for organization, perhaps motivation or drive of the professor. We make use of all of the data collected to compare student reviews of professors who have a publicly available syllabus and of those who do not. Many professors may choose to only post a syllabus through course management systems that require a password. Potential students of these courses are unable to access the syllabus to determine whether the course would be a good fit. We treat the posting of an online syllabus as a factor and test for differences in clarity ratings between the two groups using our model.

We find statistically significant differences between clarity, helpfulness and interest ratings and report the clarity estimates for the two groups in Table 7. We note that the difference in easiness ratings is not statistically significant. We find evidence that students are more interested in professors and courses in which the syllabus is made publicly available. We note that the parameter estimates for the two groups are within one standard error of one another which suggests that the conclusions are modest.

Table 7: Online Syllabi

	Clarity	Std. Err	t-value	Pr<|t|	n
Available	3.33	0.07	44.48	0	2953
Not Found	3.26	0.07	46.03	0	7702

4. RELATION TO PRIOR WORK

Research has recently focused on online faculty ratings with mixed conclusions. Felton et al. [4] found that online instructor ratings were associated with perceived easiness, and that a "halo effect" existed in which raters gave high scores to instructors perhaps because their courses were easier. We find that student ratings of clarity and easiness are correlated (ρ=0.45) although not as strongly associated as clarity and helpfulness. We do find that student ratings of clarity and helpfulness are highly correlated (ρ=0.84). We chose to focus on clarity ratings as we assumed these were less susceptible to a "halo effect" and other bias relative to the overall ratings of a course or professor. Otto et al [13] found issues related to bias in online ratings stating that online ratings are characterized by selection bias as anyone can enter faculty ratings at any time. Carini et al [1], Hardy [5], McGhee and Lowell [6] had contradictory results finding that an online format did not lead to more biased ratings. Otto et al. [12] hypothesized that instructor clarity and helpfulness as captured by Rate My Professor are more positively associated with student learning than easiness.

Several approaches have been proposed to synthesize responses using crowd sourcing systems such as Amazon's Mechanical Turk. Majority voting is perhaps the simplest way to combine crowd responses using equal weights irrespective of respondent experience. The results of our preliminary analysis in accessing the accuracy of non-Masters level respondents correspond to the steep drop in respondent accuracy noted by Karger [9] when low-quality respondents are present. Whitehill et al [15] proposed a probabilistic model for combining crowd responses called Generative model of Labels, Abilities and Difficulties (GLAD). The GLAD methodology makes use of the EM algorithm to calculate parameter estimates of unobserved variables including an approximation of the expertise of the rater. Khattak and Salleb-Aouissi compared the accuracy and percentage of bad responses using majority voting, probabilistic models, and their novel approach entitled Expert Label Injected Crowd Estimation (ELICE) [10]. ELICE makes use of a few "ground truth" responses and incorporates expertise of the labeler, difficulty of the instance and an aggregation of labels. Khattak and Salleb-Aouissi found that their approach was robust and outperformed GLAD and iterative methods even when bad labelers were present. Our simple approach was to use Masters level respondents from Mechanical Turk although GLAD and ELICE are alternative methods to reduce the number of expert level respondents required while also obtaining high quality data.

5. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

We demonstrate how the Analyzing CurrIculum Decisions (ACID) methodology can be used to leverage collective intelligence and learn student preferences. In introductory

computer science courses, we find that students that are taught interpreted languages find their classes clearer. We also that find students who are given an even weighting of exams and projects find their classes clearer; and that interest in a course corresponds to the availability of an online syllabus. Our study does not necessarily suggest that teachers should change their programming language. Further research is needed before drawing causal inferences. We argue that ACID is a beneficial tool to discover patterns in student behavior. Syllabus data and course ratings data are becoming increasingly available on the Web. This data is used by millions of students and worthy of further research.

This study can be expanded in several ways. Student evaluations often include free form text where students can describe their experience in the course. Sentiment analysis is a probabilistic approach for categorizing student comments as being either positive or negative. One extension is to regress text sentiment on course features. There is arguably a strong association between comment sentiment and student preference. Another way ACID can be applied is to disciplines other than computer science, or to discover patterns in syllabi across disciplines that can provide insight into learner experiences.

6. REFERENCES

[1] R. Carini, J. Hayek, G. Kuh, J. Kennedy, and J. Ouimet. College student responses to web and paper surveys: does mode matter? *Research in Higher Education*, 44(1):1–19, 2003.

[2] A. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva, editors, *User Modeling 2001*, volume 2109 of *Lecture Notes in Computer Science*, pages 137–147. Springer Berlin Heidelberg, 2001.

[3] E. J. Emanuel. Online education: Moocs taken by educated few. *Nature*, 503(7476):342–342, 2013.

[4] J. Felton and J. Mitchell. Web based student evaluations of professors: the relations between perceived quality, easiness and sexiness. *Assessment and Evaluation in Higher Education*, 29(1):91–108, 2004.

[5] N. Hardy. Online ratings: fact and fiction. *New Directions for Teaching and Learning*, (96):31–38, 2003.

[6] N. Hardy. Psychometric properties of student ratings of instruction in online and on-campus courses. *New Directions for Teaching and Learning*, 2003(96):39–48, 2003.

[7] M. Haungs, C. Clark, J. Clements, and D. Janzen. Improving first-year success and retention through internet-based cs0 courses. *ACM SIGCSE*, pages 549–594, 2012.

[8] S. Jaggars and T. Bailey. Effectiveness of fully online courses for college students: Response to a department of education meta-analysis. *Teachers College: Community College Research Center*, 2010.

[9] S. Karger, D. Oh and D. Shah. Budget–optimal task allocation for reliable crowdsourcing systems. *CoRR*, arXiv:1110.3564, 2011.

[10] F. Khattak and A. Salleb-Aouissi. Robust crowd labeling using little experience. *Discovery Science*, 8140:94–109, 2013.

[11] A. Moretti, J. Gonzalez-Brenes, and K. McKnight. Towards data–driven curriculum design: Mining the web to make better teaching decisions. *EDM*, 2014.

[12] J. Otto, D. A. Sanford Jr, and D. N. Ross. Does ratemyprofessor. com really rate my professor? *Assessment & Evaluation in Higher Education*, 33(4):355–368, 2008.

[13] J. Otto, D. A. Sanford Jr, and W. Wagner. Analysis of online student ratings of university faculty. *Journal of College Teaching & Learning*, 2(7):25–30, 2005.

[14] Shanghai. Academic ranking of world universities. *Retrieved from http://www.shanghairanking.com/*, Accessed at 2013 12 01.

[15] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Neural Information Processing Systems*, pages 2035–2043, 2009.

[16] J. Zelle. Python as a first language. *Retrieved from http://mcsp.wartburg.edu/zelle/python/python-first.html/*, Accessed at 2014 02 23.

APPENDIX

A. SAMPLE OF UNIVERSITIES SELECTED

	Country	n Professors	n Courses	n Reviews
Colorado State	USA	1	9	32
Carnegie Mellon University	USA	3	21	102
North Carolina State	USA	2	10	63
Pennsylvania State	USA	12	74	938
Rensselaer Polytechnic Institute	USA	3	22	131
Rutgers	USA	8	30	468
Simon Fraser	Canada	27	98	1873
SUNY Stony Brook	USA	8	55	505
UC Davis	USA	10	44	589
UNC Chapel Hill	USA	1	4	49
University of Alberta	Canada	2	6	69
University of Arizona	USA	3	13	158
University of Delaware	USA	15	56	806
University of Florida Gainsville	USA	5	36	321
University of Illinois at Urbana	USA	5	14	339
University of Massachusetts	USA	6	39	405
University of Montreal	USA	1	6	59
University of Toronto	Canada	14	66	775
University of Utah	USA	2	17	66
University of Virginia	USA	3	19	131
University of Waterloo	Canada	46	125	2700
Vanderbilt University	USA	2	10	76

Non-cognitive factors of learning as predictors of academic performance in tertiary education

Geraldine Gray, Colm McGuinness, Philip Owende
Institute of Technology Blanchardstown
Blanchardstown Road North
Dublin 15, Ireland
geraldine.gray@itb.ie

ABSTRACT

This paper reports on an application of classification and regression models to identify college students at risk of failing in first year of study. Data was gathered from three student cohorts in the academic years 2010 through 2012 (n=1207). Students were sampled from fourteen academic courses in five disciplines, and were diverse in their academic backgrounds and abilities. Metrics used included non-cognitive psychometric indicators that can be assessed in the early stages after enrolment, specifically factors of personality, motivation, self regulation and approaches to learning. Models were trained on students from the 2010 and 2011 cohorts, and tested on students from the 2012 cohort. Is was found that classification models identifying students at risk of failing had good predictive accuracy ($> 79\%$) on courses that had a significant proportion of high risk students (over 30%).

Keywords

Educational data mining, learning analytics, academic performance, non cognitive factors of learning, personality, motivation, learning style, learning approach, self-regulation

1. INTRODUCTION AND LITERATURE REVIEW

Learning is a latent variable, typically measured as academic performance in continuous assessment and end of term examinations [33]. Identifying predictors of academic performance has been the focus of research for many years [20, 34], and continues as an active research topic [6, 8], indicating the inherent difficulty in generating models of learning [29, 46]. More recently, the application of data mining to educational settings is emerging as an evolving and growing research discipline [40, 43]. Educational Data Mining (EDM) aims to better understand students and how they learn through the use of data analytics on educational data [42, 10]. Much of the published work to date is based on ever-increasing volumes of data systematically gathered by education providers, particularly log data from Virtual Learning Environments and Intelligent tutoring systems [16, 2]. Further work is needed to determine if gathering additional predictors of academic performance can add value to existing models of learning.

Research from educational psychology has identified a range of non-cognitive psychometric factors that are directly or indirectly related to academic performance in tertiary education, particularly factors of personality, motivation, self regulation and approaches to learning [8, 9, 35, 39, 44, 25]. Personality based studies have focused on the Big-5 personality dimensions of conscientiousness, openness, extroversion, stability and agreeableness [9, 22, 27]. There is broad agreement that conscientiousness is the best personality based predictor of academic performance [44]. For example, Chamorro et al. [9] reported a correlation of r=0.37 ($p<0.01$, n=158) between conscientiousness and academic performance. Correlations between academic performance and openness to new ideas, feelings and imagination are weaker. Chamorro et al. [9] reported a correlation of r=0.21 ($p<0.01$, n=158) but lower correlations were reported in other studies (see Table 1) which may be explained by variations in assessment type. Open personalities tend to do better when assessment methods are unconstrained by submission rules and deadlines [27]. Studies are inconclusive on the predictive validity of other personality factors [44].

A meta-analysis of 109 studies analysing psychosocial and study skill factors found two factors of motivation, namely self-efficacy (90% CI [0.444,0.548]) and achievement motivation (90% CI [0.353, 0.424]), had the highest correlations with academic performance [39]. Distinguishing between learning (intrinsic) achievement and performance (extrinsic) achievement goals, Eppler and Harju [19] found learning goals (r=0.3, $p<0.001$, n=212) were more strongly correlated with academic performance than performance goals (r=0.13, $p> 0.05$, n=212). Covington [13] however argues that setting goals in itself is not enough, as ability to self-regulate learning can be the difference between achieving, or not achieving, goals set. Self-regulated learning is recognised as a complex concept to define as it overlaps with a number of other concepts including personality, self-efficacy and goal setting [4]. Ning and Downing [35] reported high correlations between self regulation and academic performance, specifically self-testing (r=0.48, $p<0.001$) and monitoring understanding (r= 0.42, $p<0.001$). On the other hand, Komarraju and Nadler [31] found effort management, includ-

Published in CEUR-WS:
NCFPAL workshop (Ritter and Fancsali)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

ing persistence, had higher correlation with academic performance (r=0.39, p<0.01) than other factors of self-regulation and found that self-regulation (monitoring and evaluating learning) did not account for any additional variance in academic performance over and above self-efficacy, but study effort and study time did account for additional variance.

Research into approaches to learning has its foundations in the work of Marton & Säljö [32] who classified learners as shallow or deep. Deep learners aim to understand content, while shallow learners aim to memorise content regardless of their level of understanding. Later studies added strategic learners [18, pg. 19], whose priority is to do well, and will adopt either a shallow or deep learning approach depending on the requisites for academic success. Comparing the influence of approaches to learning on academic performance, Chamorro et al [9] reported a deep learning approach (r=0.33, p<0.01) had higher correlations with academic performance than a strategic learning approach (r=0.18, p<0.05). Cassidy [8] on the other hand found correlations with a deep learning approach (r=0.31, p<0.01) were marginally lower than with a strategic learning approach (r=0.32, p<0.01). Differences found have been explained, in part, by assessment type [49], highlighting the importance of assessment design in encouraging appropriate learning strategies.

Knight, Buckingham Shum and Littleton argued learning measurement should go beyond measures of academic performance [29], promoting greater focus on learning environment and encouragement of malleable, effective learning dispositions. Disposition relates to a tendency to behave in a certain way [6]. An effective learning disposition describes attributes and behaviour characteristic of a good learner [6]. A range of non-cognitive psychometric factors have been associated with an effective learning disposition such as a deep learning approach, ability to self-regulate, setting learning goals, persistence, conscientiousness and subfactors of openness, namely intellectual curiosity, creativity and open-mindednesss [6, 29, 47]. A lack of correlation between such non-cognitive factors and academic performance is in itself insightful, suggesting assessment design that fails to reward important learning dispositions. It has been argued that effective learning dispositions are as important as discipline specific knowledge [6, 29].

Statistical models have dominated data analysis in educational psychology [15], particularly correlation and regression [25]. Relatively high levels of accuracy were reported in regression models of academic performance that included cognitive and non-cognitive factors. For example, Chamorro-Premuzic et al [9] reported a coefficient of determination (R^2) of 0.4 when predicting 2^{nd} year GPA (based on essay type examinations) in a regression model that included prior academic ability, personality factors and a deep learning approach. Robbins [39] reported similar results (R^2=0.34) in a meta-analysis of models of cognitive ability, motivation factors and socio-economic status. Models of non-standard students were less accurate, for example Swanberg & Martinsen [44] reported R^2=0.21 in models of older students (age: m=24.8) based on prior academic performance, personality, learning strategy, age and gender. Lower accuracies were also reported in studies not including cognitive ability. Robbins [39] reported R^2=0.27 in a meta-analysis of models of factors of motivation. Komarraju et al. [30] predicted GPA (R^2=0.15) from variables of personality and learning approach, while Bidjerano & Dai [4] had similar results (R^2=0.11) with factors of personality and self-regulation.

Linear regression assumes constant variance and linearity between independent and dependent attributes. There is evidence to suggest variance is not constant for some non-cognitive factors. For example, De Feyter et al. [14] found low levels of self-efficacy had a positive, direct effect on academic performance for neurotic students, and for stable students, average or higher levels of self-efficacy only had a direct effect on academic performance. In addition, Vancouver & Kendall [48] found evidence that high levels of self-efficacy can lead to overconfidence regarding exam preparedness, which in turn can have a negative impact on academic performance. Similarly, Poropat [38] cites evidence of non-linear relationships between factors of personality and academic performance, including conscientiousness and openness. It is therefore pertinent to ask if data mining's empirical modelling approach is more appropriate for models based on non-cognitive factors of learning.

A growing number of educational data mining studies have investigated the role of non-cognitive factors in models of learning [6, 41, 36]. Bergin [3] cited an accuracy of 82% using an ensemble model based on prior academic achievement, self-efficacy and study hours, but due to the small sample size (n=58) could not draw reliable conclusions from the findings. The class label distinguished strong (grade>55%) versus weak (grade<55%) academic performance based on end of term results in a single module. Gray et al. [23] cited similar accuracies (81%, n=350) with a Support Vector Machine model using cognitive and non cognitive attributes to distinguish high risk (GPA<2.0) from low risk (GPA≥2.5) students based on first year GPA. Model accuracy was contingent on modelling younger students (under 21) and older students (over 21) separately.

The focus of this study was to investigate if non-cognitive factors of learning, measured during first year student induction, were predictive of academic performance at the end of first year of study. We evaluated both regression models of GPA and classification models that predicted first year students at risk of failing. Participants were from a diverse student population that included mature students, students with disabilities, and students from disadvantaged socio-economic backgrounds.

2. METHODOLOGY

The following sections report on study participants and the study dataset. Data analysis was conducted following the CRoss Industry Standard for Data Mining (CRISP-DM) using RapidMiner V5.3 and R V3.0.2.

2.1 Description of the study participants

The participants were first year students at the Institute of Technology Blanchardstown (ITB), Ireland. The admission policy at ITB supports the integration of a diverse student population in terms of age, disability and socio-economic background. Each September 2010 to 2012, all full-time, first-year students at ITB were invited to participate in the study by completing an online questionnaire administered

Table 1: Correlations with Academic Performance in Tertiary Education

Study	N	age	AP	Temperament		Motivation			Learning Approach			Learning Strategy		
				Conscientious	Open	Self Efficacy	Intrinsic Goal	Extrinsic Goal	Deep	Shallow	Strategic	Self Regulation	Study Time	Study Effort
[4]	217	m=22	self reported GPA										0.33**	0.0.23**
[8]	97	m=23.5	GPA			0.397***			0.398**	-0.013	0.316**			
[9]	158	18-21	GPA	0.37**	0.21**				0.398*	-0.15	0.18*			
[17]	146	17-52	GPA	0.21	0.06				0.097	-0.054	0.153			
[19]	212	m=19.2	GPA				0.3***	0.13						
[27]	133	18-22	GPA	0.46**	-0.08									
[30]	308	18-24	self reported GPA	0.29**	0.13*									
[31]	257	m=20.5	GPA			0.3**						0.14*	0.31**	0.39**
[35]	581	20.48	GPA											0.0.24**
[39]	meta	analysis, 18+	GPA			0.496		0.179						
[44]	687	m=24.5	single exam						0.16	-0.25				

$^{*}p < .05$, $^{**}p < .01$, $^{***}p < 0.001$

during first year student induction. A total of 1,376 (52%) full-time, first year students completed the online questionnaire. Eliminating students who did not give permission to be included in the study (35) and invalid data (134) resulted in 45% of first year full time students participating in the study (n=1207).

Participants ranged in age from 18 to 60, with an average age of 23.27; of which, 355 (29%) were mature students (over 23), 713 (59%) were male and 494 (41%) were female. There were 32 (3%) participants registered with a disability. Students were enrolled on fourteen courses across five academic disciplines, Business (n=402, 33%), Humanities (n=353, 29%), Computing (n=239, 20%), Engineering (n=172, 14%) and Horticulture (n=41, 3%).

Academic performance was measured as GPA, an aggregate score of between 10 and 12 first year modules, range 0 to 4, and was calculated on first exam sitting only. The GPA distribution (profiled sample) was compared with the GPA distribution of the full cohort of students for that year (reference sample) using a Kolmogorov-Smirnov non-parametric test. The recorded differences in the distribution for 2010 (D=0.032, p=0.93), 2011 (D=0.036, p=0.90) and 2012 (D=0.042, p=0.69) were not statistically significant. The distribution of GPA was also similar across the three years of study. The largest difference was between the 2010 and 2012 profiled samples (D=0.063, p=0.37) and was not significant. To pass overall, a student must achieve a GPA $\geq$ 2.0 and pass each first year module. 89% of students with GPA $>$ 2.5 passed all modules indication a low risk group that can progress to year two. 84% of students with a GPA $<$ 2 failed three or more modules, indicating a high risk group falling well short of progression requirements. Of the students in GPA range [2.0, 2.49], 39% passed all modules, 36% failed one module, 18% failed two modules, and 7 % failed more than two modules. This is a less homogenous group in terms of academic profile, but could be generally regarded as borderline, either progressing on low grades or required to repeat one or two modules in the repeat exam sittings. Figure 1 and Table 2 illustrate GPA distribution by course.

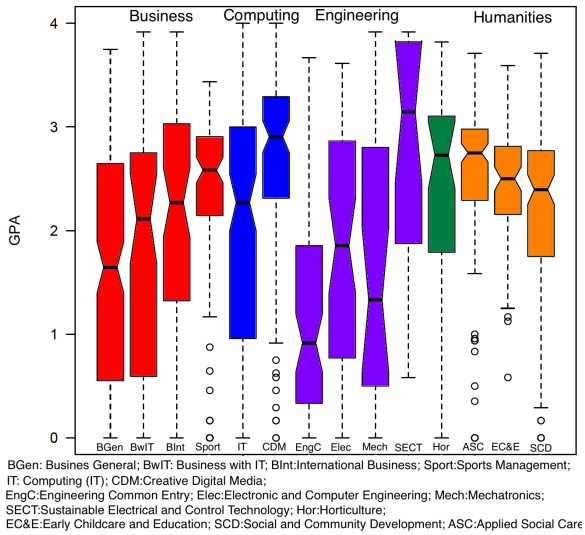


Figure 1: Notched box plots for GPA by course

2.2 The Study Dataset

Table 3 lists the psychometric factors included in the dataset, collected using an online questionnaire developed for the study (www.howilearn.ie). With the exception of learning modality, questions were taken from openly available, validated instruments, with some changes to wording to suit the context. Where two questions were similar on the published instrument, only one was included. This choice was made to reduce the overall size of the questionnaire, despite the likely negative impact on internal reliability statistics. Questionnaire validity and internal reliability were assessed using a paper-based questionnaire that included both the revised wording of questions used on the online questionnaire (reduced scale), and the original questions from the published instruments (original scale). The paper questionnaire was administered during scheduled first year lectures across all academic disciplines. Pearson correlations between scores calculated from the reduced scale, and scores calculated from the original scale, were high for all factors (>=0.9) except intrinsic goal orientation and study time and environment, confirming the validity of the study instrument for those factors. Internal reliability was assessed using Cronbach's alpha. All factors had acceptable reliability (>0.7)[1] given the small number of questions per scale (between 3 and 6), with the exception again of intrinsic goal orientation and study time and environment. Learner modality data (Visual, Auditory, Kinaesthetic (VAK) [21]) was based an instrument developed by the National Leaning Network Assessment Services (NLN) (www.nln.ie).

[1]While generally a Cronbach alpha of $>$ 0.8 indicates good internal consistency, Cronbach alpha closer to 0.7 can be regarded as acceptable for scales with fewer items [12, 45].

Table 2: Academic profile by course

Course Name	n	GPA*	high risk	borderline	low risk
all participants	1207	2.1±1.1	28%	16%	46%
Computing (IT)	137	2.0±1.2	47%	11%	42%
Creative Digital Media	102	2.6±1.0	20%	8%	72%
Engineering common	73	1.1±0.9	79%	8%	13%
Electronic & computer eng.	52	1.8±1.2	52%	10%	38%
Mechatronics	27	1.6±1.2	63%	7%	30%
Sustainable Electrical & Control Technology	20	2.8±1.1	30%	5%	65%
Horticulture	41	2.4±1.1	27%	2%	71%
Business General	183	1.7±1.1	56%	15%	29%
Business with IT	60	1.8±1.2	46%	22%	32%
Business International	64	2.2±1.1	41%	14%	45%
Sports Management	95	2.3±0.9	22%	24%	54%
Applied Social Care	146	2.5±0.7	15%	16%	69%
Early Childcare	80	2.4±0.6	20%	28%	52%
Social & Community Development	127	2.2±0.9	30%	27%	43%

*GPA mean and standard deviation.

Table 3: Study factors, mean and standard deviation

Category & Instrument	Study Factor
Personality: IPIP scales (ipip.ori.org) [22]	Conscientiousness (5.9±1.5) Openness (6.1±1.3)
Motivation: MSLQ [37]	Intrinsic Goal Orientation (7.1±1.4) Self Efficacy (6.9±1.4) Extrinsic Goal Orientation (7.8±1.4)
Learning approach: R-SPQ-2F [5]	Deep Learner (5.4±2.9) Shallow Learner (1.3±1.9) Strategic Learner (3.4±2.5)
Self-regulation: MSLQ [37]	Self Regulation (5.9±1.4) Study Effort (5.9±1.8) Study Time & Environment (6.2±2.3)
Learner modality: NLN profiler	Visual (7.2±2.1) Auditory(3.3±2.2) Kinaesthetic(4.5±2.4)
Other factors:	Preference for group work (6.5±3.4) Age (23.27±7.3) Male=713 (59%), Female=494 (41%)

Note: All ranges are 0 to 10 apart from age.

Prior knowledge of the student available to the college at registration, namely age, gender and prior academic performance, was also available to the study. Access to full time college courses in Ireland is based on academic achievement in the Leaving Certificate, a set of state exams at the end of secondary school. College places are offered based on CAO[2] points, an aggregate score of grades achieved in a student's top six leaving certificate subjects, range 0 to 600. Table 4 summarises participant profile by course.

Table 4: Participant profile based on prior knowledge, means and standard deviation

Course Name	n	CAO points	age	%age male	Z*
Computing (IT)	137	232±67	24±8	91%	9
Creative Digital Media	102	305±79	23±7	68%	7
Engineering common	73	220±61	20±3	92%	8
Electronic & computer eng	52	232±53	22±7	92%	3
Mechatronics	27	238±46	21±3	85%	1
Sustainable Electrical & Control Technology	20	199±97	27±7	95%	0
Horticulture	41	273±66	28±11	8%	4
Business General	183	256±57	21±5	54%	10
Business with IT	60	229±75	22±5	60%	6
Business International	64	248±51	21±5	24%	6
Sports Management	95	306±86	23±6	84%	8
Applied Social Care	146	259±84	28±9	32%	10
Early Childcare	80	308±78	22±5	6%	7
Social & Community Development	127	266±78	25±8	29%	9

*Number of study factors differing significantly from a normal distribution ($p<<0.001$).

3. RESULTS

Correlation and regression were used to analyse relationships between study factors and GPA. Subsequent analysis used classification techniques to identify students at risk of failing. Unless otherwise stated, models are based age, gender and non-cognitive factors of learning as listed in Table 3.

All non-cognitive factors of learning failed the Shapiro−Wilk normality test which is common in data relating to education and psychology [26]. However factors of personality were normally distributed within each discipline except for business. Intrinsic motivation and study effort were also normally distributed for engineering and computing students. There were further improvements when analysing subgroups by academic course. Factors of personality, self regulation and intrinsic motivation were normally distributed for all courses. With the exception of approaches to learning, learner modality, preference for group work and GPA, other factors were normally distributed for most courses. Table 4 illustrates the number of attributes that differed significantly from a normal distribution by course. Larger groups were more likely to fail tests of normality.

3.1 Correlations with Academic Performance

Correlations between study factors and GPA were assessed using Pearson's product-moment correlation coefficient (PPMCC). As some attributes violated the assumption of normal distribution, significance was verified with bootstrapped 95% confidence intervals using the bias corrected and accelerated method [7] on 1999 bootstrap iterations.

Bootstrap correlation coefficients are given in Table 5. With the exception of learning modality, all non-cognitive factors were significantly correlated with GPA. The highest correlations with GPA were found for approaches to learning, specifically deep learning approach (r=0.23, bootstrap 95% CI[0.18, 0.29]), and study effort (r=0.19, bootstrap 95% CI [0.13, 0.24]). Age also had a relatively high correlation with GPA (r=0.25, bootstrap 95% CI [0.19, 0.3]). A shallow learning approach (r=-0.15, bootstrap 95% CI[-0.21, -0.09]) and preference for group work (r=-0.076, bootstrap 95% CI [-0.14, -0.02]) were negatively correlated with GPA. Openness had one of the weakest significant correlations with GPA (r=0.08, bootstrap 95% CI [0.03, 0.14]). Correlations were comparable with other studies that included a diverse student population [4, 9, 28] with the exception of self efficacy (r=0.12, bootstrap 95% CI [0.06, 0.17])) which was lower than expected. This may be reflective of the low entry requirements for some courses.

[2]CAO refers to the Central Applications Office with responsibility for processing applications for undergraduate courses in the Higher Education Institutes in Ireland.

3.2 Regression models

Regression models predicting GPA from non-cognitive variables were run for the full dataset and for subgroups by disciplines and by course. The coefficient of determination (R^2) is reported to facilitate comparison with other studies. However R^2 is influenced by the variability of the underlying independent variables. Consequently Achen [1, pg 58-61] argued that prediction error is a more appropriate fitness measure for psychometric data. Therefore absolute error mean and standard deviation is also reported.

A regression model for all participants ($R^2 = 0.14$) was comparable with other reported models of non-cognitive factors [4, 30]. However when modelling students by discipline and by course, there were significant differences in model performance. A chow test [11] comparing the residual error in a regression model of all participants (full model) with the residual errors of models by discipline (restricted models) showed significant differences between the full and restricted models (F(17,1098)=22.02, p=0). There was also significant differences between models based on a particular discipline (full model) and models of courses within that discipline (restricted models). In computing, significant differences of F(17,205)=2.22 (p=0.005) were found between the full model and the two restricted models. Within engineering, a model combining mechatronics with electronic & computing engineering was not significantly different from a model of those two courses individually (F(17,79)=0.58, p=0.89), but including either common entry students and/or sustainable electrical & control technology resulted in significant differences between the full and restricted models. Sustainable electrical & control technology was therefore excluded from further consideration because of the small sample size (n=20). Significant differences were also found in models of each of the three humanities courses compared with those courses combined (F(17,302)=2.22, p=0.004). The least significant differences were found in models of business students provided sport management was excluded (F(17, 307)=1.95, p=0.015). Adding sports management further increased the difference in model residual errors (F(17,334)=8.36, p=0). Table 6 gives model details by course and factors used in each model. Electronic & computer engineering students and mechatronic students were combined.

In general, models based on technical courses had a higher R^2 than models for non technical courses. For example, engineering courses, computing (IT) and business with IT all had $R^2 > 0.3$. Absolute error for these courses was in the range [0.63,0.8]. The difference between the highest absolute error (m=0.8, s=0.56[3]) and the lowest absolute error (m=0.63, s=0.54) was not significant (t(15)=1.74, p=0.1). Regression results for International Business was also relatively good (R^2=0.27). For the remaining non-technical disciplines R^2 was lower (range [0.12,0.17]) but the absolute error was more varied. Early childcare had the lowest absolute error (m=0.37, s=0.34) while general business had the highest absolute error (m=0.9, s=0.53). The difference was significant (t(15)=10.3, p<0.001) and may be explained by the greater distribution of GPA scores in general business.

There was little agreement across models on which study factors were most predictive of GPA. Approaches to learning and age were significant for models of all participants, computing students and engineering students, but motivation and learning strategy were more significant for Business with IT. Factors of motivation, learning strategy and approaches to learning were also relevant to models in the humanities courses. All regression models improved when prior academic performance was included in the model. The most significant increase was for sports management, R^2 increased from 0.16 to 0.30. Business with IT and applied social care also increased by more than 0.1. For all other regression models, R^2 increased by between 0.05 and 0.09

3.3 Classification models

Classification models were generated using four classification algorithms, namely Naïve Bayes (NB), Decision Tree (DT), Support Vector Machine (SVM), and k-Nearest Neighbour (k-NN). A binary class label was used based on end of year GPA score, range [0-4]. The two classes were: high risk students (GPA<2, n=459); and low risk students (GPA≥2.5, n=558) giving a dataset of n=1017. Borderline students (2.0 ≤ GPA ≤ 2.49) have not been considered to date. Gray et al. [24] found that cross validation over-estimated model accuracy compared to models applied to a different student cohort. Therefore models were trained on participants from 2010 and 2011 and tested on participants from 2012. All datasets were balanced by over sampling the minority class, and attributes were scaled to have a mean of 0 and standard deviation of 1. Significant attributes were identified by finding the optimal threshold for selecting attributes by weight. Attributes were weighted based on uncertainty[4] for DT, k-NN and Naïve Bayes models, and based on SVM weights for SVM models. Table 6 shows the accuracies achieved and factors used in each model.

k-NN had the highest accuracy for models of all students (66%). Accuracies for DT (61%), SVM (62%) and Naïve Bayes (62%) were similar. The most significance attributes by weight were age, deep learning approach and study effort. Including factors of prior academic performance improved model accuracy marginally to 72%.

Model accuracy improved when modelling each course separately. In general, k-NN had either the highest accuracy, or close to the highest accuracy, for all groups with the exception of two courses, international business and early childcare & education. Naïve Bayes had the highest accuracy for both those courses and their attributes of significance were normally distributed. Five courses had accuracies marginally higher than the model for all students, social & community development (70%), applied social care (68%), early childcare & education (69%), creative digital media (67%) and sports management (70%). As illustrated in Table 1, these courses were distinguished by a high average GPA and a low failure rate. Consequently, patterns identifying high risk students may be under represented in these groups. Accuracies for other courses were significantly higher (≥ 79%). For example the difference between sports management (70%) and the next highest accuracy (Engineering other, 79%) was significant (Z=5.86, p<0.001)[5].

[3] m=mean, s=standard deviation

[4] Symmetrical uncertainty with respect to the class label.

[5] Accuracy comparisons were based on the mean accuracy of

Table 5: Bootstrap correlations of non-cognitive factors with GPA

Study Factors:	Temperament		Motivation			Learning Approach			Learning Strategy			Other			Modality		
	C	O	SE	IM	EM	De	Sh	St	SR	ST	StE	Group	Age	Gen	V	A	K
Correlation with GPA (n=1207):	0.15 ***	0.08 **	0.12 ***	0.15 ***	0.12 ***	0.23 ***	-0.15 ***	-0.16 ***	0.13 ***	0.1 **	0.19 ***	-0.08 **	0.25 ***	0.09 **	0.06	0.02	0.06

*$p < .05$, **$p < .01$, ***$p < 0.001$; C:Conscientiousness; O:Openness; SE:Self Efficacy; IM:Intrinsic Goal Orientation; EM:Extrinsic Goal Orientation; De:Deep Learner; Sh: Shallow Learner; St: Strategic Learner; SR: Self Regulation; ST:Study Time; StE: Study Effort; Group:Likes to work in groups; Gen=Gender; V:Visual Learner; A:Auditory Learner; K:Kinaesthetic Learner.

Table 6: Regression and classification models by discipline, using non-cognitive factors only

Regression models:				Temperament		Motivation			Approach			Strategy			Other			Modality		
Course	N	Absolute error	R^2	C	O	SE	IM	EM	De	Sh	St	SR	ST	StE	G	age	In	V	A	K
All	1207	0.83±0.56	0.125	+	+	+	+	***	****	***	***	**		***	***	****	*	+		
Computing	137	0.8 ±0.56	0.34	+	+		+			**	+				*	****		*		
Creative Dig Media	103	0.68±0.58	0.11			+		+	****	****	****	+	+	+	+		***			
Eng Common Entry	73	0.67±0.53	0.34		*			+	+	+	+		+			***	***	+		+
Engineering other	99	0.72±0.5	0.43		+	***	+	+	+	**	**	**	*	+	*	****			+	
Horticulture	41	0.63±0.54	0.34	+	+	+	+	+	****	****	****	+	+	+	+	*	****		**	
General Business	183	0.9±0.53	0.13	+		+		+	+	+	+			+	+	+	**			+
Business With IT	60	0.67±0.52	0.48	+			**	**	*			+		***	**	**	**			**
International Business	64	0.78±0.5	0.27		***	+	+		*						+	*	****			+
Sports Management	95	0.64±0.53	0.16	+	+		+			**		+	***							
Applied Social Care	146	0.5±0.5	0.08	+	+	+	+	+	+	*	*	+	+	+	+		****			
Early childcare	80	0.37±0.34	0.17				+	+	+	*	**	+	+	+		+		+		
Social & Comm Dev	127	0.63±0.5	0.12				+	+	+	+				**	+					+
Classification models:				Temperament		Motivation			Approach			Strategy			Other			Modality		
Course	N	Learner	Accuracy	Kappa	C	O	SE	IM	EM	De	Sh	St	SR	ST	StE	G	age	gen	V	A	K
All	1017	11-NN	66%	0.33	✓	✓	✓	✓		✓	✓	✓	✓		✓		✓	✓			
Computing	122	SVM	81%	0.62		✓		✓	✓		✓				✓	✓			✓		✓
Creative Dig Media	94	2-NN	67%	0.35		✓		✓	✓	✓		✓			✓					✓	
Eng Common Entry	73	SVM	94%	0.88	✓	✓				✓	✓	✓			✓		✓		✓		✓
Engineering other	72	DT	79%	0.58						✓					✓		✓				
Horticulture	40	7-NN	86%	0.71	✓	✓	✓	✓	✓		✓	✓			✓					✓	✓
Business General	156	5-NN	85%	0.69										✓	✓						
Business With IT	47	7-NN	83%	0.67		✓	✓	✓			✓	✓	✓		✓		✓				✓
International Business	55	NB	80%	0.6		✓	✓														
Sports Mgmt	72	SVM	70%	0.39	✓				✓		✓		✓	✓							
Applied Social Care	122	4-NN	68%	0.37	✓		✓	✓		✓			✓	✓							✓
Early childcare	58	NB	69%	0.38	✓				✓		✓						✓				
Community dev	93	2-NN	70%	0.39	✓						✓						✓				

Significant model coefficients: +$p > .05$, *$p < .05$, **$p < .01$, ***$p < 0.001$, ****$p << 0.001$; ✓: factors included in the classification model
C:Conscientiousness; O:Openness; SE:Self Efficacy; IM:Intrinsic Goal Orientation; EM:Extrinsic Goal Orientation; De:Deep Learner; Sh: Shallow Learner; St: Strategic Learner; SR: Self Regulation; ST:Study Time; StE: Study Effort; G:Likes to work in groups; IN:Regression model intercept; gen=Gender; V:Visual Learner; A:Auditory Learner; K:Kinaesthetic Learner; Engineering others: Mechatronics and Electrical & Computer Engineering.

It could be argued that the smaller sample size of course groups over estimated model accuracy as smaller samples may under represent the complexity of patterns predictive of academic achievement. Therefore 30 samples randomly generated from the full dataset (n=100) were also modelled. Model accuracy for the random samples was normally distributed, with mean=63.12% (s=11%), which was marginally lower than the model of all students (Z=2.68, p=0.017).

There was little agreement across models on which study factors were most predictive of high risk and low risk students. Conscientiousness, study effort and a shallow learning approach were used most frequently, followed by openness, intrinsic motivation and age. There was no significant improvement in model accuracy when prior academic performance was included in each model. For example, the largest increase in accuracy was from 79% to 82% in a model of Engineering students.

4. CONCLUSIONS

Results from this study suggest that models of academic performance, based on non-cognitive psychometric factors measured during first year student induction, can achieve good predictive accuracy, particularly when individual courses are modelled separately. A deep learning approach, study effort and age had the highest correlations with GPA across all disciplines. These factors were also significant in both the regression model and classification model of all students. Extrinsic motivation, preference for working alone and self regulation were also significant in the regression model, while all factors except extrinsic motivation, preference for working alone and study time were significant in a classification model of all students. Models of individual courses also differed in the range of factors used. The lack of consensus in identification of significant factors may be explained by an overlap in the constructs measured by each [24]. Openness appeared frequently in both classification and regression models despite its relatively low correlation with GPA.

In general, regression models for students in technical disciplines, such as engineering, computing and business with IT, had a higher coefficient of determination (R^2) than models of non technical disciplines. However the coefficient of determination did not reflect prediction error, highlighting the underlying variability in independent variables. For example, early childcare (R^2=0.17) and sports management (R^2=0.16) had the same R^2, but sports management had a higher absolute error (0.64±0.53) than early childcare (0.37 ± 0.34). The difference was significant (t(15)=3.996, p=0.001). Prediction error was reflective of the GPA distribution for each course regardless of discipline.

Classification models that distinguished between high and low risk students based on GPA had good accuracy for both technical and non technical disciplines, particularly for courses with a significant proportion (>30%) of high risk students. As with regression, models of individual courses outper-

100 bootstrap samples from each group.

formed both models of the full dataset and models of random samples taken from the full dataset. This would suggest models trained for specific courses can outperform models generalising patterns for all students. k-NN, a non-linear classification algorithm, gave optimal or near optimal accuracies for most course groups. This may be reflective of non-linear patterns in the dataset.

Including a cognitive factor of prior academic performance did not improve the accuracy of classification models significantly. On the other hand, Gray et al. [23] reported that predictive accuracy of models based on cognitive factors only (prior academic performance) increased marginally when non-cognitive factors were included in the model. This would suggest a high overlap in constructs captured by both cognitive and non-cognitive factors of learning.

Model accuracies are based on a heuristic search of attribute subsets. A more exhaustive search is needed to verify optimal attribute subsets. Further work is also required to investigate principal components amongst non-cognitive factors. In addition, results are based on full time students in a traditional classroom setting at one college. Further work is needed to determine if these results generalise to students in other colleges, and other delivery modes.

5. ACKNOWLEDGMENTS

The authors would like to thank Institute of Technology Blanchardstown for their support in facilitating this research, and staff at the National Learning Network for assistance administering questionnaires during student induction.

6. REFERENCES

[1] Achen, C. *Intrepreting and Using Regression*. Number 07-029 in Quantitative Applications in the Social Sciences. Sage Publications, Inc, 1982.

[2] Baker, R. S. J. D. and Yacef, K. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1(1):3–17, 2010.

[3] Bergin, S. *Statistical and machine learning models to predict programming performance*. PhD thesis, Computer Science, NUI Maynooth, 2006.

[4] Bidjerano, T. and Dai, D. Y. The relationship between the big-five model of personality and self-regulated learning strategies. *Learning and Individual Differences*, 17:69 – 81, 2007.

[5] Biggs, J., Kember, D., and Leung, D. The revised two-factor study process questionnaire: R-SPQ-2F. *British Journal of Education Psychology*, 71:133–149, 2001.

[6] Buckingham Shum, S. and Deakin Crick, R. Learning dispositions and transferable competencies. pedagogy, modelling and learning analytics. In *2nd International Conference on Learning Analytics and Knowledge*, pages 92–101, Vancouver, BC, Canada, 2012.

[7] Carpenter, J. and Bithell, J. Bootstrap confidence intervals - when, which, what? A practical guide for medical statisticians. *Statistics in Medicine*, 19:1141–1164, 2000.

[8] Cassidy, S. Exploring individual differences as determining factors in student academic achievement in higher education. *Studies in Higher Education*, 37(7):1–18, 2011.

[9] Chamorro-Premuzic, T. and Furnham, A. Personality, intelligence and approaches to learning as predictors of academic performance. *Personality and Individual Differences*, 44:1596–1603, 2008.

[10] Chatti, M. A., Dychhoff, A. L., Schroeder, U., and Thüs, H. A reference model for learning analytics. *International Journal of Technology Enhanced Learning. Special Issue on State of the Art in TEL*, pages 318–331, 2012.

[11] Chow, G. C. Tests of equality between sets of coefficients in two linear regressions. *Econometrica*, 28(3):591–605, 1960.

[12] Cooper, A. J., Smillie, L. D., and Corr, P. J. A confirmatory factor analysis of the mini-IPIP five-factor model personality scale. *Personality and Individual Differences*, 48(5):688–691, 2010.

[13] Covington, M. V. Goal theory, motivation, and school achievement: An integrative review. *Annual Review of Psychology*, 51:171–200, 2000.

[14] De Feyter, T., Caers, R., Vigna, C., and Berings, D. Unraveling the impact of the big five personality traits on academic performance. The moderating and mediating effects of self-efficacy and academic motivation. *Learning and Individual Differences*, 22:439–448, 2012.

[15] Dekker, G., Pechenizkiy, M., and Vleeshouwers, J. Predicting students drop out: a case study. In Barnes, T., Desmarais, M. C., Romero, C., and Ventura, S., editors, *Proceedings of the 2nd International Conference on Educational Data Mining*, pages 41–50, Cordoba, Spain, 2009.

[16] Drachsler, H. and Greller, W. The pulse of learning analytics. Understandings and expectations from the stakeholders. In *2nd International Conference on Learning Analytics and Knowledge*, pages 120–129, Vancouver, BC, Canada, 29 April- 2 May 2012. ACM.

[17] Duff, A., Boyle, E., Dunleavy, K., and Ferguson, J. The relationship between personality, approach to learning and academic performance. *Personality and Individual Differences*, 36:1907–1920, 2004.

[18] Entwhistle, N. Contrasting perspectives in learning. In Marton, F., Hounsell, D., and Entwhistle, N., editors, *The Experience of Learning*, pages 3–22. Edinburgh: University of Edinburgh, Centre for Teaching, Learning and Assessment, 2005.

[19] Eppler, M. A. and Harju, B. L. Achievement motivation goals in relation to academic performance in traditional and nontraditional college students. *Research in Higher Education*, 38 (5):557–573, 1997.

[20] Farsides, T. and Woodfield, R. Individual differences and undergraduate academic success: The roles of personality, intelligence, and application. *Personality and Individual Differences*, 34:1225–1243, 2003.

[21] Fleming, N. D. I'm different, not dumb. Modes of presentation (VARK) in the tertiary classroom. *Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia*, 18:308–313, 1995.

[22] Goldberg, L. R. The development of markers for the

big-five factor structure. *Psychological Assessment*, 4 (1):26–42, 1992.

[23] Gray, G., McGuinness, C., and Owende, P. An investigation of psychometric measures for modelling academic performance in tertiary education. In D'Mello, S. K., Calvo, R. A., and Olney, A., editors, *Sixth International Conference on Educational Data Mining*, pages 240–243, Memphis, Tennessee, July 6-9 2013.

[24] Gray, G., McGuinness, C., and Owende, P. An application of classification models to predict learner progression in tertiary education. *4th IEEE International Advanced Computing Conference*, pages 549–554, February 2014.

[25] Gray, G., McGuinness, C., Owende, P., and Carthy, A. A review of psychometric data analysis and applications in modelling of academic achievement in tertiary education. *Journal of Learning Analytics*, 1(1):75–106, 2014.

[26] Kang, Y. and Harring, J. R. Reexamining the impact of non-normality in two-group comparison procedures. *Journal of Experimental Education*, in press.

[27] Kappe, R. and van der Flier, H. Using multiple and specific criteria to assess the predictive validity of the big five personality factors on academic performance. *Journal of Research in Personality*, 44:142–145, 2010.

[28] Kaufman, J. C., Agars, M. D., and Lopez-Wagner, M. C. The role of personality and motivation in predicting early college academic success in non-traditional students at a hispanic-serving institution. *Learning and Individual Differences*, 18:492 – 496, 2008.

[29] Knight, S., Buckingham Shum, S., and Littleton, K. Epistemology, pedagogy, assessment and learning analytics. In *Third Conference on Learning Analytics and Knowledge (LAK 2013)*, pages 75–84, Leuven, Belgium, April 2013.

[30] Komarraju, M., Karau, S. J., Schmeck, R. R., and Avdic, A. The big five personality traits, learning styles, and academic achievement. *Personality and Individual Differences*, 51:472–477, 2011.

[31] Komarraju, M. and Nadler, D. Self-efficacy and academic achievement. Why do implicit beliefs, goals, and effort regulation matter? *Learning and Individual Differences*, 25:67–72, 2013.

[32] Marton, F. and Säljö, R. Approaches to learning. In Marton, F., Hounsell, D., and Entwhistle, N., editors, *The Experience of Learning*, pages 36–58. Edinburgh: University of Edinburgh, Centre for Teaching, Learning and Assessment, 2005.

[33] Mislevy, R. J., Behrens, J. T., and Dicerbo, K. E. Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *Journal of Educational Data Mining*, 4 (1):11–48, 2012.

[34] Moran, M. A. and Crowley, M. J. The leaving certificate and first year university performance. *Journal of Statistical and Social Enquiry in Ireland*, XXIV, part 1:231–266, 1979.

[35] Ning, H. K. and Downing, K. The reciprocal relationship between motivation and self-regulation: A longitudinal study on academic performance. *Learning and Individual Differences*, 20:682–686, 2010.

[36] Pardos, Z. A., Baker, R. S. J. D., San Pedro, M. O. C. A., Gowda, S. M., and Gowda, S. M. Affective states and state test. Investigating how affect throughout the school year predicts end of year learning. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge (LAK '13)*, pages 117–124, Leuven, Belgium, April 2013. ACM.

[37] Pintrich, P., Smith, D., Garcia, T., and McKeachie, W. A manual for the use of the motivated strategies for learning questionnaire. Technical Report 91-B-004, The Regents of the University of Michigan, 1991.

[38] Poropat, A. E. A meta-analysis of the five-factor model or personality and academic performance. *Psychological Bulletin*, 135(2):322–338, 2009.

[39] Robbins, S. B., Lauver, K., Le, H., Davis, D., and Langley, R. Do psychosocial and study skill factors predict college outcomes? A meta analysis. *Psychological Bulletin*, 130 (2):261–288, 2004.

[40] Sachin, B. R. and Vijay, S. M. A survey and future vision of data mining in educational field. In *Advanced Computing Communication Technologies (ACCT), 2012 Second International Conference on*, pages 96–100, Jan 2012.

[41] Shute, V. and Ventura, M. *Stealth Assessment. Measuring and Supporting Learning in Video Games*. The John D. and Catherine T. MacArthur Foundation Reports on Digital Media and Learning. MIT Press, 2013.

[42] Siemens, G. Learning analytics. Envisioning a research discipline and a domain of practice. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 4–8, 2012.

[43] Siemens, G. and Baker, R. S. J. D. Learning analytics and educational data mining. Towards communication and collaboration. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 252–254, 2012.

[44] Swanberg, A. B. and Martinsen, Ø. L. Personality, approaches to learning and achievement. *Educational Psychology*, 30(1):75–88, 2010.

[45] Tavakol, M. and Dennick, R. Making sense of Cronbach's alpha. *International Journal of Medical Education*, 2:53–55, 2011.

[46] Tempelaar, D. T., Cuypers, H., van de Vrie, E., Heck, A., and van der Kooij, H. Formative assessment and learning analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge (LAK '13)*, pages 205–209, New York, NY, USA, 2013. ACM.

[47] Tishman, S., Jay, E., and Perkins, D. N. Teaching thinking disposition: From transmission to enculturation. *Theory into Practice*, 32:147–153, 1993.

[48] Vancouver, J. B. and Kendall, L. N. When self-efficacy negatively relates to motivation and performance in a learning context. *Journal of Applied Psychology*, 91(5):1146–53, 2006.

[49] Volet, S. E. Cognitive and affective variables in academic learning: the significance of direction and effort in students' goals. *Learning and Instruction*, 7(3):235–254, 1996.

Workshop Approaching Twenty Years of Knowledge Tracing (BKT20y)

Knowledge Tracing is an extremely popular method for student modeling because of its capability to infer a student's dynamic knowledge state in real time as the student is observed solving a series of problems (Corbett & Anderson, 1995). After its introduction in 1995, many extensions to the original technique have been proposed to improve its predictive accuracy. Variants include: fitting model parameters to individuals rather than populations (e.g., Lee & Brunskill, 2012; Yudelson, Koediger, & Gordon, 2010), contextualizing model parameters based on past and current usage of an intelligent tutoring system (Baker, Corbett, & Aleven, 2008, Baker et al., 2010; GonzálezBrenes, 2014; Pardos et al., 2010) and on latent characteristics of students and problems (Khajah et al, 2014), clustering similar students and sharing parameters among them (Pardos et al, 2012), soft sharing of parameters via hierarchical Bayesian inference (Beck & Chang, 2007; Beck, 2007), and considering knowledge state as a continuous variable (SohlDickstein, 2013; Smith et al., 2004).

As we approach twenty years since the introduction of Knowledge Tracing, what lessons have we learned? This workshop's motivation is to open the floor for the discussion of the recent advances in Knowledge Tracing and student modeling in general, take stock of the promises and failures of current approaches, and work toward developing integrated approaches.

We gratefully acknowledge the following members of the workshop program committee:

Albert Corbett, Carnegie Mellon University
Neil Heffernan, Worcester Polytechic Institute
Zachary Pardos, University of California, Berkeley
Steve Ritter, Carnegie Learning, Inc.

The BKT20y workshop organizers
Michael Yudelson
José P. González-Brenes
Michael Mozer

Table of Contents BKT20y

Additionally, the workshop scheduling will include the full talk “EEG Helps Knowledge Tracing!” based on the following paper:

Xu, Y., K.-M. Chang, Y. Yuan, and J. Mostow. EEG Helps Knowledge Tracing! In Proceedings of the ITS2014 Workshop on Utilizing EEG Input in Intelligent Tutoring Systems. 2014: Honolulu, p. 43-48.

Choosing Sample Size for Knowledge Tracing Models *

Derrick Coetzee
University of California, Berkeley
dcoetzee@berkeley.edu

ABSTRACT

An important question in the practical application of Bayesian knowledge tracing models is determining how much data is needed to infer parameters accurately. If training data is inadequate, even a perfect inference algorithm will produce parameters with poor predictive power. In this work, we describe an empirical study using synthetic data that provides estimates of the accuracy of inferred parameters based on factors such as the number of students used to train the model, and the values of the underlying generating parameters. We find that the standard deviation of the error is roughly proportional to $1/\sqrt{n}$ where n is the sample size, and that model parameters near 0 and 1 are easier to learn accurately.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Measurement,Theory.

Keywords

Educational data mining,knowledge tracing,sample size

1. INTRODUCTION

Simple Bayesian knowledge tracing models a student's observed responses to a sequence of items as a Markov process, with their knowledge state as a hidden underlying variable. If values are given for the four standard parameters, learning rate, prior, guess, and slip, the likelihood of a particular set of response sequences can be computed. Using standard search procedures like expectation maximization (EM), the parameter set giving the highest likelihood for a given set of sequences can be determined, provided that the procedure converges to the global maximum.

However, even if the procedure identifies the global maximum correctly and precisely, the resulting parameters may not reflect the actual parameters that generated the data; this is a *sampling error* effect. It's clearest with very small samples, such as samples of size 1, but exists with larger samples as well. Empirical studies with synthetic data generated from known parameters show that the inferred parameters for a given data set can differ substantially from the generating parameters, and this same issue would arise in real settings. An understanding of the magnitude of sampling error in a particular scenario can help to explain why the resulting model does or does not make effective predictions. Moreover, by providing a means to describe the distribution of possible generating parameter values, the uncertainty of calculations based on those parameters such as predictions can also be determined.

2. RELATED WORK

For simple problems, such as identifying the mean value of a parameter in a population, or the proportion of the population falling into a subgroup, there are simple and well-understood statistical approaches for determining sample size based on statistical power. Such analytic approaches are not immediately applicable to the problem of minimizing the HMM error function because of its complexity and high dimensionality.

Falakmasir et al [2] have noted that training time increases linearly with the size of the training set. Choosing an appropriate sample size for a certain desired level of accuracy can thus help to reduce training time, which is important both for research and in some real-time interactive tutor applications.

Nooraei et al [3] found that using only the 15 most recent data points from each student to train a knowledge tracing model yielded root mean-square error during prediction comparable to using the student's full history. For one data set, the most 5 recent items sufficed. Our study conversely does not vary the number of items per student, but instead varies the number of students and the four parameters generating the data. By allowing sample size to be reduced to meet a desired accuracy, our work offers an orthogonal method of further reducing training time.

De Sande [8] has suggested that as samples become larger, models with small parameter sets may no longer be rich enough to capture the sample's complexity. Thus our exclu-

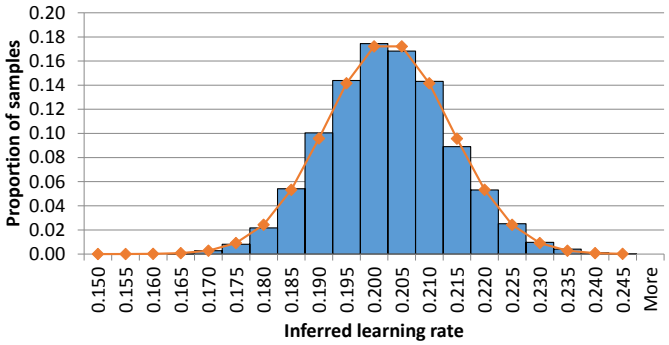


Figure 1: Given the fixed model learn=0.2, prior=0.4, guess=0.14, slip=0.05, we generated 10000 samples with 1000 students each, and for each, inferred all four parameters using EM. The distribution of the inferred learning rate parameter over the samples is above. The mean differs by 3×10^{-6} from the true generating parameter 0.2. The standard deviation is 0.01121, and the orange line shows the expected height of each bar if the proportions precisely followed a normal distribution. Scipy's normaltest [7] rejects that the distribution is perfectly normal ($p < 0.0002$), and a small amount of negative (left) skew is visible; the median is 0.00016 smaller than the mean. But the distribution is close enough to normal for our purposes.

sive reliance on a simple four-parameter BKT model even for very large samples is a limitation of our approach.

3. METHODOLOGY

In our experiments we relied on a simple standard Bayesian knowledge tracing model with four parameters: learning rate, prior, guess and slip. There is only one value for each parameter, and no specialization by student or problem. Each synthetic student responded to five items; we do not vary this parameter in this study, since Nooraei et al [3] report that increasing this parameter has diminishing returns, but future work may investigate it.

We generate separate datasets for each of our experiments. In each case, we enumerate a sequence of models (each specified by values for learn, prior, guess, slip, sample size), and for each of those models, we generate a large number of random samples consistent with that model. For example, for a particular model, we may generate 1000 samples each containing 1000 students.

We then run EM on each sample to find the parameter set giving the maximum likelihood value. All parameters are permitted to vary during the search. EM is run starting at the generating parameters and run until fully converged (within 10^{-12} or until 100 iterations are complete). Starting at the generating parameters is not feasible in a realistic setting, but here it allows EM to run quickly and consistently reach the global minimum. As shown in Figure 1, the parameter values inferred from these samples approximate a normal distribution with a mean equal to the generating parameter.

Finally, we take all samples generated from a single model and, for each parameter, record the mean and standard deviation of the inferred values for that parameter. We chose the number of samples generated for each model large enough so that these statistics remain stable under repeated runs. Mean values for each parameter were consistently near the generating parameter, typically within at most 0.1 standard deviations. Standard deviation provides an estimate of variation in the inferred parameter values, and is plotted. Different models yield different standard deviation values.

Because of the very large number of large samples involved in this approach, we use the *fastHMM* C++ BKT library designed by Pardos and Johnson [5] to quickly generate datasets and perform EM, invoked from a Matlab script.

3.1 Varying one parameter

In our first experiment, we start with typical, plausible values for all four parameters: learn=0.2, prior=0.4, guess=0.14, slip=0.05. These values are consistent with prior work that found large guess and slip values (> 0.5) to be implausible in most scenarios [6], and in our 5-problem scenario, the chance of learning the material by the end is about 67%, which is reasonable.

Then, for each of the four parameters, we hold the other parameters at their single plausible value, and vary the remaining parameter from 0 to 1 in steps of 0.01. This results in 404 total parameter sets.

For each parameter set, we generate 1000 random samples of 1000 students each. In this experiment, the number of students is fixed at 1000, which is large enough to consistently produce a standard deviation not exceeding 0.03 — this avoids the boundary effects near 0 and 1 that would occur for very small samples.

In this experiment, we focus on the variance of our estimates of the parameter that is being varied, and don't consider variance of the other (fixed) parameters.

3.2 Interactions between parameters

In this experiment, similiar to the first, we hold three parameters fixed (learn=0.2, prior=0.4, guess=0.14), and vary slip between 0 and 1 in steps of 0.01. This gives 101 parameter sets. For each, we generate 1000 random samples of 1000 students each. However, in this experiment we examine variance of our estimates of all four parameters, rather than just the one being varied (slip). This experiment helps to demonstrate to what extent varying one parameter can affect the difficulty of accurately inferring other parameters.

3.3 Varying sample size

In our third experiment, we fix the value of all four parameters, but vary the sample size in powers of two from 2 to 2097152. For sample sizes below 10000, we generate 1000 samples of that size, while for those above we generate 100 samples. The parameter values are heuristically chosen based on the prior experiments above to generate large error values (but not necessarily the worst possible error). We examine how variation of our estimates of all four parameters varies with sample size, and identify any trends.

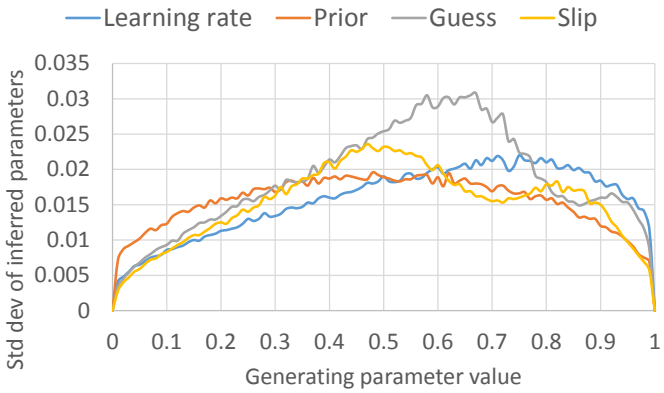


Figure 2: Variation of inferred parameters, based on underlying generating parameter. For each curve, all parameters other than one being examined are fixed at plausible values. Values near 0 and 1 are the easiest to infer accurately, and each parameter exhibits a unique pattern.

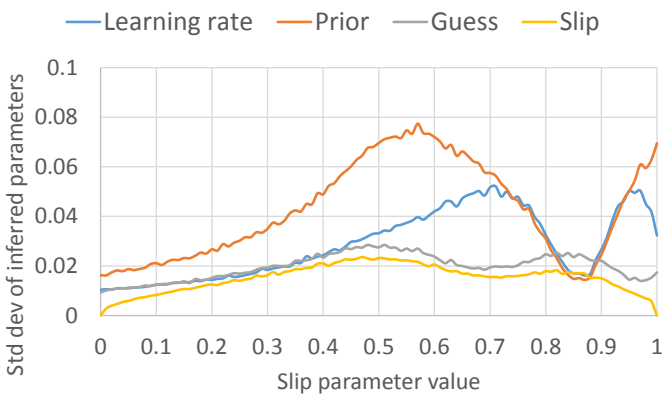


Figure 3: As the slip parameter is varied and the other parameters are held fixed (learn=0.2, prior=0.4, guess=0.14), the error in our inference of all other parameters varies in a strong and complex fashion, indicating interactions in the inference of different parameters.

3.4 Interaction between sample size and parameters

In our final experiment, we vary both the learning rate (from 0 to 1 in steps of 0.01) and the sample size (between the values 1000, 10000, 100000) at the same time. This enables us to examine whether there is any interaction between parameters and sample size. For 1000 and 10000 students we use 1000 samples, while for 100000 students we use 100 samples, to reduce runtime.

4. RESULTS

4.1 Varying one parameter

As described in section 3.1, in this experiment we vary each parameter between 0 and 1 while holding the other parameters fixed, and examined how the variation in our inference of that parameter changed with its value. As shown in Figure 2, parameters with values near 0 or 1 are easier to accurately estimate, while those with values in the 0.4 to 0.8 range are more difficult to infer. Each parameter exhibits a unique pattern, with prior behaving worst for small values, guess behaving worst for values in the middle, and learning rate performing worst for the largest values. Slip is unique in having two peaks in its curve near 0.5 and 0.8.

4.2 Interactions between parameters

As described in section 3.2, in this experiment we vary slip between 0 and 1 while keeping the other parameters fixed, and examine how the variation of all four inferred parameters varies, as shown in Figure 3. All variance values exhibit a strong, complex dependence on the slip parameter—in particular there is a dramatic and unexpected drop from large variance to small variance around slip=0.85. We conclude that the variance of an inferred parameter depends not only on the value of that parameter, but also the values of other parameters.

4.3 Varying sample size

We fix the parameters at the values empirically determined in section 4.1 to give maximum variance (roughly based on the maximums of the curves, with prior and guess at 0.5, and learning rate and slip at 0.67). Because section 4.2 suggests that there are interactions between parameters, this may not give the worst-case variance possible of all combinations, but it is a reasonable starting point for realistic values.

As described in section 3.3, sample size is varied in powers of two from 2 to 2097152. Figure 4 shows the result, suggesting that (except for very small samples) the standard deviation of the error is roughly proportional to $n^{-0.5}$, or $1/\sqrt{n}$, where n is the sample size. For these particular parameter values, slip is consistently inferred most accurately, learning rate is inferred least accurately, and guess and prior are between the two and are similar.

4.4 Interaction between sample size and parameters

In our final experiment, as described in section 3.4, we vary both the learning rate and the sample size at the same time. The standard deviation curves for the three sample sizes are then plotted on the same plot, each divided by the $1/\sqrt{n}$ factor, where n is the sample size, as shown in Figure 5. The curves are nearly identical, and we find no evidence of interaction between parameters and sample size, but we can't rule out interaction for other combinations of parameter values. This also offers additional evidence for the $1/\sqrt{n}$ trend from the previous section.

5. DISCUSSION

Because accuracy is good for parameter values near 0 and 1, this implies that for large enough samples, boundary effects (in which the distribution of error is skewed because values outside of the 0-1 range are not permitted) are not a serious concern.

Interactions between parameters are complex, suggesting that attempting to characterize error in each parameter independently is unlikely to yield good predictions of error. Moreover, attempts to model these interactions analytically

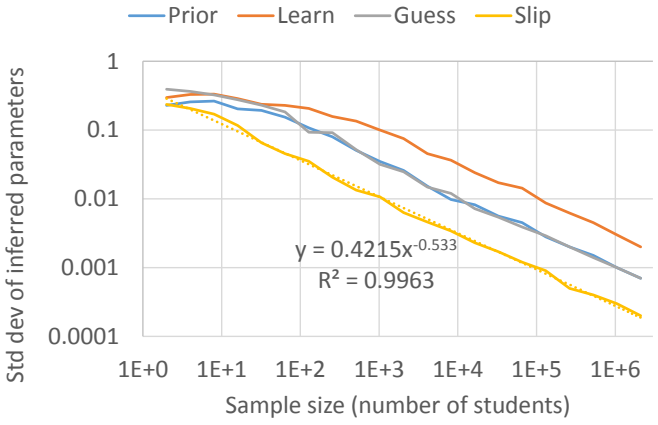


Figure 4: Accuracy of inferred parameters, based on sample size (training set size), with fixed parameters (prior=guess=0.5, learning=slip=0.67). This is a log-log plot, and (once the $y = 0.1$ level is reached) the lines each remain straight and have slope of roughly -0.5. This suggests that the standard deviation of the error is roughly proportional to $1/\sqrt{n}$, where n is the sample size.

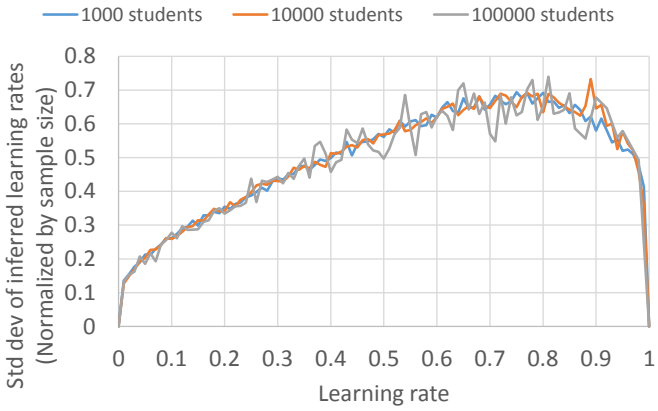


Figure 5: Here we vary learning rate from 0 to 1, and also vary sample size between the values 1000, 10000, and 100000. The resulting standard deviations are divided by $1/\sqrt{n}$ to normalize for improvement in error due to increased sample size. The resulting curves are nearly identical; the curve for 100000 students appears noisier only because of a lower number of samples (100 instead of 1000). We find no evidence of interaction between sample size and the learning rate.

may be challenging because they cannot be fit well by low-degree polynomials. A more viable strategy is to form a conservative estimate of error by conducting a grid search of parameter sets that are plausible in a given scenario. On the other hand, once the range of variances at a particular (sufficiently large) sample size is characterized, Figure 4 and Figure 5 show that altering the sample size has a uniform and predictable effect on the error.

The main result that standard deviation is proportional to $1/\sqrt{n}$ suggests that, in order to decrease the margin of error in the estimate of a parameter by a factor of 2, an increase in sample size by a factor of 4 is required. Additionally, Figure 4 shows that achieving even a single valid significant digit in the learning rate requires sample sizes of 1000 students or more. This suggests that studies using BKT with less than 1000 students should be considered carefully for sampling error.

5.1 Confidence Intervals and Decreasing Training Time

As noted in Figure 1, provided that the sample size is large enough, the distribution of samples is approximated well by a normal distribution, and the standard deviations computed in synthetic simulations such as the preceding ones can be used to compute confidence intervals containing the true generating parameters (e.g. 95% of possible values are within two standard deviations). Parameters used in these simulations can be set either by using domain knowledge, and/or by conservatively selecting values that give poor accuracy.

To use our results to decrease training time for a large data set, one approach is to create many small samples (e.g. 100 of size 1000) by sampling uniformly randomly with replacement from the full data set. By training on these, we can estimate the variance of our estimates of each parameter at a sample size of 1000. Then, given a desired level of accuracy and a desired probability of achieving it, we can use $1/\sqrt{n}$ to estimate the best final sample size. If the estimated sample size exceeds the data size, this suggests that more data needs to be gathered.

6. IDENTIFIABILITY PROBLEM

Although we have in this work considered a particular generating parameter set to be the correct and desired parameters, BKT exhibits an Identifiability Problem [1] in which there are an infinite family of four-parameter solutions that make the same predictions. This creates the risk that a solution that appears to be far from the generating parameters is actually very close to an equivalent parameter set (or an equivalent solution is).

Van de Sande [9] more specifically characterized BKT (in its HMM form) as a three-parameter system in which two systems having the same slip, learning rate, and A value will yield the same predictions, where A is given by

$$A = (1 - \text{slip} - \text{guess})(1 - \text{prior}).$$

One way to address the issue is to perform both data gener-

ation and parameter search in this reduced three-parameter system; this would be similar to our current approach, but error in the A parameter is more difficult to interpret. Intuitively, we expect search in a lower-dimensional space to give better accuracy with the same amount of data. However, Van de Sande also notes that the algorithm form of BKT has no analytic solution, and so the degree to which BKT is underdetermined may depend on the specific application.

Beyond the underdetermined nature of BKT, there are also information-theoretic bounds that limit the accuracy of inferring parameters regardless of the system. In particular, given a collection of at least k different parameter sets, and student data that can only take on $< k$ values, there is no procedure that can reliably infer the generating parameters without error. As the size of the data continues to decrease, the minimum possible error increases. Although these bounds are general, they typically apply only to very small data sets.

7. CONCLUSIONS AND FUTURE WORK

We've only explored a small part of the space of input parameters that can affect inferred parameter accuracy; the possible interactions between parameters are complex and not fully understood. It would also be useful to examine different sizes of problem sets, scenarios where different students complete different numbers of problems, models where parameters such as learning rate and guess/slip are per problem, and models where priors are measured per student (as in Pardos and Heffernan [4]).

Although it seems intuitive that insufficient sample size can lead to poor parameter estimates with poor predictive power, this deserves verification: it's not clear which errors will damage prediction and which are benign. An empirical synthetic study that examines prediction accuracy could assess this cheaply. Going a step further, it would be useful to simulate an interactive tutoring system and assess a cost function that penalizes the system for both incorrect assessment of mastery, and for failing to assess mastery when it is reached. By applying weights to these error types, the simulation could represent the real-world cost of inaccurate parameters in such a system.

Another important direction is extending our results to real-world data. There are a few approaches. One is to use a very large real-world data set and use its inferred parameters as the ground-truth generating parameters, then examine smaller subsets to determine whether parameters are inferred less accurately. If the BKT model is appropriate, we expect to observe similar relationships between sample size and variance as with our synthetic data. This approach can be compared to one experiment of Ritter [6] (Figure 4), in which they took a large real data set and computed mean-squared error using the best-fit parameters on subsets with smaller number of students ranging from 5 to 500.

There are other approaches to real-world validity. One would be a survey of prior BKT applications, to identify whether there is a consistent relationship between sample size and reported prediction accuracy. A third approach would be a controlled experiment in which two groups of very different sizes each use an ITS, the BKT is trained on the resulting data, and then the groups continue to use the ITS and their learning performance is examined (note however that asymmetric group sizes limit statistical power).

Finally, an analytical model that can explain some of our empirical results—such as the skewed normal distribution of inferred parameter values, the improvements in parameter inference near 0 and 1 parameter values, or the $1/\sqrt{n}$ relationship between sample size and standard deviation—would be a valuable contribution.

8. ACKNOWLEDGMENTS

We thank Zachary A. Pardos for his *fastHMM* C++ BKT library [5], for providing helpful comments on this work, and for designing the assignment which inspired it.

9. REFERENCES

[1] J. E. Beck and K.-M. Chang. Identifiability: A fundamental problem of student modeling. In *Proceedings of the 11th International Conference on User Modeling*, UM '07, pages 137–146, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] M. H. Falakmasir, Z. A. Pardos, G. J. Gordon, and P. Brusilovsky. A spectral learning approach to knowledge tracing. In *6th International Conference on Educational Data Mining (EDM 2013).*, pages 28–35. International Educational Data Mining Society, 2013.

[3] B. B. Nooraei, Z. A. Pardos, N. T. Heffernan, and R. S. J. de Baker. Less is more: Improving the speed and prediction power of knowledge tracing by using less data. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, and J. C. Stamper, editors, *EDM*, pages 101–110. www.educationaldatamining.org, 2011.

[4] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In P. D. Bra, A. Kobsa, and D. N. Chin, editors, *UMAP*, volume 6075 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2010.

[5] Z. A. Pardos and M. J. Johnson. Scaling cognitive modeling to massive open environments (in preparation). 2015.

[6] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle. Reducing the knowledge tracing space. In T. Barnes, M. C. Desmarais, C. Romero, and S. Ventura, editors, *EDM*, pages 151–160. www.educationaldatamining.org, 2009.

[7] SciPy v0.13.0 Reference Guide: scipy.stats.normaltest. `http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html`, May 2013. [Online; accessed 24-April-2014].

[8] B. Van de Sande. Applying three models of learning to individual student log data. In *6th International Conference on Educational Data Mining (EDM 2013).*, pages 193–199. International Educational Data Mining Society, 2013.

[9] B. Van de Sande. Properties of the bayesian knowledge tracing model. *Journal of Educational Data Mining*, 5(2):1–10, 2013.

A Unified 5-Dimensional Framework for Student Models

Yanbo Xu and Jack Mostow
Carnegie Mellon University Project LISTEN
RI-NSH 4103
5000 Forbes Ave, Pittsburgh, PA 15213
{yanbox, mostow}@cs.cmu.edu

ABSTRACT
This paper defines 5 key dimensions of student models: whether and how they model time, skill, noise, latent traits, and multiple influences on student performance. We use this framework to characterize and compare previous student models, analyze their relative accuracy, and propose novel models suggested by gaps in the multi-dimensional space. To illustrate the generative power of this framework, we derive one such model, called HOT-DINA (Higher Order Temporal, Deterministic Input, Noisy-And) and evaluate it on synthetic and real data. We show it predicts student performance better than previous methods, when, and why.

Keywords
Knowledge tracing, Item Response Theory, temporal models, higher order latent trait models, multiple subskills, DINA.

1. Introduction
Morphological analysis [1] is a general method for exploring a space of possible designs by identifying key attributes, specifying possible values for each attribute, and considering different combinations of choices for the attributes. Structuring the space in this manner compares different designs in terms of which attribute values they share, and which ones differ. Characterizing the space of existing designs in terms of these attributes exposes gaps in the space, suggesting novel combinations to explore.

Some prior work on student modeling has used this approach to characterize spaces of possible knowledge tracing models. Knowledge tracing (KT) [2] generally has 4 or 5 parameters: the probability *slip* of failing on a known skill; the probability *guess* of succeeding on an unknown skill; the probability *knew* of knowing a skill before practicing it; the transition probability *learn* from not knowing the skill to knowing it; and sometimes the transition probability *forget* from knowing the skill to not knowing it, usually assumed to be zero.

Mostow et al. [3] defined a space of alternative parameterizations of a given KT model, based on whether they assigned each knowledge tracing parameter a single overall value, a distinct value for each individual student and/or skill, or different values for different categories of students and/or skills. Thus the number of values to fit is 4 if using a single global value for each parameter, but with separate probabilities for each <student, skill> pair, the number of values to fit is 4 × # students × # skills. This work ordered the space of possible parameterizations of a single model by the number of values to fit.

Xu and Mostow [4] factored the space of different knowledge tracing models in terms of three attributes: how to ***fit*** their parameters, how to ***predict*** students' performance from their estimated knowledge, and how to ***update*** those estimates based on observed performance. We will use this factoring in Section 3.2.

Section 2 introduces the proposed framework. Section 0 describes HOT-DINA, a novel knowledge tracing method that the framework inspired. Sections 4 and 5 evaluate HOT-DINA on synthetic and real data, respectively. Section 6 concludes.

2. A Unified 5-Dimensional Framework
We characterize student models in terms of these five dimensions:

Temporal effect: skills time-invariant vs. time-varying.
- Static, e.g. IRT [5] and PFA [6]
- 2 or more fixed time points, e.g. at pre- and post-test
- Dynamic, e.g. KT [2]

Skill dimensionality: single skill vs. multiple skills at a step.

Credit assignment: how credit (or blame) is allocated among influences on the observed success (or failure) of a step. Mostow et al. [3] define a space of KT parameterizations. Corbett and Andersen [2] originally fit KT per skill. Pardos and Heffernan [7] individualized KT and fit parameters per student. Wang and Heffernan [8] simultaneously fit KT per student and per skill. In contrast, multiple-skills models require combination functions to assign credit or blame among the skills. Product KT [9] assigns full responsibility to each skill and multiplies the estimates. Conjunctive KT [10] assigns fair credit or blame to skills and multiplies the estimates. Weakest KT [11] credits or blames the weakest skill and takes the minimum of the estimates. LR-DBN [12] apportions credit or blame and performs logistic regression over the estimates. We summarize credit assignment methods as:
- Contingency table
  - Per student
  - Per skill
  - Per <student, skill>
  - Per student + per skill
- Binary or probabilistic
  - Conjunctive (min)
  - Independent (product)
  - Disjunctive (max)
- Other
  - Compensatory (+)
  - Mixture (weighted average)
  - Logistic regression (sigmoid)

Higher order: treat static student properties as latent traits or not. We say IRT [5] models "higher order" effects because it estimates static student proficiencies independent of skill properties such as skill difficulty in 1PL (1 Parameter Logistic), skill discrimination in 2PL, and skill guess rate in 3PL. De la Torre [13] first combined IRT with static Cognitive Diagnosis Models such as

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

NIDA (Noisy Inputs, Deterministic And Gate) [14-16] and DINA (Deterministic Inputs, Noisy And Gate), and proposed higher order latent trait models (HO-NIDA and HO-DINA). Xu and Mostow [17] used IRT to estimate the probability of knowing a skill initially in a higher order knowledge tracing model (HO-KT).

Noise: how to represent errors in model, or discrepancies between what a student knows versus does. KT assumes students may guess a step correctly even though they don't know its underlying skill(s), or slip at a step even though they know its skill(s). Such "noise" is also characterized in other models, including single-skill KT variants such as PPS (Prior Per Student) [7] and SSM (Student Skill Model) [8], and IRT models such as 3PL. NIDO and DINO respectively add noise either before or after combining estimates of multiple skills. We refer to these noise modeling methods as:

- None
- Slip/Guess
- NIDO (noisy input, deterministic output)
- DINO (deterministic input, noisy output)

Table 1 summarizes student models in the proposed unified 5-dimensional framework. Note that we only discuss known cognitive models (e.g. Q-matrix) in this paper, so we omit methods that discover unknown cognitive models [18, 19].

Table 1. A unified 5-dimensional framework for student models

Student models	Temporal effect	Skill dimensionality	Credit assignment	Higher order effect	Noise model
IRT 1PL (Rasch model) [5] IRT 2PL (2 Parameter Logistic) [5]	Static	Single skill	Per student + per skill	Latent trait	None
IRT 3PL (3 Parameter Logistic) [5]					Slip/Guess
LLM (Linear Logistic Model) [16]		Multiple skills	Sigmoid	No latent trait	None
LFA (Learning Factor Analysis) [20] PFA (Performance Factor Analysis) [6]					
NIDA [14-16]			Product		NIDO
DINA [14-16]					DINA
LLTM (Linear Logistic Test Model) [21]			Sigmoid	Latent trait	None
HO-NIDA [13]			Product		NIDO
HO-DINA [13]					DINO
KT [2]	Dynamic	Single skill	Per skill	No latent trait	Slip/Guess
PPS (Prior Per Student) [7]			Per student		
SSM (Student Skill Model) [8]			Per student + Per skill		
HO-KT [17]				Latent trait	
DIR (Dynamic IRT 1PL) [22]					None
KT+NIDA [23]		Multiple skills	Product	No latent trait	NIDO
Product KT [9]					
CKT [10]					
Weakest KT [11]			Minimum		
KT+DINA [23]			Product		DINO
LR-DBN [12]			Sigmoid		
HOT-NIDA [Section 0]			Product	Latent trait	NIDO
HOT-DINA [Section 0]					DINO

Table 2. Comparative framework to train, predict and update multiple-skills models

Student models	Train	Predict	Update
CKT	Train skills separately. Assign each skill full responsibility.	Multiply skill estimates.	Update skills together. Bayes' equations assign responsibility.
Product KT			Update skills separately, each with full responsibility.
Weakest KT (Blame weakest, credit rest)		Minimum of skill estimates.	
Weakest KT (Update weakest skill)			Update only the weakest skill.
HOT-NIDA HOT-DINA [Section 3.2]	Train skills together. Assign each skill full responsibility.	Multiply skill estimates.	
KT+NIDA/DINA			Update skills together, each with full responsibility.
LR-DBN	Train skills together. Logistic regression assigns responsibility.	Logistic regression on skill estimates.	Update skills together. Logistic regression assigns responsibility.

Table 2 (adapted from [4]) expands **Credit assignment** in terms of how to **train**, **predict** and **update** skills, e.g. to assign full responsibility to every skill, blame the weakest skill and credit the rest, update only the weakest skill, or use logistic function.

The tables suggest transformations of models along the dimensions in the framework. For example, Dynamic IRT [22] varies student proficiency by time, transforming static IRT to dynamic. KT+NIDA/DINA [23] varies skill estimates by time, transforming static NIDA/DINA to dynamic. HO-NIDA/DINA/KT adds latent traits, transforming NIDA/DINA/KT to higher order. LLM [16] and LLTM [21] change the combination function, transforming conjunctive models to logistic models. In Section 0 we generate a novel student model by transforming HO-KT to a multi-skill model.

3. A Higher-Order Temporal Student Model to Trace Multiple Skills: HOT-DINA

Xu and Mostow [17] extended the static IRT model into HO-KT (Higher Order Knowledge Tracing), which accounts for skill-specific learning by using the static IRT model to estimate the probability Pr(*knew*) of knowing a skill before practicing it. By generalizing to steps that require conjunctions of multiple skills, we arrive at a combined model we call HOT-DINA (Higher Order Temporal, Deterministic Input, Noisy-And). Note we can transform it into HOT-NIDA simply by changing its noise type.

3.1 HOT-DINA = IRT + KT + DINA

Let $\{\mathbf{Y}^{(0)}, \mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(t)}, \ldots\}$ denote a sequential dataset recorded by an intelligent tutor system, where $Y_{nj}^{(t)} = 1$ iff student n correctly performs a step that requires skill j at time t. KT is a Hidden Markov Model (HMM) that models a binary hidden state $\mathbf{K}^{(t)}$ indicating if the student knows the skill at time t. The probability of knowing the skill is *knew* at time $t = 0$, and then changes based on the student's observed performance on the skill, according to the standard KT parameters *slip*, *guess*, *learn*, and *forget* (usually set to zero).

KT can fit these four parameters (taking *forget* = 0) for each <student, skill> pair, but the resulting large number of values to fit is likely to cause over-fitting. Thus, Corbett and Andersen [2] originally proposed to estimate *knew* per student, and *learn, guess* and *slip* per skill. IRT assumes a latent trait that represents a student's underlying proficiency in all the skills. For example, the Two Parameters Logistic (2PL) IRT model assumes that the probability of a student's correct response is a logistic function of a unidimensional student proficiency θ with two skill-specific parameters: discriminability a and difficulty b (see Equation 1).

$$P(Y = 1) = \frac{1}{1 + \exp(-1.7a(\theta - b))}$$

Equation 1. The logistic function of 2PL model

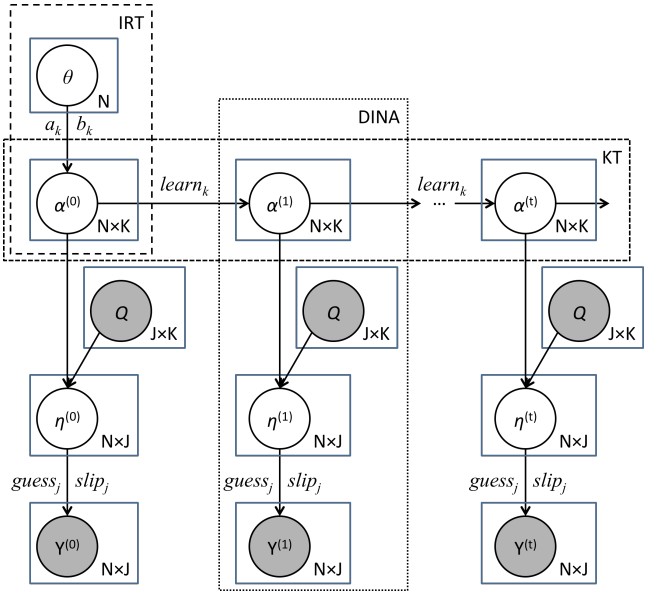
The two skill parameters determine the shape of the IRT curve. As a student's proficiency increases beyond the skill difficulty, the student's chance of performing correctly surpasses 50%. The skill discriminability reflects how fast the logit (log odds) increase or decrease when the proficiency changes. Thus IRT fits parameters individually on each dimension, without losing the information from the other. HO-KT uses 2PL to estimate *knew* in KT, by fitting student specific proficiency θ_n, skill discriminability a_j and skill difficulty b_j. It then uses KT to trace each skill, by fitting skill-specific $learn_j$, $guess_j$ and $slip_j$. Thus, HO-KT models students' initial overall knowledge before they practice any skills; then it updates its estimates of students' knowledge of each individual skill by observing additional practice on the skill. It also models two attributes of the skills, difficulty and discriminability, which are assumed to be constants that do not change over time.

To incorporate DINA into HO-KT, we still model a hidden binary state in each step to indicate whether a student knows the overall skill used in the step, denoted as $\eta_{nj}^{(t)}$ for student n with skill j at time t. However, we also model a hidden binary state $\alpha_{nk}^{(t)}$ to indicate whether student n knows skill k at time t. Given a matrix $\mathbf{Q} = \{Q_{jk}\}$, indicating whether the overall skill j requires skill k, we conjoin the skills as follows:

$$\eta_{nj}^{(t)} = \prod_{k=1}^{K} (\alpha_{nk}^{(t)})^{q_{jk}}$$

Equation 2. Conjunction of skills in HOT-DINA

This formula gives us the DINA (Deterministic Input, Noisy-And gate) structure [15], with the conjunction as the "and" gate and *guess* and *slip* as the noise. Thus by combining HO-KT with DINA, we obtain the HOT-DINA higher order temporal model to trace multiple skills. Figure 1 shows how the plate diagram for HOT-DINA integrates IRT, KT, and DINA.



θ_n: Proficiency of student n.

a_k: Discrimination of subskill k.

b_k: Difficulty of subskill k.

Q_{jk}: 1 if step j requires subskill k;
0 otherwise.

$\alpha_{nk}^{(t)}$: 1 if student n knows subskill k at time t;
0 otherwise.

$\eta_{nj}^{(t)}$: 1 if student n knows the skill of step j at time t;
0 otherwise.

Figure 1. Graphical representation of Higher-Order Temporal DINA (HOT-DINA) to trace multiple skills

Equation 3 shows the formula for using 2PL to estimate the probability *knew* of a student knowing a skill at time *t = 0*:

$$P(knew_{nk}) = P\left(\alpha_{nk}^{(0)} = 1\right) = \frac{1}{1 + \exp\left(-1.7\, a_k(\theta_n - b_k)\right)}$$

Equation 3. 2PL to estimate *knew* in HOT-DINA

Equation 4 shows the formula for tracing the skills with skill-specific *learn* and zero *forget*:

$$P\left({\alpha_{nk}}^{(t)} = 1 \middle| {\alpha_{nk}}^{(t-1)} = 0\right) = learn_k$$

$$P\left({\alpha_{nk}}^{(t)} = 0 \middle| {\alpha_{nk}}^{(t-1)} = 1\right) = forget_k = 0$$

Equation 4. Knowledge tracing of skills in HOT-DINA

Equation 5 shows the likelihood of a student's performance given the hidden state $\eta^{(t)}$ and the skill-specific *guess* and *slip*:

$$L\left(Y_{nj}^{(t)} = 1 \middle| \eta_{nj}^{(t)}\right) = {guess_j}^{\left(1-\eta_{nj}^{(t)}\right)} \times (1 - slip_j)^{\eta_{nj}^{(t)}}$$

$$L\left(Y_{nj}^{(t)} = 0 \middle| \eta_{nj}^{(t)}\right) = (1 - guess_j)^{\left(1-\eta_{nj}^{(t)}\right)} \times {slip_j}^{\eta_{nj}^{(t)}}$$

Equation 5. Likelihood in HOT-DINA

3.2 How to Train, Predict, and Update

Following the organization of Table 2, Section 3.2.1 details how HOT-DINA trains the skills together and assigns each skill full responsibility; Section 3.2.2 specifies how HOT-DINA predicts student performance by using a product of skill estimates; and Section 3.2.3 shows how HOT-DINA updates the weakest skill.

3.2.1 Training the model with MCMC

We estimate the parameters of HOT-DINA using Markov Chain Monte Carlo (MCMC) methods, which require that we specify the prior distributions and constraints for every parameter. We assume that student general proficiency θ_n is normally distributed with mean 0 and standard deviation 1. The skill discrimination a_n is positive and uniformly distributed between 0 and 2.5, while the skill difficulty b_n is also normally distributed with mean 0 and standard deviation 1. *Learn* has prior Beta (1,1), whereas *guess* and *slip* have uniform prior from 0 to 0.4.

Thus, the priors on each parameter are:

$\theta_n \sim Normal(0,1)$

$b_k \sim Normal(0,1)$

$a_k \sim Uniform(0,2.5)$

$learn_k \sim Beta(1,1)$

$guess_j \sim Uniform(0,0.4)$

$slip_j \sim Uniform(0,0.4)$

We use the following conditional distributions for each node:

$\alpha_{nk}^{(0)} | \theta_n \sim Bernoulli(\{1 + \exp(-1.7\, a_k(\theta_n - b_k))\}^{-1})$

${\alpha_{nk}}^{(t)} |\, {\alpha_{nk}}^{(t-1)} = 0 \sim Bernoulli(learn_k)$

${\alpha_{nk}}^{(t)} |\, {\alpha_{nk}}^{(t-1)} = 1 \sim Bernoulli(1)$

${Y_{nj}}^{(t)} | {\eta_{nj}}^{(t)} = 0 \sim Bernoulli(guess_j)$

${Y_{nj}}^{(t)} | {\eta_{nj}}^{(t)} = 1 \sim Bernoulli(1 - slip_j)$

Given $\boldsymbol{\eta}$ as a conjunction of $\boldsymbol{\alpha}$, the likelihood of **Y** given $\boldsymbol{\eta}$, the conditional independence of $\boldsymbol{\alpha}^{(0)}$ given $\boldsymbol{\theta}$, and of $\boldsymbol{\alpha}^{(t)}$ given $\boldsymbol{\alpha}^{(t-1)}$, the posterior distribution of ***θ, a, b, a, η, learn (l), guess (g)*** and ***slip(s)*** given **Y** is

$$P(\boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{l}, \boldsymbol{g}, \boldsymbol{s} | \boldsymbol{Y}) \propto L(\boldsymbol{Y} | \boldsymbol{g}, \boldsymbol{s}, \boldsymbol{\eta}, \boldsymbol{\alpha}) P\left(\boldsymbol{\alpha}^{(0)} \middle| \boldsymbol{\theta}, \boldsymbol{a}, \boldsymbol{b}\right)$$

$$\left(\prod_{t=1}^{T} P\left(\boldsymbol{\alpha}^{(t)} \middle| \boldsymbol{\alpha}^{(t-1)}, \boldsymbol{l}\right)\right) P(\boldsymbol{\theta}) P(\boldsymbol{a}) P(\boldsymbol{b}) P(\boldsymbol{l}) P(\boldsymbol{g}) P(\boldsymbol{s})$$

3.2.2 Predicting student performance

For inference, we introduce uncertainty to η_{nj}, and rewrite the Equation 2 as follows:

$$P\left(\eta_{nj}^{(0)} = 1\right) = \prod_{k=1}^{K} \left(\frac{1}{\exp\left(-1.7 a_k(\theta_n - b_k)\right)}\right)^{q_{jk}}$$

$P\left(\eta_{nj}^{(t)} = 1\right) = \prod_{k=1}^{K} (P(\alpha_{nk}^{(t)} = 1))^{q_{jk}})$ for t = 1,2,3…

Equation 6. Conjunction of skills in HOT-DINA inference

Then we predict student performance by using Equation 7:

$$P\left(Y_{nj}^{(t)} = 1\right) = (1 - slip_j) P\left(\eta_{nj}^{(t)} = 1\right) + guess_j (1 - P\left(\eta_{nj}^{(t)} = 1\right))$$

Equation 7. Prediction in HOT-DINA

3.2.3 Updating estimated skills

We update the estimates of latent states $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$ after observing actual student performance. The estimate of knowing a skill or a subskill should increase if the student performed correctly at the step. It is easy to update a skill by using Bayes' rule, as shown in Equation 8. The posterior $P(\eta_{nj}^{(t)} = 1 | Y_{nj}^{(t)} = 1)$ should be higher than $P(\eta_{nj}^{(t)} = 1)$ if and only if $(1\text{-}slip_j) > guess_j$.

$$P\left(\eta_{nj}^{(t)} = 1 \middle| Y_{nj}^{(t)} = 1\right) = \frac{P\left(Y_{nj}^{(t)} = 1 \middle| \eta_{nj}^{(t)} = 1\right) P\left(\eta_{nj}^{(t)} = 1\right)}{P\left(Y_{nj}^{(t)} = 1\right)}$$

$$= \frac{(1 - slip_j)\, P\left(\eta_{nj}^{(t)} = 1\right)}{(1 - slip_j)\, P\left(\eta_{nj}^{(t)} = 1\right) + guess_j \left(1 - P\left(\eta_{nj}^{(t)} = 1\right)\right)}$$

Equation 8. Bayes' rule to update η in HOT-DINA

Although we could update HOT-DINA by assigning full responsibility to each skill, it would be interesting to update the weakest (or say hardest) skill since HOT-DINA fits the parameter 'difficulty' for each skill. Thus, we update the skill that is the hardest among all the required skills in a step:

$$P\left(\eta_{nj}^{(t)} = 1 \middle| Y_{nj}^{(t)} = 1\right) = P\left(\alpha_{nk'}^{(t)} = 1 | Y_{nj}^{(t)} = 1\right) \prod_{k \neq k'} P(\alpha_{nk}^{(t)} = 1)$$

for $k = \arg\max_{k:\, Q_{jk} = 1} b_k$.

Equation 9. Update the hardest skill in HOT-DINA

In short, we extend HO-KT to the HOT-DINA higher order temporal model, which traces multiple skills. We use the MCMC algorithm to estimate the parameters, and update the estimates of a student knowing a skill given observed student performance. How well does the HOT-DINA model work? To

evaluate it, we performed a simulation study. Section 4 now describes the study and reports its results.

4. Simulation Study

To study the behavior of HOT-DINA, we generated synthetic training data for it according to the priors and conditional distributions defined in Section 3.2.1. Section 4.1 describes the synthetic data. One purpose of this experiment was to test how accurately MCMC can recover the parameters of HOT-DINA, as Section 4.2 reports. It is important not only to test how well a method works, but to analyze when and why. Thus another purpose was to determine how many students and observations are needed to estimate the difficulty and discriminability of a given number of skills, as Section 4.3 explains.

4.1 Synthetic Data

We use the following procedure to generate the synthetic data, with all the variables as defined in Section 3.2:

1. We chose K = 4 and J = 14, which results in a 14 × 4 **Q** matrix. The **Q** matrix, as shown below, indicates that we generate the skills by combining all the possible skills.

$$\mathbf{Q}^{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

2. We randomly generated θ_n from *Normal (0,1)* for *n = 1,..,N.*

3. We chose ***a***, ***b*** and ***l*** as shown in Table 3.

Table 3. True value of skill-specific discrimination, difficulty and learning rate in synthetic data simulation

k	1	2	3	4
a	1.5	1.2	1.9	1.0
b	-0.95	1.42	-0.66	0.50
learn	0.8	0.6	0.5	0.3

4. We randomly generated ***g*** and ***1-s*** from *Unif(0,0.4)* and *Unif (0.6,1)* respectively, as shown in Table 4.

Table 4. True value of skill-specific guess and not slip parameters in synthetic data simulation

j	1	2	3	4	5	6	7
guess	0.35	0.40	0.13	0.15	0.29	0.39	0.10
1-slip	0.67	0.66	0.67	0.90	0.65	0.60	0.61
j	8	9	10	11	12	13	14
guess	0.40	0.15	0.16	0.38	0.11	0.26	0.35
1-slip	0.81	0.74	0.76	0.73	0.83	0.89	0.85

5. We chose N = 100, T = 100, randomly picked one skill at each step, and simulated sequential data with size of 10,000.

4.2 Results

We used OpenBUGS [24] to implement the MCMC algorithm of HOT-DINA. We chose 5 chains starting at different initial points. We monitored the estimates of skill discrimination $\widehat{\boldsymbol{a}}$ and difficulty $\widehat{\boldsymbol{b}}$ to check their convergence, when all the chains appear to be overlapping each other. As a result, we ran the simulation for 10,000 iterations with a burn-in of 3000.

Table 5 reports the sample means and their 95% confidence interval for parameter estimates $\widehat{\boldsymbol{a}}$, $\widehat{\boldsymbol{b}}$ and $\widehat{\boldsymbol{learn}}$ respectively. We also report the Monte Carlo error (MC error) and sample standard deviation (s.d.) to assess the accuracy of the posterior estimates for each parameter. MC error, which is an estimate of the difference between the estimated posterior mean (i.e. the sample mean) and the true posterior mean, should be less than 5% of the s.d. in order to obtain an accurate posterior estimate.

Table 5. Estimates of skill-specific discrimination, difficulty, and learning rate (N = 100, T = 100, K = 4, J = 14)

k	***a***	$\widehat{\boldsymbol{a}}$ (95% C.I.)	s.d.	MC_error
1	1.50	1.33 (0.36, 2.43)	0.65	0.03216
2	1.20	1.23 (0.12, 2.43)	0.72	0.03561
3	1.90	1.85 (0.22, 2.73)	0.64	0.03146
4	1.00	0.98 (0.19, 2.12)	0.58	0.02870
k	***b***	$\widehat{\boldsymbol{b}}$ (95% C.I.)	s.d.	MC_error
1	-0.95	-0.95 (-2.15, -0.04)	0.50	0.02339
2	1.42	1.51(0.90, 2.21)	0.45	0.01936
3	-0.66	-0.69 (-1.81, -0.63)	0.42	0.01990
4	0.5	0.5 (0.05,1.18)	0.38	0.01691
k	***learn***	$\widehat{\boldsymbol{learn}}$ (95% C.I.)	s.d.	MC_error
1	0.8	0.81 (0.48, 0.99)	0.13	0.006599
2	0.6	0.60 (0.52, 0.70)	0.05	0.002132
3	0.5	0.57 (0.38, 0.84)	0.11	0.005432
4	0.3	0.29 (0.25, 0.33)	0.02	7.79E-04

We calculated Root Mean Squared Error (RMSE) of the estimates of the continuous variables $\widehat{\boldsymbol{guess}}$, $\widehat{\boldsymbol{1-slip}}$, and $\widehat{\boldsymbol{\theta}}$. We report the accuracy of recovering the true value of the latent binary variable $\boldsymbol{\alpha}$ in Table 6.

Table 6. Estimation RMSE of skill-specific guess, not slip, and student specific proficiency; Prediction accuracy of a student mastering a subskill (N = 100, T = 100, K = 4, J = 14)

	$\widehat{\boldsymbol{guess}}$	$\widehat{\boldsymbol{1-slip}}$	$\widehat{\boldsymbol{\theta}}$
RMSE	0.0103	0.0196	0.9183
	$\widehat{\boldsymbol{\alpha}}$		
Accuracy	99.38%		

From the results, we can see that the MCMC algorithm accurately recovered the parameters we used in generating the synthetic data for HOT-DINA. In addition to seeing how accurately it can estimate the parameters, we are also interested in finding out how many observations would be sufficient for the training algorithm to recover the hidden variables. Therefore, we conducted the study we now describe in Section 4.3.

4.3 Study Design

HOT-DINA requires data from enough students to rate the difficulty and discriminability of each skill, and data on enough skills to estimate the proficiency of each student. So we fixed the number of skills at K = 4, and varied the number of students N or the number of steps observed from each student T, to discover how many observations would be sufficient to estimate the parameters. In particular, we evaluated each model on how accurately it estimated the latent binary state $\boldsymbol{\alpha}$, which indicates if a student masters a skill. We generated the data by using the same parameters as in Section 4.1. Besides the general HOT-DINA model that accounts for multiple skills, we also studied the single-skill model by shrinking the number of skills J to equal K, and set **Q** as an identity matrix. Thus we specified the HOT-DINA model to be a HO-KT model alternatively.

We increased N, the number of students, from 10 to 1000, and T, the number of observations per student, from 5 to 100. Table

7 and Table 8 respectively show the accuracy of estimating the latent state α in HO-KT and HOT-DINA. Both tables show a trend of increasing accuracy when N or T increases (though at the cost of longer training time, roughly $O(N^2 \times T)$).

Table 7. Accuracy of estimating the latent binary states α with different N and T (K = J = 4)

N \ T	5	10	20	50	100
10	71.01%	80.81%	83.01%	93.11%	96.16%
20	72.32%	82.74%	86.52%	94.06%	97.33%
50	73.58%	83.79%	87.34%	95.27%	98.90%
100	77.55%	84.43%	88.08%	95.81%	**99.41%**
200	76.52%	84.02%	89.48%	97.26%	NA
500	78.13%	84.34%	92.50%	NA	NA
1000	80.10%	84.59%	NA	NA	NA

Due to the lack of sampling ability of OpenBUGS for high dimensional dynamic models, we have no available scores to show for N×T bigger than 10,000. We can see that the multiple skill model predicts better than the single-skill model because the average number of observations per skill in the former one is larger than the latter. As observed in both tables, it is more efficient to increase T, than N, to get a better estimate. Both of the models reach the best prediction accuracy score (> 99%) when N = 100 and T = 100. In order to obtain an accuracy > 90% for K = 4 skills, the least amount of data we need for HO-KT is N = 10 with T ≈ 50 observations as shown in Table 7, for HOT-DINA is N = 10 with T > 20 observations, as shown in Table 8.

Table 8. Accuracy of estimating the latent binary states α with different N and T (K = 4, J = 14)

N \ T	5	10	20	50	100
10	72.07%	75.57%	91.14%	96.90%	98.10%
20	74.32%	83.60%	91.56%	97.46%	98.53%
50	76.55%	84.71%	92.62%	97.52%	98.98%
100	77.80%	86.82%	93.83%	97.67%	**99.82%**
200	79.92%	88.78%	94.26%	99.41%	NA
500	82.15%	89.95%	98.61%	NA	NA
1000	83.58%	92.34%	NA	NA	NA

Next we apply the proposed model to real data logged by an algebra tutor. We evaluate the model fit and compare it against two baselines.

5. Evaluation on Real Data

We apply HOT-DINA to a real dataset from the Algebra Cognitive Tutor® [25]. Because of limited time, we chose a subset of the data, by crossing out the "isolated" algebra tutor steps. An "isolated" step here means a step that requires one skill all its own. We grouped the remaining steps that require the same multiple skills into one skill, resulting in J = 15 distinct skills that require K = 12 subskills. Following the study design in Section 4.3, we randomly chose N = 50 students with T = 100 in order to obtain enough data for the MCMC estimation.

Table 9. Data split of the Algebra Tutor data: training on I and IV, and testing on II and III

	Skill group A	Skill group B
Student group A	I	II
Student group B	III	IV

We split the 50 students into two groups of 25, and split the 15 skills into two groups of 8 and 7. As shown in Table 9, we combine data from I (student-group-A practicing on skill-group-A) and IV (student-group-B practicing on skill-group-B) to obtain the training data. Accordingly, we combined the data from II and III to obtain the test data. As a benefit of the data split, we are able to test the models on unseen students for the same group of skills, and also test on the unseen skills for the same group of students.

We compared HOT-DINA with the conjunctive minimum KT model [11] since it showed the best prediction accuracy among all the previous KT based methods [4]. It fits KT parameters by blaming each skill that is required at a step, predicts student's performance by the weakest skill, and updates only the weakest skill. Accordingly, we updated the most difficult skill in HOT-DINA as discussed in Section 3.2.3. As two baseline models, we fit per-skill KT and per-student KT. Comparing HOT-DINA with these two baselines also allows us to discuss some more interesting research questions later in this section.

Table 10 and Table 11 respectively show the models' prediction accuracy and log-likelihood on the test data. We report the majority class because of the unbalanced data. HOT-DINA beat the two baselines in predicting the student performance, and also obtained the maximum log-likelihood on the test data. The per-student KT model obtained the worst scores on both measures. It predicted student performance almost as poorly as majority class because it misclassified almost all the data in the minority class.

Table 10. Comparison of prediction accuracy on real test data

	Overall Accuracy	Accuracy on Correct Steps	Accuracy on Incorrect Steps
HOT-DINA	**82.48%**	96.63%	27.27%
Per-skill KT	80.87%	94.02%	29.60%
Per-student KT	79.63%	99.74%	1.20%
Majority class	79.60%	100.00%	0.00%

Table 11. Comparison of log-likelihood on real test data

	Log-likelihood
HOT-DINA	**-2021.04**
Per-skill KT	-2075.67
Per-student KT	-2464.74

We are also interested in three other hypotheses comparing HOT-DINA with KT. We describe them, test them, and show the results as follows.

1. HOT-DINA should predict early steps more accurately than KT since its estimate of *knew* reflects both skill difficulty and student proficiency, not just one or the other. In fact HOT-DINA beat KT throughout, as Figure 2 shows.

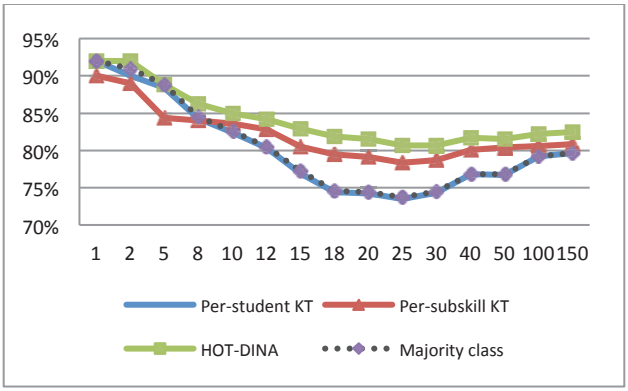


Figure 2. Accuracy on student's 1st, 2nd, 3rd, … test steps

2. HOT-DINA should beat KT on sparsely trained skills thanks to student proficiency estimates based on other skills. As Figure 3 shows, HOT-DINA tied or beat KT throughout.

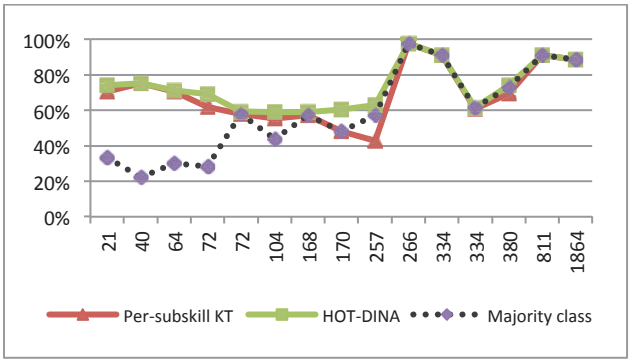


Figure 3. Skills sorted by amount of training data

3. HOT-DINA should beat KT on sparsely trained students thanks to skill difficulty and discriminability estimates based on other students. As Figure 4 shows, HOT-DINA beat KT throughout.

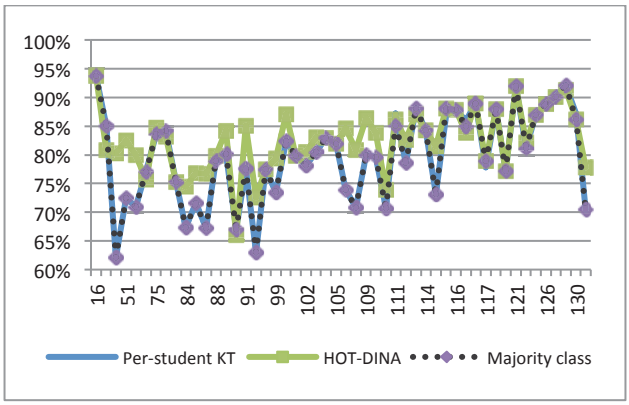


Figure 4. Students sorted by amount of training data

Thus, HOT-DINA outperformed the two baselines in model fit. It also beat them as specified by the three hypotheses above.

6. Contributions, limitations, future work

In this paper we make several contributions. We defined a 5-dimensional framework for student models. We showed how numerous student models fit into it. We described the new combination of IRT, KT, and DINA it suggests in the form of HOT-DINA. We specified how to train HOT-DINA by using MCMC, how to test it by predicting student performance, and how to update estimated skills based on observed performance.

HOT-DINA uses IRT to estimate *knew* based on student proficiency and skill difficulty. Thus it does not need training data on every <student, skill> pair, since it can estimate student proficiency based on other skills, and skill difficulty and discriminability based on other students. Likewise, it should estimate *knew* more accurately than KT for skills and students with sparse training data. HOT-DINA uses KT to model learning over time, and DINA to model combination of multiple skills underlying observed steps (unlike conventional KT and with fewer parameters than CKT [10] or LR-DBN [12]).

Tracing multiple skills underlying an observed step requires allocating responsibility among them for its success or failure. DINA simply conjoins them, a common method but inferior to others. Future work includes using the best known method [4], which we didn't use here because the logistic regression it performs is non-trivial to integrate with MCMC.

We evaluated HOT-DINA on synthetic and real data, not only showing that it predicts student performance better than previous methods, but analyzing when and why.

We reported a simulation study to test if training could recover model parameters, and to determine the amount of data needed. HOT-DINA requires data on enough students and skills to estimate their proficiency and difficulty, respectively. We explored how its accuracy varies with the number of test steps and the amount of training data per student and per skill. These analyses were correlational, based on variations that happened to occur in the training data. Future work should invest in the computation required to vary the amount of training data to establish its true causal effect on accuracy.

Evaluation on real data from an algebra tutor showed that HOT-DINA achieved higher predictive accuracy and log likelihood than KT with parameters fit per student or per skill. This evaluation was limited to a single data set and two baselines (not counting majority class). Future work should compare HOT-DINA to other methods – notably the Student Skill model [8], which is similar in spirit – and on data from other tutors.

We assumed that student proficiency is one-dimensional. Future work can test if k dimensions capture enough additional variance to make it worthwhile to fit k times as many parameters.

Finally, our choice of 5 dimensions is useful but limiting. Additional dimensions may provide useful finer-grained insights into the models covered by the current framework, and expand it to encompass other types of student models, e.g. where the cognitive model is unknown and must be discovered [18, 19].

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation through Grants 1124240 and 1121873 to Carnegie Mellon University. The opinions expressed are those of the authors and do not necessarily represent the views of the National Science Foundation or U.S. government. We thank Ken Koedinger for his algebra tutor data.

REFERENCES

[1] Zwicky, F. *Discovery, Invention, Research - Through the Morphological Approach*. 1969, Toronto: The Macmillian Company.

[2] Corbett, A. and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 1995. *4*: p. 253-278.

[3] Mostow, J., Y. Xu, and M. Munna. Desperately Seeking Subscripts: Towards Automated Model Parameterization. *Proceedings of the 4th International Conference on Educational Data Mining*, 283-287. 2011. Eindhoven, Netherlands.

[4] Xu, Y. and J. Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best? [Best Student Paper Award]. *Proceedings of the Fifth International Conference on Educational Data Mining*, 41-48. 2012. Chania, Crete, Greece.

[5] Hambleton, R.K., H. Swaminathan, and H.J. Rogers. *Fundamentals of Item Response Theory*. Measurement Methods for the Social Science. 1991, Newbury Park, CA: Sage Press.

[6] Pavlik Jr., P.I., H. Cen, and K.R. Koedinger. Performance factors analysis - a new alternative to knowledge tracing. *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED09)*, 531-538. 2009.

[7] Pardos, Z. and N. Heffernan. Modeling individualization in a Bayesian networks implementation of knowledge tracing. *Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, 255-266. 2010. Big Island, Hawaii.

[8] Wang, Y. and N.T. Heffernan. The student skill model. *Intelligent Tutoring Systems - 11th International Conference*, 399-404. 2012. Chania, Crete, Greece. Springer.

[9] Cen, H., K.R. Koedinger, and B. Junker. Comparing Two IRT Models for Conjunctive Skills. *Ninth International Conference on Intelligent Tutoring Systems*, 796-798. 2008. Montreal.

[10] Koedinger, K.R., P.I. Pavlik, J. Stamper, T. Nixon, and S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. In *Proceedings of the 4th International Conference on Educational Data Mining*. 2011: Eindhoven, NL, p. 91-100.

[11] Gong, Y., J.E. Beck, and N.T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, 35-44. 2010. Pittsburgh, PA. Springer Berlin / Heidelberg.

[12] Xu, Y. and J. Mostow. Using logistic regression to trace multiple subskills in a dynamic Bayes net. *Proceedings of the 4th International Conference on Educational Data Mining*, 241-245. 2011. Eindhoven, Netherlands.

[13] de la Torre, J. and J.A. Douglas. Higher-order latent trait models for cognitive diagnosis. *Psychometrika* 2004. *69*(3): p. 333-353.

[14] Junker, B. and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 2001. *25*(3): p. 258-272.

[15] de la Torre, J. DINA Model and Parameter Estimation: A Didactic *Journal of Educational and Behavioral Statistics*, 2009. *34*(1): p. 115-130.

[16] Maris, E. Estimating multiple classification latent class models. *Psychometrika*, 1999. *64*(2): p. 197–212.

[17] Xu, Y. and J. Mostow. Using item response theory to refine knowledge tracing. In *Proceedings of the 6th International Conference on Educational Data Mining*, S.K. D'Mello, R.A. Calvo, and A. Olney, Editors. 2013, International Educational Data Mining Society: Memphis, TN, p. 356-357.

[18] González-Brenes, J.P. and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Proceedings of the 6th International Conference on Educational Data Mining*, S.K. D'Mello, R.A. Calvo, and A. Olney, Editors. 2013, International Educational Data Mining Society: Memphis, TN, p. 236-239.

[19] González-Brenes, J.P. and J. Mostow. Dynamic cognitive tracing: towards unified discovery of student and cognitive models. *Proceedings of the Fifth International Conference on Educational Data Mining* 2012. Chania, Crete, Greece.

[20] Cen, H., K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 164-175. 2006. Jhongli, Taiwan.

[21] Fischer, G.H. The linear logistic test model. In G.H. Fischer and I.W. Molenaar, Editors, *Rasch Models: Foundations, Recent Developments, and Applications*, 131-155. Springer: New York, 1995.

[22] Wang, X., J.O. Berger, and D.S. Burdick. Bayesian analysis of dynamic item response models in educational testing. *Annals of Applied Statistics*, 2013. *7*(1): p. 126-153.

[23] Studer, C. *Incorporating Learning Over Time into the Cognitive Assessment Framework.* Unpublished PhD, Carnegie Mellon University, Pittsburgh, PA, 2012.

[24] Lunn, D., D. Spiegelhalter, A. Thomas, and N. Best. The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 2009. *28*: p. 3049–306.

[25] Koedinger, K.R., R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: the PSLC DataShop. In C. Romero, et al., Editors, *Handbook of Educational Data Mining*, 43-55. CRC Press: Boca Raton, FL, 2010.

The Sequence of Action Model: Leveraging the Sequence of Attempts and Hints

Linglong Zhu
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA
lzhu@wpi.edu

Yutao Wang
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA
yutaowang@wpi.edu

Neil T. Heffernan
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA
nth@wpi.edu

ABSTRACT

Intelligent Tutoring Systems (ITS) have been proven to be efficient providing student assistance and assessing their performance when they do their homework. Researchers have analyzed how students' knowledge grows and predict their performance from within intelligent tutoring systems. Most of them focus on using correctness of the previous question or the number of hints and attempts students need to predict their future performance, but ignore the sequence of hints and attempts. In this research work, we build a Sequence of Actions (SOA) model taking advantage of the sequence of hints and attempts a student needed for the previous question to predict students' performance. A two step modeling methodology is put forward in the work and is a combination of Tabling method and the Logistic Regression. We compared SOA with Knowledge Tracing (KT) and Assistance Model (AM) and combinations of SOA/AM and KT. The experimental results showed that the Sequence of Action model has reliably better predictive accuracy than KT and AM and its performance of prediction is improved after combining with KT.

Keywords

Knowledge Tracing, Educational Data Mining, Student Modeling, Sequence of Action, Assistance Model, Ensemble.

1. INTRODUCTION

One of the student modeling tasks is to trace the student's knowledge by using student's performance. Corbett and Anderson (1995) put forward the well-known Knowledge Tracing (KT) based on their observation that the students' knowledge is not fixed, but is assumed to be increasing. KT model makes use of Bayesian network to model students' learning process and predicate their performance.

A variety of extensions of KT model are put forward in recent years. Baker, Corbett, and Aleven (2008) build a contextual guess and slip model based on KT that provides more accurate and reliable student modeling than KT. Pardos and Heffernan extends KT four parameters model to support individualization and skill specific parameters and get better prediction of students' performance. Qiu and Qi et al. find that forgetting is a more likely cognitive explanation for the over prediction of KT when considering the time students take to finish their tasks.

Alternative methods to KT model have been developed. For example, in order to generate adaptive instructions for students, Pavlik Jr., Cen, and Koedinger (2009) put forward the Performance Factor Analysis (PFA) model that can make predictions for individual students with individual skills. Gong, Beck, and Heffernan (2010) compared KT with PFA using multiple model fitting procedures and showed that there are no real differences in predictive accuracy between these two models.

However, little attention is paid to the data generated when students interact with computer tutors. Shih, Koedinger, and Scheines (2010) utilize Hidden Markov Model clustering to discover different strategies students used while working on a ITS and predict learning outcomes based on these strategies. Their work is based on a dataset that consists of a series of transactions and each transaction is a <Student, Step, Action, Duration> tuple. This model takes into account both students' action, attempt or help request, and action duration. The experimental results of their Stepwise-HMM-Cluster model shows that persistent attempts lead to better performance than hint-scaffolding strategy. Some papers have shown the value of using the raw number of attempts and hints. In fact, the National Educational Technology Plan cited Feng, Heffernan, and Koedinger's work (2006) and the User Modeling community gave it an award for best paper for showing that the raw number of hints and attempts is informative in predicting state test scores. Wang and Heffernan (2011) built an Assistance Model (AM) and generated a performance table based on students' behavior of doing the previous question. Hawkins et al.(2013) extended AM by looking at students' behavior for the two previous questions.

These educational data mining models that utilize the number of assistance students request and the number of attempts they make to predict students' performance have ignored the sequencing of students' interaction with ITS. Consider a thought experiment. Suppose you know that Bob Smith asked for one of the three hints and makes one wrong answer before eventually getting the question correct. What if someone told you that Bob first made an attempt then had to ask for a hint compared to the first requesting a hint and then making a wrong attempt. Would this information (whether he started with an attempt or a hint) add value to your ability to predict whether Bob will get the next question correct? We suspected that a student who first makes an attempt tends to learn by himself and has higher probability to master the knowledge and answer the next same question correct.

In our previous work, we showed a Sequence of Action (SOA) model that made use of information about the action sequence of attempts and hints for a student in previous question better predicted the correctness of a current question.. We reported experimental results of an improvement upon the KT model. However, we later found a mistake in that experiment. So this paper serves as a correction of the previous results and as a formal presentation of the SOA model to the community. We present the SOA model and compare it to the KT model and the Assistance model, as well as the combined models to see if knowing sequence of action information does improve upon a

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

standard Knowledge Tracing model, or even upon knowing number of hints and number of attempts alone.

The raw data and experiment result is available online: https://sites.google.com/site/assistmentsdata/projects/zhu2014.

1.1 The Tutoring System and Dataset

The data we used originated from the ASSISTments platform, an online tutoring system for K12 students that gives immediate feedback to teachers, students, and parents. The ASSISTments gives tutorial assistance if a student makes a wrong attempt or asks for help. Figure 1 shows an example of a hint, which is one type of assistance. Other types of assistance include scaffolding questions and context-sensitive feedback messages, known as "buggy messages."

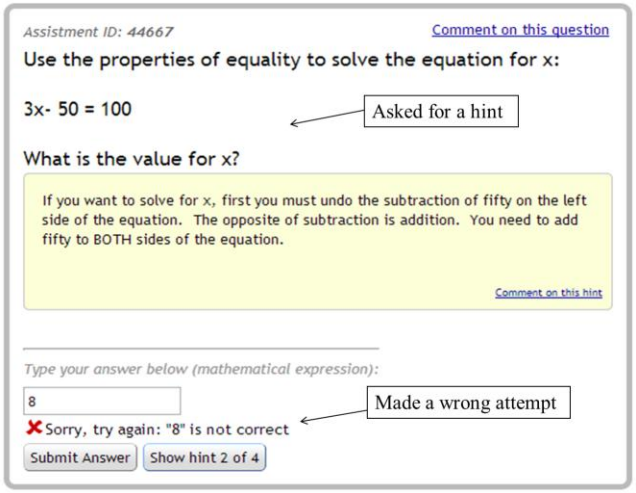


Figure 1. Assistance in ASSISTments. Which is first: asking for a hint or make an attempt?

Figure 1 shows a student who asked for a hint (shown in yellow and also indicated by the button says "Show hint 2 of 4"), but it also shows that the student typed in eight and got feedback that this was wrong. Though Figure 1 shows the number of hints and attempts, interestingly you cannot tell whether the student asked a hint first or made an attempt first. This paper's argument is that information is very important.

ASSISTments records all the details about how a student does his or her homework and tests from which scientists can get valuable material to investigate students' behavior and their learning process. These records include the start time and end time of a problem, the time interval between an attempt, if he or she asks for a hint, the number of attempts a student makes, the number of hints a student asks for, as well as the answer and result for each attempt a student makes.

Figure 2 shows an example of a detailed sequence of action recorded by the system. The row in blue means that the answer is correct, the row in red means that the answer is wrong, and the row in orange means the student asked for a hint. We can see that this student answered correctly on his first attempt for the first problem PRAQM5U. The sequence of action is 'a' ('a' represents an attempt). For the second problem PRAQM2W, he asked three hints continuously before making the correct answer. The sequence of action is 'hhha' ('h' represents a hint). For the third problem PRAQM2F, he alternatively asked for hints and made attempts, and the sequence of action is 'hahaha'. For the last problem PRAQZPN, he made one wrong attempt before making the correct answer and its action sequencing is 'aa.'

Assignment: 18 - Equivalent Fractions (2) 4.NF.A.1

Time	Action	Object ID / Input text
Wed Feb 29 13:00:49 -0500 2012	Started a problem	PRAQM5U
1 mins 18 secs	Answered	269/17
Wed Feb 29 13:02:18 -0500 2012	Started a problem	PRAQM2W
5 mins 27 secs	Asked for a hint	
3 mins 8 secs	Asked for a hint	
0 mins 26 secs	Asked for a hint	
0 mins 9 secs	Answered	11 6/17
Wed Feb 29 13:11:41 -0500 2012	Started a problem	PRAQM2F
0 mins 5 secs	Asked for a hint	
1 mins 32 secs	Answered	3.6
0 mins 2 secs	Asked for a hint	
0 mins 24 secs	Answered	3/2
0 mins 2 secs	Asked for a hint	
0 mins 7 secs	Answered	3 2/5
Wed Feb 29 13:12:30 -0500 2012	Started a problem	PRAQZPN
0 mins 6 secs	Answered	76000
0 mins 15 secs	Answered	80000

Figure 2. Students' action records in ASSISTments

We used data from one Mastery Learning class. Mastery Learning is a strategy that requires students to continually work on a problem set until they have achieved a preset criterion (typically three consecutive correct answers). Questions in each problem set are generated randomly from several templates and there is no problem-selection algorithm used to choose the next question.

Sixty-six 12-14 year-old, 8th grade students participated in these classes and generated 34,973 problem logs. We only used data from a problem set for a given student if they had reached the mastery criterion. This data was collected in a suburban middle school in central Massachusetts. Students worked on these problems in a special "math lab" period, which was held in addition to their normal math class.

If a problem only has one hint, the hint is the answer of the problem and is called the bottom hint. After a student asks for a bottom hint, any other attempt is meaningless because he or she already knows the answer. In the experiment, we only consider the problem logs that have at least two hints. And the answer will be marked as incorrect if students ask for a hint or the first attempt is incorrect. Moreover, we excluded such problem logs where: 1) students quit the system immediately after they saw the question and the action logs were blank ,or 2) after they requested hints, but did not make any attempts and no answer was recorded.

Here we only consider the question pairs that have the same skill and skills having only one question were removed because they do not help in predicting. Questions of the same skills were sorted by start time in ASSISTments. We split equally 66 students into six groups, 11 students in each, to run 6-fold cross validation. We trained the SOA model and the KT model on the data from five of the groups and then computed the prediction accuracy on the sixth group. We did this for all six groups.

2. INDIVIDUAL MODELS

2.1 KT

Knowledge Tracing (KT) is one of the most common methods that are used to model the process of student's knowledge gaining and to predict students' performance. The KT models is an Hidden Markov Model (HMM) with a hidden node (student

knowledge node) and an observed node (student performance node). It assumes that a skill has four parameters; two knowledge parameters and two performance parameters. The two knowledge parameters are: prior and learn. The prior knowledge parameter is the probability that a particular skill was known by the student before interacting with the tutor. The learn parameter is the probability that a student transits from the unlearned state to the learned state after each learning opportunity, i.e., after see a question. The two performance parameters are: guess and slip. Guess is the probability that a student will guess the answer correctly even if the skill associated with the question is in the unlearned state. Slip is the probability that a student will answer incorrectly even if he or she has mastered the skill for that question.

The goal of KT is to estimate the student knowledge from his or her observed actions. At each successive opportunity to apply a skill, KT updates its estimated probability that the student knows the skill, based on the skill-specific learning and performance parameters and the observed student performance (evidence). It is able to capture the temporal nature of data produced where student knowledge is changing over time. KT provides both the ability to predict future student response values, as well as providing the different states of student knowledge. For this reason, KT provides insight that makes it useful beyond the scope of simple response prediction.

2.2 Assistance Model

Motivated by the intuition that students who need more assistance have lower probability possessing the knowledge, Wang and Heffernan (2011) built a purely data driven "Assistance" model to discover the relationship between assistance information and students' knowledge.

A parameter table was built in which rows represent the number of attempts a student required in the previous question and columns represent the number of hints the student asked for. Each cell contains the probability that the student will answer the current question correctly. The attempts are separated into three bins: one attempt, small number of attempts (2-5 times), and large numbers of attempts (more than five attempts). Hints are separated into four bins: no hint, small number of hints (1, 50%], large number of hints [50%, 100%), and all hints where students for all hints. Table 1 shows the parameter table gained from our dataset. As with Wang and Heffernan's experimental results, the parameter table confirms that students requiring more assistance to solve a problem probably have less corresponding knowledge.

Table 1. Assistance Model parameter table, average across six folds

	attempt= 1	**0<attempt<6**	**attempt>=6**
hint_percent = 0	0.8410	0.7963	0.7808
0<hint_percent<=.5	0.6286	0.6933	0.6741
.5<hint_percent<1	0.4494	0.6290	0.6522
hint_percent = 1	0.4293	0.6147	0.6218

2.3 The Sequence of Action Model

The Sequence of Action (SOA) model we present takes advantage of the order information about how students make attempts and ask for hints. Different students have different sequences of actions. Some students answered correctly only after one attempt and some students kept trying many times. Some students asked for hints and made attempts alternatively and we believe they were learning by themselves. In the data, there are 217 different sequences of actions. Intuitively, students' actions reflect their study attitude and this determines their performance. Based on the assumption that students who make more attempts tend to master knowledge better than students who ask for more hints, we divided them into five categories or bins: (1) One Attempt: the student correctly answered the question after one attempt; (2) All Attempts: the student made many attempts before finally getting the question correct; (3) All Hints: the student only asked for hints without any attempts at all; (4) Alternative, Attempt First: the students asked for hints and made attempts alternatively and made an attempt at first; and (5) Alternative, Hint First: the students asked for hint and made attempts alternatively and asked for a hint first. Table 2 shows the division and some examples of the action sequences in each category.

Table 2. Sequence of Action Category and Examples

Sequence of Action Category/ Bin Name	**Examples**
One Attempt/Bin 'a'	a
All Attempts/Bin 'a+'	aa, aaa, …, aaaaaaaaaaa
All Hints/Bin 'h+'	ha, hha,…, hhhhhhha
Alternative, Attempt First/Bin 'a-mix'	aha, aahaaha,…, aahhhhaaa
Alternative, Hint First/Bin 'h-mix'	haa, haha,…, hhhhaha

Notice that each sequence ends with an attempt because in ASSISTments, a student cannot continue to next question unless he or she fills in the right answer of the current problem. In Table 2, 'a' stands for answer and 'h' stands for hint. An action sequence "ahha" means that a student makes an attempt and then asks for two hints before he or she types the correct answer and moves on to the next question.

2.3.1 Sequence of Action Tabling

After dividing all of sequence of actions into five categories, we use a Tabling method, which gets the next percent correct directly from the training data. For each fold, one table is generated by the tabling method by counting the number of total appearance and the number of next correct of each bin. After counting, a next correct percent is calculated by dividing *Next Correct Count* by *Total Count* of Bin.

Table 2. Next correct percent table of training group of fold 1

Bin Name	**Total Count**	**Next Correct Count**	**Next Correct Percent**
'a'	22964	19157	0.834
'a+'	3538	2690	0.760
'h+'	335	172	0.513
'a-mix'	2030	1318	0.649
'h-mix'	72	37	0.513

Table 3 shows the table computed for fold 1. Tables for other folds are similar. From Table 3, we can see that the percent of next-question-correct is highest among students only using one attempt since they master the skill the best. They can correctly

answer the next question with the same skill. For students in 'a+' bin, they are more self-learning oriented, they try to learn the skill by making attempts over and over again. So they get the second highest next-question-correct percent. But for students in the 'h+' category, they do the homework only relying on the hints. It is reasonable that they don't master the skill well or they don't even want to learn, so their next-question-correct percent is very low.

The alternative sequence of action reflects students' learning process. Intuitively, these students have positive attitudes for study. They want to get some information from the hint based on which they try to solve the next problem. But the results for the two alternative categories are very interesting. Though students in these two categories alternatively ask for hints and make attempts, the first action somewhat decides their learning altitude and final results. For students who make an attempt first, if they get the question wrong, they try to learn it by asking for hints. But for students who ask for a hint first, they seem to have less confidence in their knowledge. Although they also make some attempts, from the statistics of action sequence, they tend to ask for more hints than making attempts. The shortage of knowledge or the negative study attitude makes their performance as bad as the students asking exclusively for hints first.

2.3.2 Logistic Regression

In this section, we are going to introduce the second part of the SOA model that makes use of a logistic regression model and information we get from the first part of SOA, i.e., tabling method.

Even though the next correct percentage we get from the tabling method indicates that the action of sequence can reflect the trend of next correct percentage, the table is very rough and is not intelligent enough to be used to predict students' performance. However, we can use it as a feature in our logistic regression prediction model.

The dependent variable ***Next Correct*** of the logistic regression model has two states: correct and incorrect. The independent variables are ***Skill_ID*** and ***Credit*** (the next correct percentage generated by the tabling method). ***Skill_ID*** was treated as a categorical factor, while ***Credit*** was treated as a continuous factor. There are totally 51 skills of the data. As mentioned in before, there are six folds and each fold has their own next correct percentage table.

We used Binary Logistic Regression in SPSS to train the model. Logistic coefficients are fitted through Expectation Maximization of at most 20 steps. Parts of coefficients of the first fold are shown in Table 4.

Table 4. Coefficients of logistic regression model of fold 1

Parameters	Value
β_0(Intercept)	-1.679
$\beta_{1,0}$(skill_id 16)	0.322
$\beta_{1,1}$(skill_id 17)	-0.007
$\beta_{1,2}$(skill_id 24)	20.168
…….	……
$\beta_{1,50}$(skill_id 371)	0.470
β_2(Credit)	3.286

3. MODEL COMBINATION

Since the SOA model uses completely different information from KT, we expected a potential improvement from combing SOA results with the predictions from KT. We combined models using two different methods.

The first method was simply average the SOA and KT predictions. Presumably, if a group of models have high accuracies and uncorrelated errors, we can get lower error by averaging them. To compare with the combination of AM model and KT model, we also computed the average of these two models.

The second method was a linear regression model with student performance as the dependent variable. This method takes into account the fact that different models' predictions may have different weight in the final prediction. If one of the models is more useful than the other, this method will allow us to learn which model should be weighted more heavily. SPSS was used to train linear regression models. The function for KT and AM is:

-0.322+0.639*AM_prediction+0.769*KT_prediction;

The function for KT and SOA is:

-0. 004+0. 687*SOA_prediction+0. 321*KT_prediction;

We did not combine AM and SOA, because both of them use information about hints and attempts. From the functions, we can tell that SOA weights heavier than KT, which indicates that SOA is more useful than KT in making a prediction.

4. EXPERIMENTAL RESULTS

4.1 Compare AM, SOA and KT

To evaluate how well each of the individual models (SOA, AM, KT) and the combined models fit the data, we used three metrics to examine the predictive performance on the unseen test set: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Area Under ROC Curve (AUC). Lower values for MAE and RMSE and higher values for AUC indicate better model fit.

Table 5. Prediction accuracy of KT, SOA, AM and Ensemble

	MAE	RMSE	AUC
AM	0.3007	0.3844	0.5795
SOA	0.2871	0.3767	0.6786
KT	0.2939	0.3790	0.6735
LR(AM, KT)	**0.2874**	0.3759	0.6824
LR(SOA, KT)	0.2878	0.3762	0.6813
AVG(SOA, KT)	0.2876	**0.3757**	**0.6836**

Table 5 shows values of the three metrics from a six-fold across validation, which are calculated by averaging corresponding numbers obtained from each validation. As with Wang and Heffernan's results (Wang & Heffernan, 2011), the performance of linear regression combination of AM and KT, called as LR(AM, KT) is better than AM itself, which indicates information about the number of hints and attempts improves the prediction of KT model. Overall, the combination of any two models have higher prediction accuracy and this is especially true

that for the average ensemble of SOA and KT, called as AVG(SOA, KT), which has better accuracy than the other two combinations. Also, the linear regression of AM and KT has better prediction accuracy than linear regression combination of SOA and KT. However, from the two tailed paired t-test results shown in Table 6, the statistical difference between any two pairs of model combinations are not significant.

To examine whether there is significant difference between these models, we performed a 2-tailed paired t-test. The p values are shown in Table 6. We observe that most of the differences between two models are reliable, except for when we compare some AM and KT combined models with SOA and KT combined models. Both SOA and AM use the information about students' actions of hints and attempts. There might be a chance that SOA and LR(AM, KT) have some prediction overlap.

Table 6. Reliability when compare KT, SOA, AM, and Ensemble

	MAE	RMSE	AUC
AM vs SOA	**0.000**	**0.000**	**0.000**
AM vs KT	**0.000**	**0.000**	**0.000**
AM vs LG(AM, KT)	**0.000**	**0.000**	**0.000**
AM vs LR(SOA, KT)	**0.000**	**0.000**	**0.000**
AM vs AVG(SOA, KT)	**0.000**	**0.000**	**0.000**
SOA vs KT	**0.000**	**0.000**	**0.037**
SOA vs LG(AM, KT)	0.298	**0.030**	0.083
SOA vs LR(SOA, KT)	**0.000**	**0.001**	**0.006**
SOA vs AVG(SOA, KT)	**0.020**	**0.000**	**0.003**
KT vs LR(AM, KT)	**0.000**	**0.000**	**0.000**
KT vs LR(SOA, KT)	**0.000**	**0.000**	**0.000**
KT vs AVG(SOA, KT)	**0.000**	**0.000**	**0.000**
LR(AM, KT) vs LR(SOA, KT)	0.265	0.296	0.469
LR(AM, KT) vs AVG(SOA, KT)	0.271	0.138	0.079
LR(SOA, KT)vs AVG(SOA, KT)	0.258	**0.001**	**0.010**

4.2 Further Analysis for SOA and KT

From the last section, we observed the best model is the AVG(SOA,KT) model. In order to better investigate this combination, we ran student level and skill level analysis.

Tables 7 and 8 shows the student level result across 66 students to account for the non-independence of their actions. Take MAE as an example, for each student; a MAE is calculated based on all data available for that student. Then an average value for MAE is computed based on MAE of all students. Table 8 shows the t-test p value for each pair of these three models, where the remaining degrees of freedom on all the tests is 65.

Table 7. Student Level accuracy of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT	0.2939	0.3790	0.6738
SOA	0.2871	0.3767	0.6786
AVG(KT, SOA)	0.2905	0.3765	0.6811

Table 8. Student level reliability of difference of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT vs SOA	**0.0000**	**0.0000**	0.0551
KT vs AVG	**0.0000**	**0.0000**	**0.0000**
SOA vs AVG	**0.0000**	0.0698	0.0698

Note that there is no significant difference of AUC between KT and SOA. We interpret these results by pointing out that RMSE and AUC are metrics that are optimized for measuring different things, and so this is quite possible.

Table 9 and 10 shows the skill level result across all 51 skills. From Table 9 we observe a very low AUC value for all the models, which indicates these models do not make a good classification at skill level. The t-test p value with remaining degrees of freedom 50 is shown in table 10.

Table 9. Skill level accuracy of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT	0.3064	0.3762	0.4675
SOA	0.2942	0.3713	0.4769
AVG(KT, SOA)	0.3003	0.3710	0.492

Table 10. Skill Level reliability of difference of KT, SOA and Ensemble

	MAE	RMSE	AUC
KT vs SOA	**0.0000**	**0.0136**	0.3492
KT vs AVG	**0.0000**	**0.0002**	**0.0003**
SOA vs AVG	**0.0000**	0.3982	**0.0059**

The student and skill level analysis generate similar conclusions, that SOA and ensemble outperform KT in all of the three metrics. When we compare the ensemble model with SOA alone, the result is not so clear.

5. DISCUSSION AND FUTURE WORK

In this paper, we put forward a Sequence Of Action model that makes use of sequence of students attempts to answer questions and asking for hints. The SOA model consists of two parts. First, the sequence of students' actions are divided into five categories. A tabling method shows that students who only make attempts tend to answer the next question more correctly than students who only ask for hints. This could be caused by students who make more attempts are trying to figure problems out by themselves and it is an efficient way to master knowledge when they are told the steps to answer these questions by asking for hints. Second, we built a logistic regression model with next question correct percentage as dependent variable and skill_id, credits of sequence of action bins as independent variables.

We conducted six-fold cross validation experiments. The experimental result showed that SOA had reliably higher prediction accuracy than the Knowledge Tracing model and Assistance Model. The average combination of the SOA and KT had the highest prediction. In sum, the sequence of students' actions provided important information in predicting students' performance.

This work is the beginning of utilizing the sequence of asking for hints and making attempts recorded by intelligent

tutoring systems to better predict student performance. There are many open spaces for us to explore. For example, the experimental data we used is from ASSISTments, does SOA model still makes a big difference if use data from other intelligent tutor systems? How much can the performance of SOA model be improved if combined with other efficient prediction model such as PFA (Pavlik et al., 2009)? What is the SOA model's performance if we use a student action sequence of several previous questions when we train the model? How does SOA perform after individualization? These are some of the questions that still need to be explored.

6. CONTRIBUTION

Predicting student performance is an important part of the student modeling task in Intelligent Tutoring Systems. A large portion of papers at EDM have focused on this. Many models and techniques have been used to model and investigate students' performance. However, little attention been paid to the temporally sequential actions of student when interacting with the tutoring system. To our knowledge we are the first to use the temporal sequencing of hints and attempts. It turns out that by paying attention to this we can better predict student performance. In this paper, we introduce the Sequence of Action model which makes use of the click-stream data of the sequence of making attempts and asking for hints when students do their homework using an Intelligent Tutoring System. Students' actions can be very different from each other, but we found there are some useful patterns.

We can think of several ways to improve upon this. First, our five bins that we put students into were somewhat arbitrary. There could be more bins or fewer. If we use more bins, we might have very different predictions. The downside is that for some of these bins we might not have enough data points to reliably fit the parameters. One way to make the model better might be to split the "All Hints" bin into one that has "Reached Bottom out Hint" and one that is "All hints excluding those that reached the bottom out." We could also try to pay attention to features like response time between hints or the response time after a hint in making an attempt.

According to our six-fold cross validation experiments and paired two-tailed t-test, both on student level and skill level, our Sequence of Action model had reliably higher prediction accuracy than KT and AM, the later uses the number of hints students ask for and the number of attempts students make. Furthermore, we combined SOA and KT using average and linear regression methods, and the ensemble model's prediction performance was much better than either SOA or KT. We also compared combination of SOA and KT with combination of AM and KT. The experimental result show that SOA contributes more useful information than AM alone, which indicates that the sequential information of action does convey more information about students' learning than the statistics information of actions students make.

7. ACKNOWLEDGMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024).

8. REFERENCES

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253–278.

Baker, R.S.J.d., Corbett, A.T. & Aleven, V. (2008). More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In: Wolf, B., Aimeur, E., Nkambou, R., Lajoie, S. (Eds.) *Intelligent Tutoring Systems. LNCS, 5091*, Springer Berlin. pp. 406-415.

Feng, M., Heffernan, N. & Koedinger, K.R. (2006a). Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*. Springer-Verlag: Berlin. pp. 31-40.

Gong, Y., Beck, J. & Heffernan, N. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer-Verlag, Berlin. pp. 35-44.

Hawkins, W., Heffernan, N., Wang, Y. & Baker, S,J,d.. (2013). Extending the Assistance Model: Analyzing the Use of Assistance over Time. In S. D'Mello, R. Calvo, & A. Olney (Eds.) *Proceedings of the 6th International Conference on Educational Data Mining (EDM2013)*. Memphis, TN. pp. 59-66.

Pardos, Z. & Heffernan, N. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Paul De Bra, Alfred Kobsa, David Chin, (Eds.) *The 18th Proceedings of the International Conference on User Modeling, Adaptation and Personalization*. Springer-Verlag. pp. 255-266.

Pavlik, P.I., Cen, H., Koedinger, K.R. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*. Brighton, UK. pp. 531-538.

Qiu, Y., Qi, Y., Lu, H., Pardos, Z. & Heffernan, N. (2011). Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining*. pp. 139-148.

Shih, B., Kenneth K., & Richard S. (2010). Unsupervised Discovery of Student Strategies. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 201-210.

Wang, Y. & Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *The 24th International FLAIRS Conference*. Florida.

Using Similarity to the Previous Problem to Improve Bayesian Knowledge Tracing

William J. Hawkins
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
whawkins90@gmail.com

Neil T. Heffernan
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
nth@wpi.edu

ABSTRACT

Bayesian Knowledge Tracing (BKT) is a popular student model used extensively in educational research and in intelligent tutoring systems. Typically, a separate BKT model is fit per skill, but the accuracy of such models is dependent upon the skill model, or mapping between problems and skills. It could be the case that the skill model used is too coarse-grained, causing multiple skills to all be considered the same skill. Additionally, even if the skill model is appropriate, having problems that exercise the same skill but look different can have effects on student performance. Therefore, this work introduces a student model based on BKT that takes into account the similarity between the problem the student is currently working on and the one they worked on just prior to it. By doing this, the model can capture the effect of problem similarity on performance, and moderately improve accuracy on skills with many dissimilar problems.

Keywords

Student modeling, Bayesian Knowledge Tracing, Problem Similarity

1. INTRODUCTION

Bayesian Knowledge Tracing (BKT) [3] is a popular student model used both in research and in actual intelligent tutoring systems. As a model that infers student knowledge, BKT has helped researchers answer questions about the effectiveness of help within a tutor [1], the impact of "gaming the system" on learning [5], and the relationship between student knowledge and affect [9], among others. Additionally, it has been used in the Cognitive Tutors [6] to determine which questions should be presented to a student, and when a student no longer needs practice on a given skill.

However, BKT models are dependent upon the underlying skill model of the system, as a separate BKT model is typically fit per skill. If a skill model is too coarse-grained or too fine-grained, it can make it more difficult for a BKT model to accurately infer student knowledge [8].

Additionally, even when a skill model is tagged at the appropriate level, seeing similar problems consecutively as opposed to seeing dissimilar problems may have effects on guessing and slipping, two important components of BKT models. For example, if a student does not understand the skill they are working on, seeing a certain type of question twice or more consecutively may improve their chances of "guessing" the answer using a suboptimal procedure that would not work on other questions from the same skill.

Whether the skill model is not at the appropriate level or seeing consecutive similar questions helps students succeed without fully learning a skill, it may be important to take problem similarity into account in student models. In this work, we introduce the Bayesian Knowledge Tracing – Same Template (BKT-ST) model, a modification of BKT that considers problem similarity. Specifically, using data from the ASSISTments system [4], the model takes into account whether the problem the student is currently working on was generated from the same *template* as the previous problem.

The next section describes the ASSISTments system, its template system and the data used for this paper. Section 3 describes BKT and BKT-ST in more detail, and describes the analyses we performed on these models. The results are reported in Section 4, followed by discussion and possible directions for future work in Section 5.

2. TUTORING SYSTEM AND DATA

2.1 ASSISTments

ASSISTments [4] is a freely available web-based tutoring system used primarily for middle and high school mathematics. In addition to providing a way for teachers to assess their students, ASSISTments also assists the students in a few different ways: through the use of series of on-demand hint messages that typically end in the answer to the question (the "bottom-out hint"), "buggy" or feedback messages that appear when the student gives a common wrong answer, and "scaffolding" questions that break the original question into smaller questions that are easier to answer.

While teachers are free to author their own content, ASSISTments provides a library of approved content, which includes problem sets called *skill-builders*, which are meant to help students practice a particular skill. While most problem sets contain a fixed number of problems that must all be completed for a student to finish, a skill-builder is a special type of problem set that assigns questions in a random order and that is considered complete once a student answers three consecutive questions correctly on the same day.

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

While requiring students to answer three consecutive questions correctly on the same day to complete a skill-builder ensures that they have some level of knowledge of the particular skill being exercised, it takes some students many problems to achieve this, meaning they may see the same problem more than once if the skill-builder does not contain enough unique problems.

To ensure this does not happen (or at least make it highly unlikely), ASSISTments has a templating system that facilitates creating large numbers of similar problems quickly. The content creator creates a question as normal, but specifies that it is a *template* and uses variables in the problem statement and answer rather than specific values. Then, they are able to generate 10 unique problems at a time from that template, where each problem is randomly populated with specific values as prescribed by the template. This is especially useful for skill-builders, whose problems should theoretically all exercise the same skill. Figure 1 shows an example of a template (a) and a problem generated from it (b).

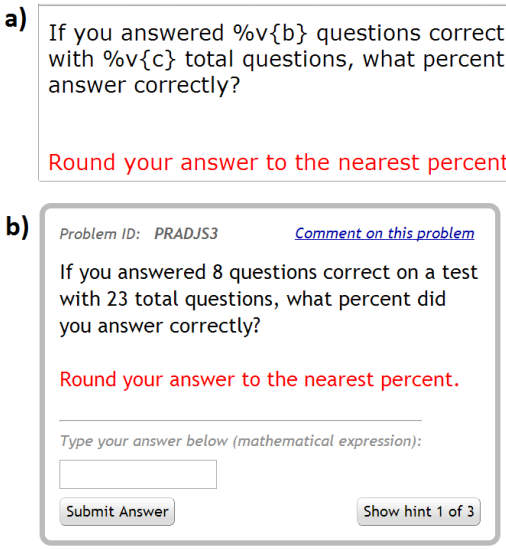


Figure 1. A template (top image) and a problem generated from it (bottom). The variables 'b' and 'c' in the template are replaced by '8' and '23' in the generated problem.

2.2 Data

In this work, we used ASSISTments skill-builder data from the 2009-2010 school year. This data set consists of 61,522 problem attempts by 1,579 students, spread across 67 different skill-builders. A (student, skill-builder) pair was only included if the student attempted three or more problems on that particular skill-builder, and a skill-builder was included if it was used by at least 10 students and at least one of them completed it.

3. METHODS

In this section, we begin by describing Bayesian Knowledge Tracing, and then move on to our modification of it, called Bayesian Knowledge Tracing – Same Template. Finally, we describe the analyses we performed using these two models.

3.1 Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [3] is a popular student model that uses a dynamic Bayesian network to infer student knowledge using only a student's history of correct and incorrect responses to questions that exercise a given knowledge component (or "skill").

Typically, a separate BKT model is fit for each skill. BKT models assume that there are only two states a student can be in for a given skill: the *known state* or the *unknown state*. Using a student's performance history on a given skill, a BKT model infers the probability that the student is in the *known state* on question *t*, $P(K_t)$.

Fitting a BKT model involves estimating four probabilities:

1. Prior Knowledge – $P(L_0)$: the probability the student knew the skill before answering the first question
2. Learn Rate – P(T): the probability the student will know the skill on the next question, given that they do not know the skill on the current question
3. Guess Rate – P(G): the probability the student will answer the current question correctly despite not knowing the skill
4. Slip Rate – P(S): the probability the student will answer the current question incorrectly despite knowing the skill

Note that forgetting is typically not modeled in BKT: it is assumed that once a student learns a skill, they do not forget it. An example of a BKT model, represented as a static unrolled Bayesian network, is shown in Figure 2.

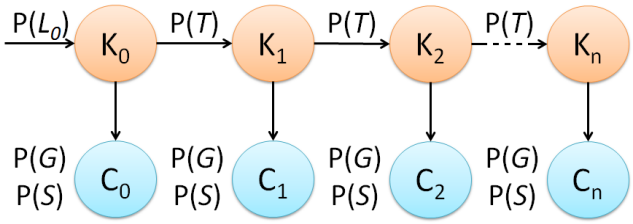


Figure 2. Static unrolled representation of Bayesian Knowledge Tracing. The K_t nodes along the top represent latent knowledge, while the C_t nodes represent performance.

3.2 Bayesian Knowledge Tracing – Same Template

The Bayesian Knowledge Tracing - Same Template (BKT-ST) model differs from the regular BKT model in one way: it takes into account whether the problem it's about to predict was generated from the same template as the previous problem the student worked on. This is modeled as a binary observed variable that influences performance.

This results in six parameters to be learned per skill: the initial knowledge rate, the learn rate, and two sets of guess and slip rates: one set for when the previous problem and current problem were generated from the same template (P(G|Same) and P(S|Same)), and one for when they aren't (P(G|Different) and P(S|Different)). The model is shown in Figure 3.

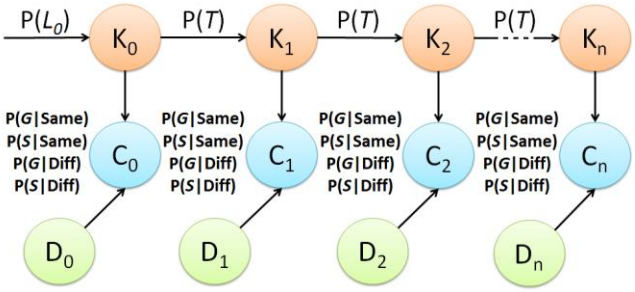


Figure 3. Static unrolled representation of Bayesian Knowledge Tracing – Same Template. The only difference from BKT is the presence of the D_t nodes, which represent whether the previous question was generated by the same template as the current one.

3.3 Analyses

The first analysis in this work simply considers how well the two models fit the data compared to each other overall. This is determined by fitting separate BKT and BKT-ST models for each skill and then predicting unseen student data using five-fold student-level cross-validation. Then, we evaluate each model's ability to predict next question correctness by computing the mean absolute error (MAE), root mean squared error (RMSE) and area under the curve (AUC) for each student and then averaging across students for each type of model. Finally, two-tailed paired t-tests are used to determine the significance of the differences in the metrics.

The second analysis considers what the metrics look like for each model based on how many templates were used for each skill-builder problem set. This is done by splitting the predictions made in the first analysis by how many templates were used in the corresponding skill-builder. We did this to see when it would be worth using BKT-ST over BKT.

Finally we consider the parameter values learned for the BKT-ST model to determine any effects that seeing problems generated by the same template consecutively has on guessing and slipping.

The BKT and BKT-ST models used in these analyses are fit using the Expectation-Maximization (EM) algorithm in the Bayes Net Toolbox for Matlab (BNT) [7]. The initial values given to EM for BKT were 0.5 for $P(L_0)$ and 0.1 for the other three parameters. This was also true for BKT-ST, except the slip rate was set to 0.2 when the current and previous problems were generated from the same template.

4. RESULTS

In this section, we first present the overall comparison of BKT and BKT-ST, then show how they compare to each other based on the number of templates used in each skill-builder. Finally, we examine the learned parameters for the BKT-ST model.

4.1 Overall

The overall results comparing BKT to BKT-ST are shown in Table 1.

Table 1. Overall results of fitting BKT and BKT-ST models.

	MAE	RMSE	AUC
BKT	0.3830	0.4240	0.5909
BKT-ST	0.3751	0.4205	0.6314

According to these results, BKT-ST outperforms BKT in all three metrics. Statistical tests confirmed that these results were reliable (MAE: $p < .0001$, $t(1578) = 9.939$; RMSE: $p < .0001$, $t(1578) = 4.825$; AUC: $p < .0001$, $t(1314) = -11.095$), though according to the values in the table, the only noticeable gain was in AUC.

4.2 By Number of Templates

Next, we considered how well each model did based on the number of templates a skill-builder contained. The results are shown in Figure 4.

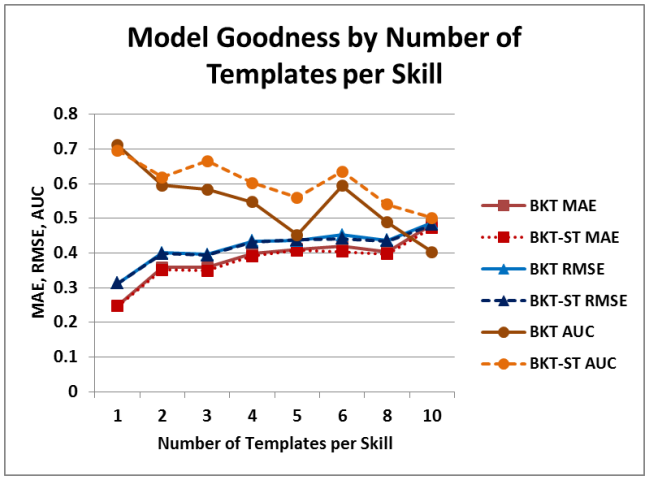


Figure 4. Graph of MAE, RMSE and AUC for the BKT and BKT-ST models, plotted against the number of unique templates per skill.

Interestingly, both BKT and BKT-ST decline rapidly in terms of model goodness as the number of templates per skill-builder increases. This is likely the case because those with more templates are more likely to have more than one skill being tested within them. Interestingly, although both models decline similarly in terms of MAE and RMSE, BKT-ST declines at a slower rate than BKT does in terms of AUC. In fact, BKT-ST outperforms BKT in terms of AUC for every group of skills with more than one template. When grouping the skills by the number of templates they had, BKT-ST achieved an AUC of at least 0.0236 better than BKT for each group that had more than one template, and achieved AUC values that were 0.1086 and 0.0980 better than BKT for skills with five and 10 templates, respectively. Additionally, while BKT performs worse than chance (AUC < 0.5) on skills with eight or more templates, BKT-ST never performs worse than chance.

4.3 Parameter Values

To analyze the parameters learned by BKT-ST, for each skill, we took the average value of each of the six parameters learned across the five folds from the overall analysis.

First, we computed the average value of each parameter across all 67 skills. These are shown in Table 2.

Table 2. Means and standard deviations of BKT-ST parameter values learned across 67 skill-builders

Parameter	Mean	SD
$P(L_0)$	0.6030	0.2617
P(T)	0.2966	0.2500

P(G|Different)	0.1880	0.1655
P(S|Different)	0.2941	0.1737
P(G|Same)	0.3337	0.2495
P(S|Same)	0.1514	0.0848

From the results in Table 2, it appears that on average, seeing consecutive questions generated from the same template both increases the guess rate ($p < .0001$, $t(66) = -4.516$) and decreases the slip rate ($p < .0001$, $t(66) = 7.186$).

Next, we examined how these parameters changed with respect to the number of templates used per skill-builder. The average values of the performance parameters (guess and slip rates for same and different templates) are shown in the graph in Figure 5. The results for skills with one template are omitted since the P(G|Different) and P(S|Different) parameters are meaningless in such cases.

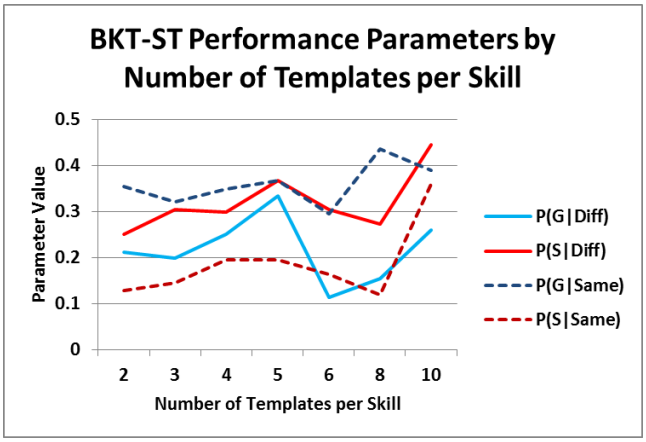


Figure 5. Average value of each performance parameter for the number of templates used per skill-builder.

Although there is no clear pattern for any of the four performance parameters shown in the graph, the average value of P(G|Same) is always higher than that of P(G|Different), and that of P(S|Same) is always lower than that of P(S|Different), with respect to the number of templates used per skill. This appears to reinforce the notion that seeing consecutive problems generated from the same template makes the latter easier to solve, whether this is due to the skill model being too coarse-grained or familiarity with a certain type of problem within a skill inflating performance.

5. DISCUSSION AND FUTURE WORK

From the results in this work, it appears that modifying Bayesian Knowledge Tracing to take similarity between consecutive problems into account moderately improves cross-validated predictive performance, especially in terms of AUC. Additionally, this work showed that seeing consecutive similar problems improves student performance by both increasing the guess rate – the probability of answering a question correctly despite not knowing the skill – and decreasing the slip rate – the probability of answering a question incorrectly despite knowing the skill. Regardless of the underlying reason for this, whether it is because the skill model is too coarse-grained or simply that familiarity with a type of problem within a skill improves performance, it appears important for student models to take the similarity of the problems students encounter into account when trying to model student knowledge.

One direction for future work would be to try going back further in the problem sequence to see how the similarity of problems earlier in a student's history affects their ability to answer the current problem. Additionally, it would be interesting to determine whether the effect changes in certain situations. For example, what is the effect of seeing two similar problems in a row, followed by one that is different from both?

Another area of interest would be to use a model that takes problem similarity into account when trying to predict a longer-term outcome, such as wheel-spinning [2], retention and transfer, as opposed to simply predicting next question correctness.

Finally, applying this model and others like it to other learning environments and skill models of various grain sizes would be helpful for understanding when it is useful. Presumably, if a skill model is at the appropriate grain size, the difference in predictive performance between BKT and BKT-ST would be reduced. The same would be true of systems that fall to one of two extremes: those whose problem sets are highly repetitive, and those whose problem sets have a rich variety of problems.

6. ACKNOWLEDGMENTS

We acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), ONR's 'STEM Grand Challenges' and IES (# R305A120125 & R305C100024).

7. REFERENCES

[1] Beck, J.E., Chang, K., Mostow, J., Corbett, A. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *Intelligent Tutoring Systems*, Springer Berlin Heidelberg, 2008, 383-394.

[2] Beck, J. E., and Gong, Y. Wheel-Spinning: Students Who Fail to Master a Skill. In *Artificial Intelligence in Education*, pp. 431-440. Springer Berlin Heidelberg, 2013.

[3] Corbett, A. and Anderson, J. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253-278.

[4] Feng, M., Heffernan, N.T., Koedinger, K.R. Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. *User Modeling and User-Adapted Interaction*, *19*(3), 243-266.

[5] Gong, Y., Beck, J., Heffernan, N., Forbes-Summers, E, The impact of gaming (?) on learning at the fine-grained level. in *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, (Pittsburgh, PA, 2010), Springer, 194-203.

[6] Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A. (1997). Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education*, *8*(1), 30-43.

[7] Murphy, K. The bayes net toolbox for matlab. *Computing science and statistics*, *33*(2), 1024-1034.

[8] Pardos, Z. A., Heffernan, N. T., & Anderson, B., Heffernan, C. L. Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. *Proceedings of the Workshop in Educational Data Mining held at the 8th Interna-*

tional Conference on Intelligent Tutoring Systems. (Taiwan, 2006).

[9] San Pedro, M., Baker, R.S.J.d, Gowda, S.M., Heffernan, N.T. Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, H.C., Yacef, K., Mostow, M., Pavlik, P. (Eds.) AIED 2013. LNCS, vol. 7926/2013, pp.41-50. Springer-Verlag, Berlin Heidelberg.

Is this Data for Real?

Rinat B. Rosenberg-Kima
University of California, Berkeley
rosenbergkima@berkeley.edu

Zachary Pardos
University of California, Berkeley
pardos@berkeley.edu

ABSTRACT
Simulated data plays a central role in Educational Data Mining and in particular in Bayesian Knowledge Tracing (BKT) research. The initial motivation for this paper was to try to answer the question: *given two datasets could you tell which of them is real and which of them is simulated?* The ability to answer this question may provide an additional indication of the goodness of the model, thus, if it is easy to discern simulated data from real data that could be an indication that the model does not provide an authentic representation of reality, whereas if it is hard to set the real and simulated data apart that might be an indication that the model is indeed authentic. In this paper we will describe initial analysis that was performed in an attempt to address this question. Additional findings that emerged during this exploration will be discussed as well.

Keywords
Bayesian Knowledge Tracing (BKT), simulated data, parameters space.

1. INTRODUCTION
Simulated data has been increasingly playing a central role in Educational Data Mining [1] and Bayesian Knowledge Tracing (BKT) research [1, 4]. For example, simulated data was used to explore the convergence properties of BKT models [5], an important area of investigation given the identifiability issues of the model [3]. In this paper, we would like to approach simulated data from a slightly different angle. In particular, we claim that the question,"*given two datasets could you tell which of them is real and which of them is simulated?",* is interesting as it can be used to evaluate the goodness of a model and may potentially serve as an alternative metric to RMSE, AUC, and others. We would like to start approaching this problem in this paper by comparing simulated data to real data with Knowledge Tracing as the model.

Knowledge Tracing (KT) models are widely used by cognitive tutors to estimate the latent skills of students [6]. Knowledge tracing is a Bayesian model, which assumes that each skill has 4 parameters: two knowledge parameters including initial (prior knowledge) and learn rate, and two performance parameters including guess and slip. KT in its simplest form assumes a single point estimate for prior knowledge and learn rate for all students, and similarly identical guess and slip rates for all students. Simulated data has been used to estimate the parameter space and in particular to answer questions that relate to the goal of maximizing the log likelihood (LL) of the model given parameters and data, and improving prediction power [7], [8], [9].

In this paper we would like to use the KT model as a framework for comparing the characteristics of simulated data to real data, and in particular to see whether it is possible to distinguish between the real and sim datasets.

2. DATA SETS
To compare simulated data to real data we started with 2 real dataset generated from the assisstment software[1] (specifically, datasets G6.207-exact.txt with 776 students and G6.259-exact.txt with 212 students) from a previous BKT study [10]. Both of the datasets consist of 6 questions in linear order where all students answer all questions. Next, we generated synthetic, simulated data using the best fitting parameters that were found for the real data as the generating parameters. By this we generated a simulated version of dataset G6.207 and a simulated version of dataset G6.259 that had the exact same number of questions, number of students, and was generated with what appears to be the best fitting parameters. The specific best fitting parameters that were found for each dataset and were used to generate the simulated data are presented in table 1.

Table 1. Best fitting parameters for each dataset. These parameters were used to generate the simulated datasets.

	N	Prior	Learn	Guess	Slip
G6.207	776	.453	.068	.270	.156
G6.259	212	.701	.044	.243	.165

3. METHODOLOGY
We are interested to find out whether it is possible to distinguish between the simulated data and the real data. The approach we took was to calculate LL for the gird of all the parameters space (prior, learn, guess, and slip). We hypothesized that the LL pattern of the simulated data and real data will be different across the parameters space. For each of the matrices we conducted a grid search with intervals of .04 that generated 25 intervals for each parameter and 390,625 total combinations of prior, learn, guess, and slip. For each one of the combinations LL was calculated and placed in a four dimensional matrix. We used fastBKT [11] to (a) calculate the best fitting parameters of the real datasets, (b) generate simulated data, and (c) calculate the LL of the parameters space. Additional code in Matlab and R was generated to put all the pieces together[2]. In particular, we calculated the LL for all the combinations of two parameters where the other two parameters were fixed to the best fitting value. In an additional analysis, we let all parameters be free and took the average LL for all combinations of two parameters, collapsed over the space of the other two parameters not visualized. The motivation for this was to visualize the error space interactions in the four dimensions of the model.

[1] Data can be obtained here: http://people.csail.mit.edu/zp/

[2] Matlab and R code will be available here:
[2] Matlab and R code will be available here:
http://myweb.fsu.edu/rr05/

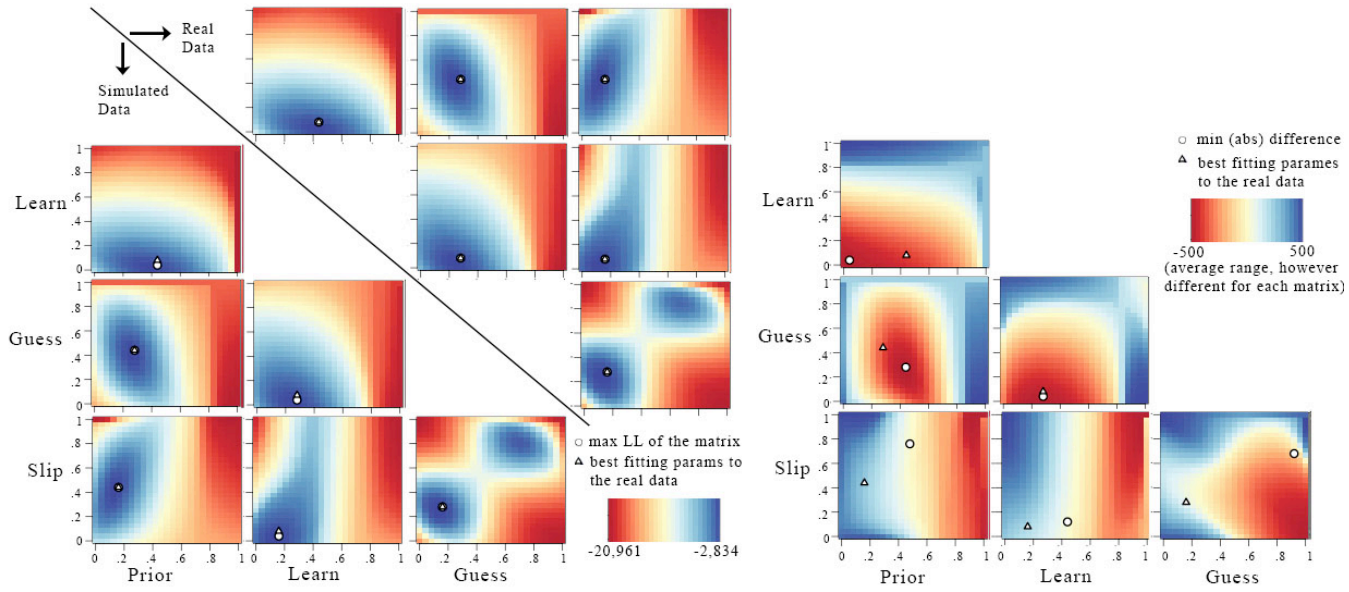


Figure 1.a (left). Heat maps of LL of real assistment dataset G6-207 (k=776 students) and a corresponding simulated data that was generated with the best fitting parameters of the real dataset. The two parameters not in each figure were fixed to the best parameters. Blue areas indicate high LL, and red areas indicate lower LL. Circles indicate maximum LL of the given matrix, and triangles indicate the best fitting parameters to the real data (that were also used to generate the simulated data). In this case the triangles and circles fit the same point.

Figure 1.b (right). Heat maps of delta LL between real dataset G6-207 and the corresponding simulated data that was generated with the best fitting parameters of the real dataset. The two parameters not in each figure were fixed to the best parameters. Blue areas indicate high difference between the real and sim LL, and red areas indicate lower difference. Circles indicate minimum absolute delta of the given matrix, and triangles indicate the best fitting parameters to the real data.

4. DOES THE LL OF SIM vs. REAL DATA LOOK DIFFERENT?

Our initial thinking was that as we are using a simple BKT model, it is not authentically reflecting reality in all its detail and therefore we will observe different patterns of LL across the parameters space between the real data and the simulated data. The LL space of simulated data in [5] was quite striking in its smooth surface but the appearance of real data was left as an open research question.

4.1 Does the LL of sim vs. real data looks different across two parameters grids?

First, we calculated the LL over all the combinations of two parameters for dataset G6.207 where the other two parameters were fixed to the best fitting value. For example, when we calculated LL for the combination of slip and prior (top right figure in figure 1.a), we fixed learn and guess to be .068 and .270 accordingly. To our great surprise, when we plotted heat maps of the LL matrices of the real data and the simulated data (Figure 1.a - real data is presented in the upper triangle and simulated (sim) data is presented in the lower triangle) we received what appears to be identical matrices (for example, the upper right heat map is the (slip x prior) LL matrix of the real data, whereas the lowest left heat map is the (slip x prior) LL matrix of the sim data).

The extent of the similarity between the matrices was surprising and in order to get a better picture of the differences between them we plotted heat maps of the deltas between the real data and the simulated data (LL_RealData-LL_SimData) for each matrix. Even though the matrices appear to be identical, as can be seen in Figure 1.b, there is in fact a difference between the LL of the matrices although it is not a big difference compared to the values of LL. Another surprising finding was that the LL of the real data was in many cases higher than the LL of the sim data. We expected that the model would better explain the sim data as there should not be additional noise as expected in reality, and therefore the LL of the sim data should be higher, yet the findings were not consistent with this expectation.

Another interesting finding was that the location of the ground truth (the triangle) in most of the cases resulted in smaller delta between the real and the sim data although not in all cases (e.g., guess x slip). Note that the circles in Figure 1.b indicate the minimum absolute difference in LL between the real and the sim data, and this point is usually not located at the exact ground truth (except for learn x guess).

Another interesting finding can be seen in Figure 1.a - slip vs. guess. Much attention has been given to this LL space which revealed the apparent co-linearity of BKT with two primary areas of convergence, the upper right area being a false, or "implausible" converging area as defined by [3]. What is interesting in this figure is that despite what appears to be two global maxima, the point with the best LL in this dataset is in fact the lower region for both sim and real data.

Next we conducted the same analysis with the second dataset.

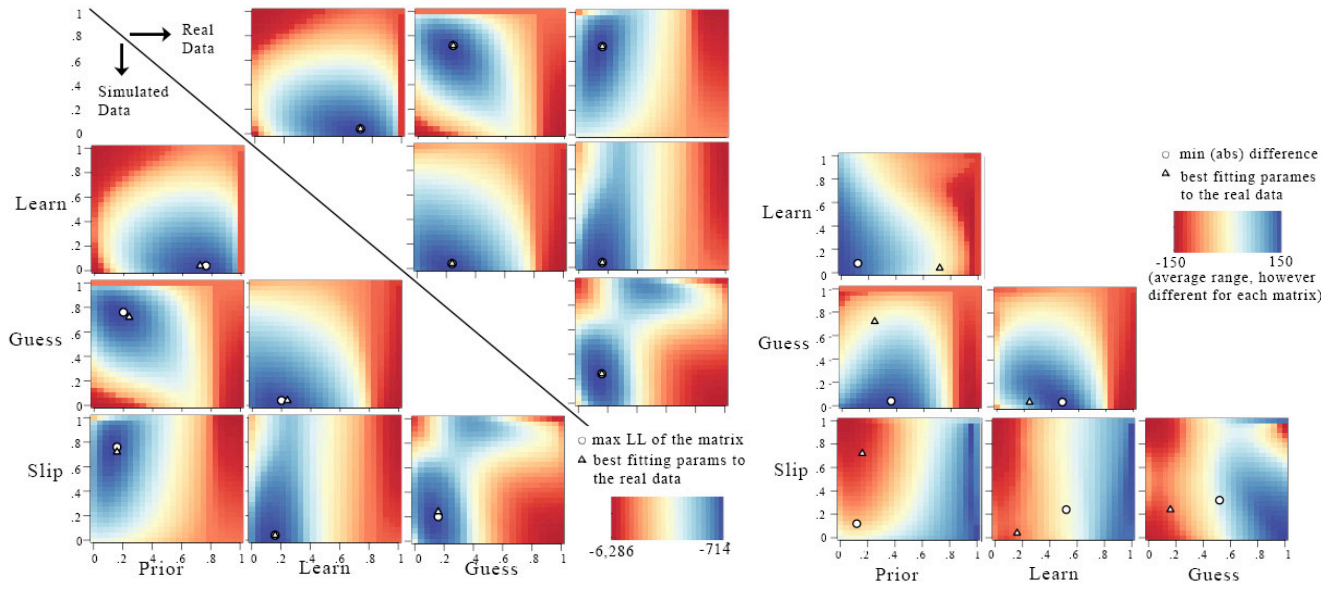


Figure 2.a (left) Heat maps of delta LL between real dataset G6-259 (k=212 students) and the corresponding simulated data that was generated with the best fitting parameters of the real dataset. The two parameters not in each figure were fixed to the best parameters. Blue areas indicate high difference between the real and sim LL, and red areas indicate lower difference. Circles indicate maximum LL of the given matrix, and triangles indicate the best fitting parameters to the real data.

Figure 2.b (right). Heat maps of delta LL between real assistment dataset G6-259 and the corresponding simulated data that was generated with the best fitting parameters of the real dataset. The average is across the two parameters not in each figure. Blue areas indicate high difference between the real and sim LL, and red areas indicate lower difference. Circles indicate minimum absolute delta of the given matrix, and triangles indicate the best fitting parameters to the real data.

Even though the G6-259 dataset was significantly smaller than the first dataset, we received very similar results to the first dataset with surprisingly similar heat maps for the sim and real data (see Figure 2.a). Like in the first dataset, notice that even though the LL heat maps look very similar, there is a difference in the delta heat maps (see Figure 2.b). Nevertheless, there is an interesting difference between the two datasets. Concretely, unlike the bigger dataset (G6-207), in G6-259 the LL of the sim data was actually higher than the real data in most cases.

4.2 What if we average LL over 2 parameters across all the combinations of the other 2 parameters?

We were interested to find out how will the heat maps look like if we do not fix the other two parameters to be best fit, but rather average the LL across the entire space of the other two parameters. For example, to calculate the matrix of guess and slip we practically calculated a matrix of guess and slip LL for each combination of learn and prior (25 x 25 = 625 matrices) instead of only one matrix for the best fit learn and prior. Then, we took the average of all these matrices for each combination of guess and slip (see Figure 3.a). The results are both surprising and interesting. As far as (guess x slip), we no longer receive the two maximum (global and local) that we received when learn and prior where fixed to best fit parameters. Another interesting finding is the relationship between the average maximum across the other two parameters and the overall best fit parameters for given two parameters. For example, if we look at the heat map of matrix (learn x prior) we can see that there is not a big difference between the average maximum point (white circle) and the overall best fit parameters (white triangle). This may indicate that changing guess and slip will not affect the value of learn and prior that maximizes the LL, therefore might suggest independency. If we look at (guess x learn), we see that changes in prior and slip will again not have an impact on the best fit value of guess, however, they will affect the value of learn. Then again, if we look at the heat map of (prior x guess), we will see that both prior and guess are sensitive to changes in learn and slip. Yet again, the extremely surprising part of these results is that the sim data appear to be almost identical to the real data. It is possible to see from Figure 3.b though that indeed there are differences between the simulation data and the real data and like before, the LL of the real data is higher than that of the sim data in the larger dataset.

Like for the fixed matrices, we received similar LL matrices for the smaller dataset (G6-259) (see table 4.a). In addition, as before, the LL of the sim data for this dataset was higher than that of the real data (the opposite direction of the larger dataset G6-207). Another interesting finding for this dataset can be seen in the (guess x slip) matrices (4.b). Notice that while the sim data converged to the lower point of the blue area, the real data converged to the higher point. Nevertheless, this only happened in the averages matrices and not in the fixed ones.

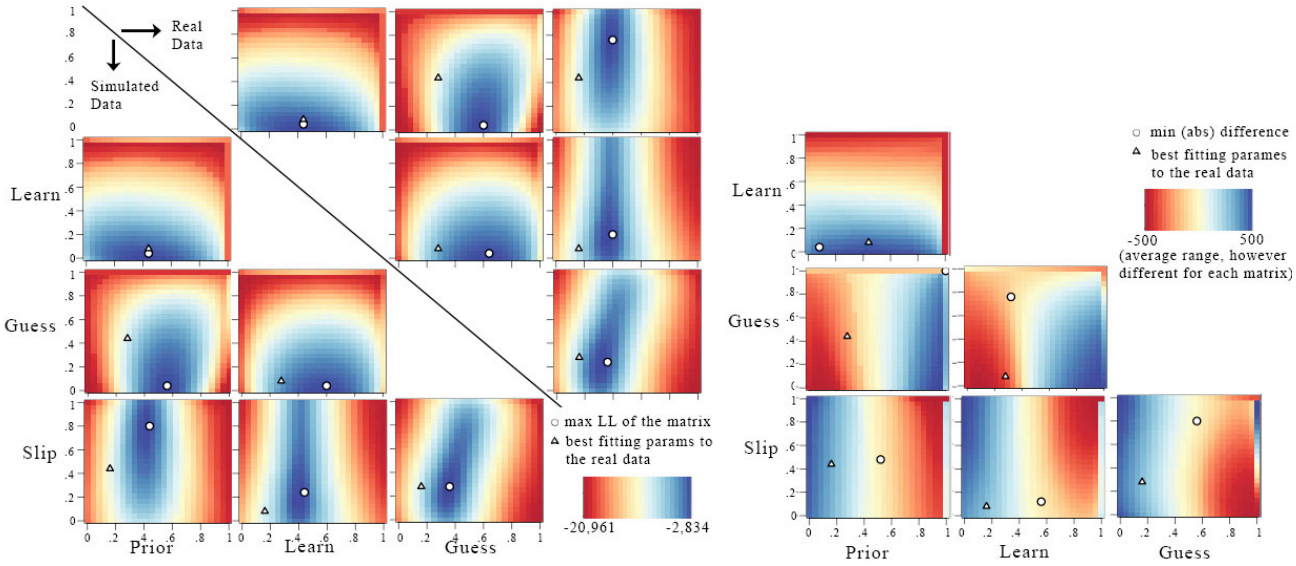


Figure 3.a (left). Heat maps of average LL of real assistment dataset G6-207 (k=776 students) and a corresponding simulated data that was generated with the best fitting parameters of the real dataset. The average is across the two parameters not in each figure. Blue areas indicate high LL, and red areas indicate lower LL. Circles indicate maximum LL of the given matrix, and triangles indicate the best fitting parameters to the real data (that were also used to generate the simulated data).

Figure 3.b (right). Heat maps of delta LL between real assistment dataset G6-207 and the corresponding simulated data that was generated with the best fitting parameters of the real dataset. The average is across the two parameters not in each figure. Blue areas indicate high difference between the real and sim LL, and red areas indicate lower difference. Circles indicate minimum absolute delta of the given matrix, and triangles indicate the best fitting parameters to the real data.

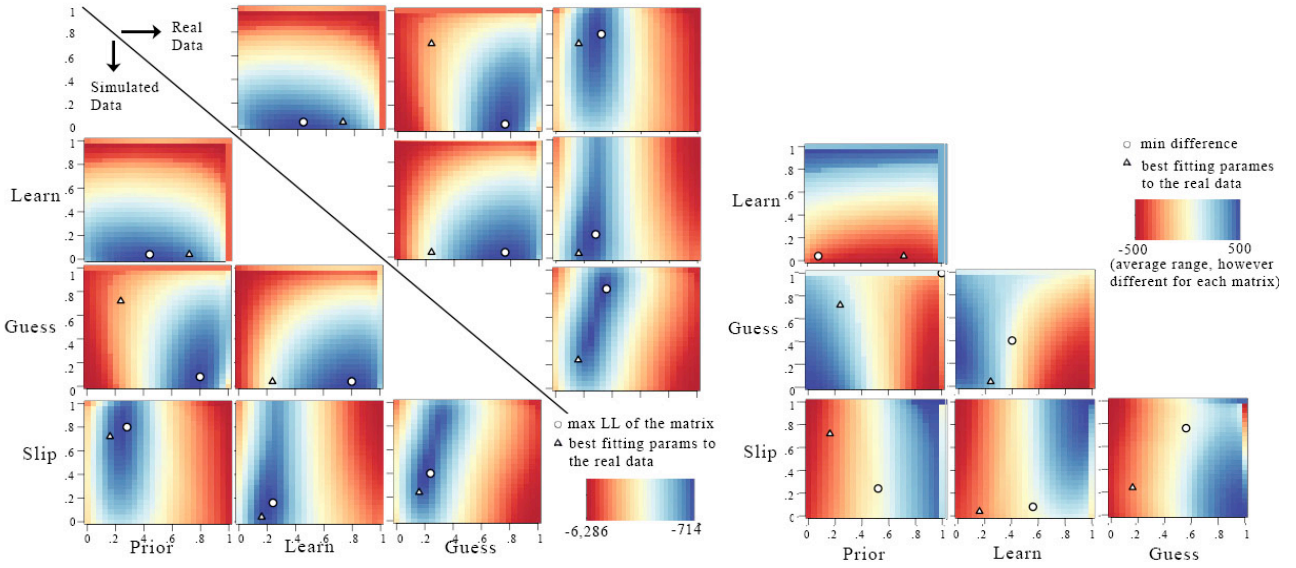


Figure 4.a (left). Heat maps of average LL of real assistment dataset G6-259 (k=212 students) and a corresponding simulated data that was generated with the best fitting parameters of the real dataset. The average is across the two parameters not in each figure. Blue areas indicate high LL, and red areas indicate lower LL. Circles indicate maximum LL of the given matrix, and triangles indicate the best fitting parameters to the real data (that were also used to generate the simulated data).

Figure 4.b (right). Heat maps of delta LL between real assistment dataset G6-259 and the corresponding simulated data that was generated with the best fitting parameters of the real dataset. The average is across the two parameters not in each figure.

5. DISCUSSION AND FUTURE WORK

The initial motivation of this paper was to find whether it is possible to discern a real data from a sim data. If for a given model it is possible to tell apart a sim data from a real data then the authenticity of the model can be questioned. This line of thinking is in particular typical of simulation use in Science context, where different models are used to generate simulated data, and then if a simulated data has a good fit to the real phenomena at hand, then it may be possible to claim that the model provides an authentic explanation of the system [12]. We believe that it may be possible to generate a new matric for evaluating the goodness of a model by comparing a simulated data from this model to real data.

In this work we explored similarities between simulated and real data. Nevertheless, we are yet to answer the question "is this data for real?". In other words, what we still did not do in this work is come up with an algorithm that can take a dataset and determine whether it is real or simulated. Another way to think of it is to come out with an algorithm that can tell us whether it is possible to discern real and simulated data and use it as an indication of the goodness of the model. We found differences between the real and sim data, but are they strong enough to be noticed by such algorithm in a consistent way? In future work we plan to further investigate this question by creating a training set of multiple real datasets and sim datasets and use machine learning techniques to extract a learning algorithm from this training dataset that can take as input a dataset and determine whether it is real or sim. We argue that if such algorithm can be found, it is an indication that the underlying model can be improved. In future work we also plan to compare different variations of the KT model and contrast their resulting simulated data with real data. In particular we plan to generate a more complex set of simulated data that is based on a more complex model (e.g., different learning rate for different types of questions), and then use it as "real" data with the (wrong) assumption that the model is simple (standard BKT model) to simulate a scenario where the real data is indeed grounded in more complex model than our assumptions and see what results would a learning algorithm that uses this "real" data in comparison to a sim data will yield.

In addition, this paper raises interesting questions that we did not think of while trying to answer our initial question. For example, it seems like there is potential to dive deeper into the average LL (Figures 3&4) and find more about the relationships and dependencies between the different parameters. Another question that emerged is how could it be that the simulated data had lower LL than the real data in the bigger dataset yet lower in the smaller dataset? Further analysis is needed to answer these questions.

Last but not least, given the remarkable resemblance between the sim data and the real data, these initial findings provide an indication that the BKT model is a model with a very strong hold in reality.

6. REFERENCES

[1] R. S. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," *J. Educ. Data Min.*, vol. 1, no. 1, pp. 3–17, 2009.

[2] M. C. Desmarais and I. Pelczer, "On the Faithfulness of Simulated Student Performance Data.," in *EDM*, 2010, pp. 21–30.

[3] J. E. Beck and K. Chang, "Identifiability: A fundamental problem of student modeling," in *User Modeling 2007*, Springer, 2007, pp. 137–146.

[4] Z. A. Pardos and M. V. Yudelson, "Towards Moment of Learning Accuracy," in *AIED 2013 Workshops Proceedings Volume 4*, 2013, p. 3.

[5] Z. A. Pardos and N. T. Heffernan, "Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm.," in *EDM*, 2010, pp. 161–170.

[6] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapt. Interact.*, vol. 4, no. 4, pp. 253–278, 1994.

[7] S. Ritter, T. K. Harris, T. Nixon, D. Dickison, R. C. Murray, and B. Towle, "Reducing the Knowledge Tracing Space.," *Int. Work. Group Educ. Data Min.*, 2009.

[8] R. S. d Baker, A. T. Corbett, S. M. Gowda, A. Z. Wagner, B. A. MacLaren, L. R. Kauffman, A. P. Mitchell, and S. Giguere, "Contextual slip and prediction of student performance after use of an intelligent tutor," in *User Modeling, Adaptation, and Personalization*, Springer, 2010, pp. 52–63.

[9] R. S. Baker, A. T. Corbett, and V. Aleven, "More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing," in *Intelligent Tutoring Systems*, 2008, pp. 406–415.

[10] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a bayesian networks implementation of knowledge tracing," in *User Modeling, Adaptation, and Personalization*, Springer, 2010, pp. 255–266.

[11] Z. A. Pardos and M. J. Johnson, "Scaling Cognitive Modeling to Massive Open Environments (in preparation)," *TOCHI Spec. Issue Learn. Scale*.

[12] U. Wilensky, "GasLab—an Extensible Modeling Toolkit for Connecting Micro-and Macro-properties of Gases," in *Modeling and simulation in science and mathematics education*, Springer, 1999, pp. 151–178.

The Effect of Variations of Prior on Knowledge Tracing

Matti Nelimarkka
School of Information, UC Berkeley
102 South Hall
Berkeley, California 94720-4600
Helsinki Institute for Information Technology HIIT,
Aalto University
PO Box 15600
Aalto, Finland 00076
matti.nelimarkka@hiit.fi

Madeeha Ghori
Department of Electrical Engineering and
Computer Sciences, UC Berkeley
387 Soda Hall
Berkeley, California 94720-17761
madeeha.ghori@berkeley.edu

ABSTRACT
Knowledge tracing is a method which enables approximation of a student's knowledge state using a Bayesian network for approximation. As the applications of this method increase, it is vital to understand the limits of this approximation. We are interested how well knowledge tracing performs when students' prior knowledge on the topic is extremely high or low. Our results indicate that the estimates become more erroneous when prior knowledge is extremely high ($prior = 0.90$).

Keywords
bayesian knowledge tracing, personalization, prior, parameter estimation

1. INTRODUCTION
The Bayesian Knowledge-Tracing (BKT) algorithm was developed in 1995 in an effort to model students' changing knowledge state during skill acquisition [5]. The idea is to interpret students' knowledge – a hidden variable – based on observed answers to a set of questions. The algorithm tracks the change in this probability distribution over time using a simple Bayes' net. The model is often presented as four parameters: prior, learn, guess and slip (see Figure 1). *Prior* refers to the probability that the student knows the material initially, before acquiring any skills, *learn* indicates that the student did not have the skill initially but acquired it through doing the exercise, *guess* refers to accidentally answering the question correct and *slip* to answering accidentally wrong.

Knowledge tracing is the most prominent method used to model student knowledge acquisition and is used in most intelligent learning systems. These systems have been said to be outperforming humans since 2001 [3] and have been used in the real world to tutor students [4]. For these reasons it is important to fully understand the strengths and limitations of knowledge tracing before applying it more widely in the classroom. As the parameters of the model are now known, there is a need to estimate these parameters from the given data. Previous research has demonstrated that the accuracy of parameter estimation – and therefore knowledge tracing – can be improved by applying different heuristics [17, 13] or methods [16, 18] including personalizing the model for each user [20, 8] or by extending the data used for analysis [15, 6, 1].

Our work starts from a different premise: how robust is the BKT approach to variation in the parameter space? Our special interest is in the *prior* variable, which correlates to a student's knowledge of the topic before answering a question. In any classroom, MOOC or otherwise, some students will come in with a better understanding of the material than others. Therefore it is important to study the effectiveness of knowledge tracing on parameter estimation when prior is extremely high or low.

If knowledge tracing models are inaccurate in modelling students of a certain prior parameter, then smart tutors and other systems designed to help those students learn will be less effective. Especially if the students being modelled inaccurately are those students doing poorly in the class, as the smart tutors exist to help them the most.

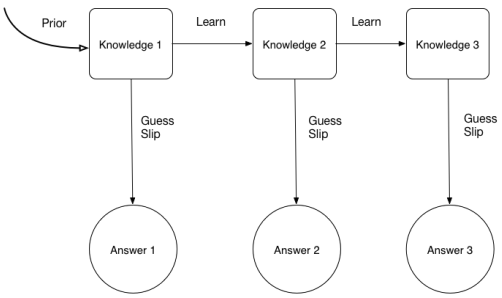


Figure 1: The model of knowledge tracing

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

2. PREVIOUS WORK

For the purposes of this work, here we shortly summarize three methods previously applied to improve the prediction capabilities of BKT models. However, these methods are insufficient to address the practical problem described above, resulting in a need for our own experiment.

2.1 Individualization

Yudelson et al. [20] experimented with individualization by bringing student-specific parameters into the BKT algorithm on a larger scale. They split the usual skill-specific BKT parameters into two components: one skill-specific and one student-specific. They then built several individualized BKT models and added student-specific parameters in batches, examining the effect each addition had on the model's performance. They found that student-specific prior parameters did not provide a vast improvement. However, student-specific learning provided a significant improvement to the model's prediction accuracy.

Pardos and Heffernan furthered the experiment by developing a method of formulating the individualization within the Bayes' Net framework [11]. Especially interesting in terms of our work is the difference prior values and methods suggested for this individualization. Pardos observes that models taking student spesific priors based on students' prior knowledge clearly outperform traditional knowledge trace approach. This is a contrast Yudelson et al.'s findings [20] but it still underscores the importance of individualization in the BKT algorithm.

Related to individualization per user, there have been discussion on using different values per resources. It can be argued that different exercises teach different topics [7, 14]. This can be further used to individualize the model for different topics, an approach which has gained initial support on empirical studies [14].

2.2 Enhancing the data

The second approach to improve these methods is related to enhanching the data used for prediction. In its most simple form, this can be done by adding additional relevant data, such as data from past years, to the analysis [15]. Others have explored the possibility of adding more data to the general domain-related knowledge on the models, and suggest that these indeed improve the estimates [6].

However, the current direction in enhanced data relates to information available on user interaction – especially in MOOC environments where it is possible to access this kind of data. To illustrate, Baker, Corbett, and Aleven [1] explore interactions with the learning system and other non-exercise related data, such as time spent on answering and asking help, to determine the difference between slips and guesses.

We applaud these efforts and acknowledge that data other than just student responses may indeed help to detect both the cases where initial knowledge (prior) is high and when it is low, instead of tweaking the EM algorithm further.

2.3 Improving the methods

There are several heuristics currently used to enhance the BKT algorithm. One such heuristic involves expecting the sum of slip and guess to be less than or equal to 1 [17]. Other work determined that one's starting estimated parameters could affect where the algorithm converged to. In order to improve the accuracy of the convergence, it was suggested that starting parameters be selected from a Dirichlet distrbution derived from the data set [2, 13].

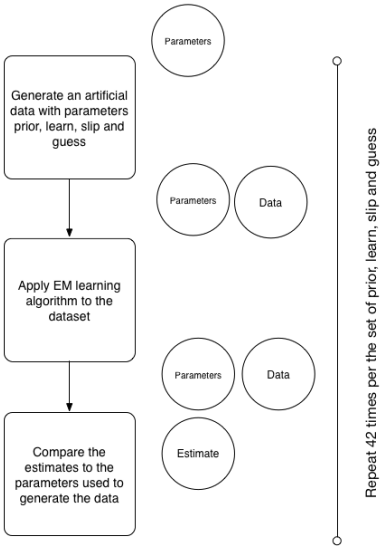


Figure 2: The approach used in this study

There have also been efforts to explore other machine learning methods on educational data. Initial trials born in the KDDCup competition use a medley of random forests and other machine learning algorithms but these methods have proven largely unsuccessful [16, 18].

The knowledge tracing community, while accepting the validity of some of these heuristics [9, 12], has criticized their inability to provide any insight into the student learning model. Individualization, however, has the potential to improve the BKT algorithm while also providing a pedagogical explanation for said improvements.

3. METHODOLOGY

We began by generating datasets with specific known initial parameters in order to simulate groups of students at different knowledge levels. We then ran expectation maximization (EM) on these datasets and allowed knowledge tracing to calculate its own estimated parameters. We then compared these estimated parameters to the original ones used for generation to determine if the accurency of the parameter estimation depends on the initial parameters.

Table 1: Ground Truth Parameter Sets

	prior	learn	guess	slip
Set 1.1 ... 1.6	0.15	0.10	0.10	0.05
Set 2.1 ... 2.6	0.30	0.10	0.10	0.05
Set 3.1 ... 3.6	0.15	0.20	0.10	0.05
	⋮			
Set 48.1 ... 48.6	0.90	0.20	0.20	0.10

3.1 Generating the Data

As our goal was to determine how the prior ground truth affects parameter estimation, we varied the prior used to synthesize the data sets. We used six different priors $(0.15, 0.30, \ldots, 0.75, 0.9)$, and two variations on learn, slip and guess[1] each (see Table 1); total of 48 variations of these parameters. Each of these data sets consists of 10,000 students and 20 observations per student. To increase the variation, we generated 6 datasets per condition. This kind of simulated approach has been previously used to evaluate the success of Bayesian machine learning methods [8].

3.2 Analysis Procedure

For each data set, we estimated the parameters using the *expectation maximization fitting* (EM) algorithm using the fastHMM implementation [10]. The parameter estimation was conducted using a grid search with ten parameters, and the best fitting model was selected using the log likelihood.

Using our 288 data sets, we can compare the estimates and ground truths for each parameter and analyze the accuracy of the estimates. We apply the standard methods of root-mean-square error (RMSE) and other visualizations to do our analysis. Using RMSE, we will be able to see if certain ground truths lend themselves to more accurate estimations.

4. RESULTS

First, let us explore the parameter estimation in detail. The avarage RMSE measurement in the data (Table 2) indicate that the prediction quality decreases as the prior increases; there is also increase of variance of the RMSE. This indicates that the predictions with higher priors are first more erronous and second, they converge in a larger area, resulting in variance. To confirm our observations, we conducted a Wilcox-Mann-Whitney test to explore if the computed RMSEs differented in statistically significant manner. As shown in Table 3, both the RMSEs computed from the data sets with priors 0.15 and 0.90 statistically differ significantly from the other datasets ($p < 0.05$). Therefore we conclude that the EM algortihm performs badly when prior is high.

To further understand this phenomena, we explore the estimates per parameter. The errors per parameter are shown in the Figure 3. The mean estimates are rather constantly close by the zero, though a higher prior does affect variance. As ground truth prior increases, the variance of guess and learn increases while the variance of prior decreases. In theory, a lesser variance on the prior prediction should imply a more accurate prior estimate. However, as we saw in Table 2, this is not actually the case. The prior estimate gets less accurate as the value of the ground truth prior increases. In Figure 3 we can see again some of the results we saw in Table 2: the prediction accuracy decreases when prior is 0.6 and continues to decrease as prior increases.

[1]Variations were 0.10 and 0.20 for learn and guess, and 0.05, 0.10 for slip.

Ground truth prior	mean RMSE	var RMSE
0.15	0.056639	0.000594
0.30	0.069073	0.001137
0.45	0.070005	0.000584
0.60	0.074044	0.001874
0.75	0.075946	0.002229
0.90	0.085257	0.004876

Table 2: The mean and variance of the root-mean-square errors per prior

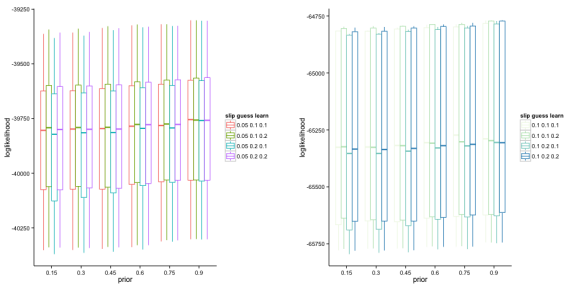


Figure 4: Log likelihoods with different parameters

Figure 4 shows that the log likelihood for each of the parameter combinations we analyzed. We see a slight, but non-significant increase in the log likelihoods, suggesting that the model is performing better – even while our RMSE error indicator demonstrates otherwise. It is also noteworthy to observe that that when slip is 0.10, all log likelihoods range between -65500 and -65250 but when slip is 0.05, all log likelihoods range between -40000 and -35750, indicating that the slip value had a dramatic effect on the model estimation accurancy.

5. IMPLICATIONS

Our findings indicate that there are higher errors in the parameter estimations when prior is high (0.90). This is probably due to the lack of evidence available for the HMM to attribute to the learn and guess parameters. One approach to examine the impact of these errors is to examine the students' subjective experience in different conditions [19]. As our data is syntetic, we can not measure the time consumed by students due to errors, as examined by Youdelson & Koedinger [19]. Instead we explore the difference on the number of questions students' need to answer to achieve mastery learning – for our purposes knowledge above 95 % and assuming that the students answer each question correctly.

Examining the case of high prior knowledge, and when the true learning was 0.1, we observed that majority of students needed to answer over 5 times to achieve mastery (or: from the 168 predicted value sets available, only 24 achieved mastery), and for the high learning (0.2) the situation was not

Table 3: Significant differences between the RMSEs

	0.15	0.30	0.45	0.60	0.75	0.90
0.15	1	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
0.30		1	0.347	0.614	0.967	0.014
0.45			1	0.660	0.125	0.081
0.60				1	0.744	0.035
0.75					1	0.007
0.90						1

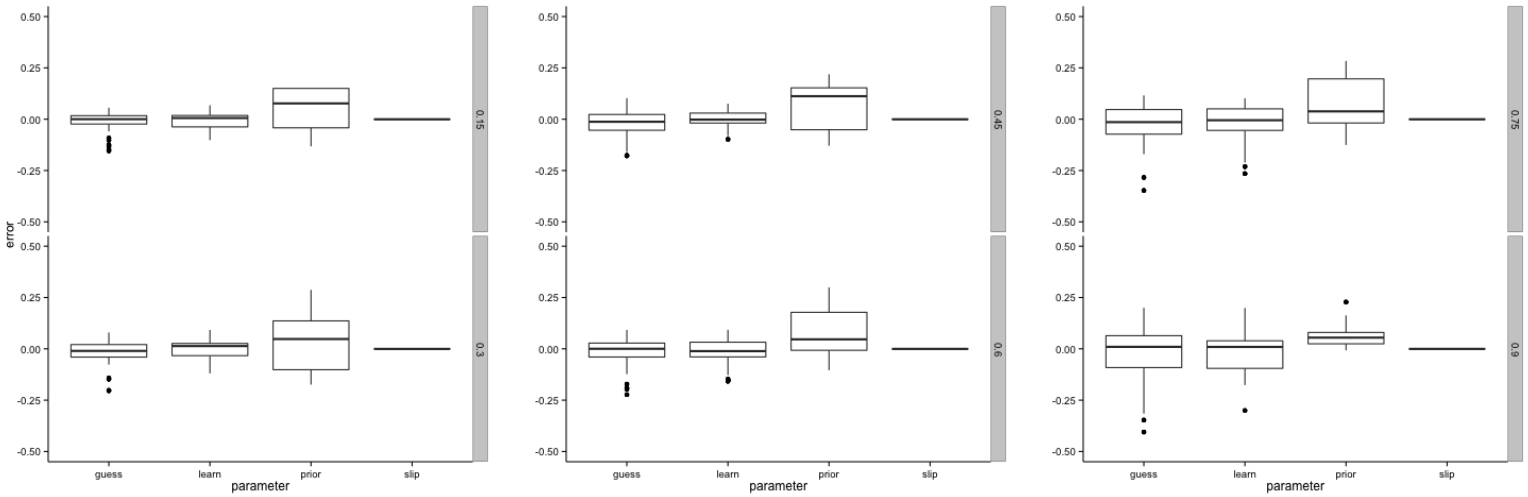


Figure 3: Predicting parameters with different values of prior

significantly better – there 56 values achieved mastery with 5 responses. This indicates that the impact indeed was significant in terms of impact to students learning and highlights the importance of this study.

6. CONCLUSIONS

We started this study with the motivation to explore how well the knowledge tracing method performs when the prior is high or low; this performance has practical implications when applying this approach in a heterogenius classroom where students arrive with highly different knowledge of the domain. We studied this empirically by generating 288 different synthetic datasets and explored the difference between the predicted parameters and the parameters used to generate the dataset.

Our results indicated a slightly increased in the estimation error when prior was 0.90, which we mostly attribute to higher error in learn and guess parameters. This observation was statistically significant and most likely due to the fact that students with higher priors produce less information to be used by the HMM to estimate the guess and learn parameters.

We explored the influence these errors had on the propability of knowledge and observed that these errors significantly reduced the speed students achieved mastery learning. This result therefore implies that more work needs to be done to detect those with high prior knowledge to cater their learning needs.

Acknowledgments

This work was conducted during UC Berkeley School of Information class “INFO290: Machine learning in education“ instructed by Zach Pardos. We thank the support of the course staff and peers on the presentation.

References

[1] RyanS.J.d. Baker, AlbertT. Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In BeverleyP. Woolf, Esma AÃŕmeur, Roger Nkambou, and Susanne Lajoie, editors, *Intelligent Tutoring Systems*, volume 5091 of *Lecture Notes in Computer Science*, pages 406–415. Springer Berlin Heidelberg, 2008.

[2] Joseph E Beck and Kai-min Chang. Identifiability : A Fundamental Problem of Student Modeling. pages 137–146, 2007. doi: 10.1007/978-3-540-73078-1_17.

[3] Albert Corbett. Cognitive computer tutors: Solving the two-sigma problem. In *User Modeling 2001*, volume 2109 of *Lecture Notes in Computer Science*, pages 137–147. Springer Berlin Heidelberg, 2001.

[4] Albert Corbett, Megan McLaughlin, and K Christine Scarpinatto. Modeling student knowledge: Cognitive tutors in high school and college. *User modeling and user-adapted interaction*, 10(2-3):81–108, 2000.

[5] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4): 253–278, 1994.

[6] Albert T Corbett and Akshat Bhatnagar. Student modeling in the act programming tutor: Adjusting a procedural learning model with declarative knowledge. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, pages 243–254, 1997.

[7] Tanja KÃďser, Severin Klingler, AlexanderGerhard Schwing, and Markus Gross. Beyond knowledge tracing: Modeling skill topologies with bayesian networks. In Stefan Trausan-Matu, KristyElizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, volume 8474 of *Lecture Notes in Computer Science*, pages 188–198. Springer International Publishing, 2014.

[8] Z. A. Pardos and N. T. Heffernan. Navigating the parameter space of Bayesian Knowledge Tracing models Visualizations of the convergence of the Expectation

Maximization algorithm. *In Proceedings of the 3rd International Conference on Educational Data Mining*, 2010.

[9] ZA Pardos and NT Heffernan. Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. *Journal of Machine Learning Research W & CP*, 2010. URL http://people.csail.mit.edu/zp/papers/pardos_JMLR_in_press.pdf.

[10] Z.A. Pardos, M.J. Johnson, and et al. Scaling cognitive modeling to massive open environments. *TOCHI Special Issue on Learning at Scale*, (in preparation).

[11] ZacharyA. Pardos and Neil T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In Paul Bra, Alfred Kobsa, and David Chin, editors, *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*, pages 255–266. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13469-2.

[12] Pardos, Zachary A, Sujith M. Gowda, Ryan S.J.d. Baker, and Neil T. Heffernan. The sum is greater than the parts. *ACM SIGKDD Explorations Newsletter*, 13(2):37, May 2012. ISSN 19310145. doi: 10.1145/2207243.2207249. URL http://dl.acm.org/citation.cfm?id=2207249 http://dl.acm.org/citation.cfm?doid=2207243.2207249.

[13] Dovan Rai, Yue Gong, and Joseph E Beck. Using dirichlet priors to improve model parameter plausibility. *International Working Group on Educational Data Mining*, 2009.

[14] Leena Razzaq, Neil T Heffernan, Mingyu Feng, and Zachary A Pardos. Developing Fine-Grained Transfer Models in the ASSISTment System. *Technology, Instruction, Cognition & Learning*, 5(3):1–16, 2007.

[15] Steven Ritter, Thomas K Harris, Tristan Nixon, Daniel Dickison, R Charles Murray, and Brendon Towle. Reducing the knowledge tracing space. *International Working Group on Educational Data Mining*, 2009.

[16] A Toscher and Michael Jahrer. Collaborative filtering applied to educational data mining. *Journal of Machine Learning Research*, 2010.

[17] Brett van De Sande. Properties of the Bayesian Knowledge Tracing Model. *Journal of Educational Data Mining*, 5(2):1–10, 2013.

[18] Hsiang-Fu Yu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, et al. Feature engineering and classifier ensemble for kdd cup 2010. *JMLR: Workshop and Conference Proceedings*, 1, 2010.

[19] Michael V Yudelson and Kenneth R Koedinger. Estimating the benefits of student model improvements on a substantive scale. In *Proceedings of the 6th International Conference on Educational Data Mining*, 2013.

[20] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer, 2013.

A Brief Overview of Metrics for Evaluation of Student Models

Radek Pelánek
Masaryk University Brno
pelanek@fi.muni.cz

ABSTRACT

Many different metrics are used to evaluate and compare performance of student models. The aim of this paper is to provide an overview of commonly used metrics, to discuss properties, advantages, and disadvantages of different metrics, and to summarize current practice in research papers. The paper should serve as a starting point for workshop discussion about the use of metrics in student modeling.

1. INTRODUCTION

A key part of intelligent tutoring systems are models that estimate the knowledge of students. To compare and improve these models we use metrics that measure quality of model predictions. Metrics are also used (sometimes implicitly) for parameter fitting, since many fitting procedures try to optimize parameters with respect to some metric.

At the moment there is no standard metric for model evaluation and thus researchers have to decide which metric to use. The choice of metric is an important step in the research process. Differences in predictions between competing models are often small and the choice of metric can influence the results more than the choice of a parameter fitting procedure. Moreover, fitted model parameters are often used in subsequent steps in educational data mining and thus the choice of metric can indirectly influence many other aspects of the research.

However, despite the fact that the choice of metric is important and that there is no clear consensus on the usage of performance metrics, the topic gets very little attention in most research papers. Most authors do not provide any rationale for their choice of metric. Sometimes it is not even clear what metric is exactly used, so it may be even difficult to use the same metric as previous authors. The main aim of this paper is to give an overview of performance metrics relevant for evaluation of student models and to explicitly discuss points that are in most papers omitted.

2. OVERVIEW OF METRICS

To attain clear focus we discuss only models that predict probability of a correct answer. We assume that we have data about n answers, numbered $i \in \{1, \ldots, n\}$, correctness of answers is given by $c_i \in \{0, 1\}$, a student models provides predictions $p_i \in [0, 1]$. A model performance metric is a function $f(\vec{p}, \vec{c})$. Note that the word "metric" is here used in a sense "any function that is used to make comparisons", not in the mathematical sense of a distance function. Since we are interested in using the metrics for comparison, monotone transformations (square root, logarithm, multiplication by constant) are inconsequential and are used mainly for better interpretability (or sometimes rather for traditional reasons).

2.1 Mean Absolute Error

This basic metric consider the absolute differences between predictions and answers: $MAE = \frac{1}{n} \sum_{i=1}^{n} |c_i - p_i|$. This is not a suitable performance metric, because it prefers models which are biased towards the majority results. As a simple illustration, consider a simulated student which answers correctly with constant probability 0.7. If we compare different constant predictors with respect to this metric, we get that the best model is the one which predicts probability of correct answer to be 1. This is clearly not a desirable result. As this example illustrates, the use of MAE can lead to rather misleading conclusions. Despite this clear disadvantage, MAE is sometimes used for evaluation (although mostly in combination with other metrics, which reduces the risk of misleading conclusions in published papers).

2.2 Root Mean Square Error

A similar metric is obtained by using squared values instead of absolute values: $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (c_i - p_i)^2}$. Note that from the perspective of model comparison, the important part is only the sum of square errors (SSE). The square root in RMSE is traditionally used to get the result in the same units of as the original "measurements" and thus to improve interpretability of the resulting number. In the particular context of student modeling and evaluation of probabilities, this is not particularly useful, since the resulting numbers are hard to interpret anyway. In order to get better interpretability researchers sometimes use R^2 metric: $R^2 = 1 - \sum_{i=1}^{n} (c_i - p_i)^2 / \sum_{i=1}^{n} (c_i - \bar{c})^2$. With respect to comparison of models, R^2 is equivalent to RMSE since here again the only model dependent part is the sum of square errors. In the context of the standard linear regression (where it is

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

most commonly used) R^2 has a nice interpretation as "explained variability". In the case of logistic regression (which is more similar to student models) this interpretation does not hold and different "pseudo R^2" metrics are used (e.g., Cox and Snell, McFadden, Nagelkerke). Thus a disadvantage of R^2 is that unless the authors are explicit about which version of R^2 they use (usually they are not), a reader cannot know for sure which metric is reported.

In educational data mining the use of RMSE metric is very common (it was also used as a metric in KDD Cup 2010 focused on student performance evaluation). In other areas, particularly in meteorology, mean square error (RMSE without the square root) is called the Brier score [1]. The Brier score is often decomposed into additive components (e.g., reliability and refinement) which provide further insight into the behaviour of the predictor. Moreover, in an analogy to AUC metric and ROC curve (described below), this metric can be interpreted as area under Brier curves. These methods may provide interesting inspirations for student modeling.

2.3 Metrics Based on Likelihood

The likelihood of data (the answers) given a model (predicted probabilities) is $L = \prod_{i=1}^{n} p_i^{c_i} \cdot (1-p_i)^{(1-c_i)}$. Since we are indifferent to monotonic transformations we typically work with the numerically more stable logarithm of the likelihood $LL = \sum_{i=1}^{n} c_i \log(p_i) + (1-c_i)\log(1-p_i)$. This metric can also be interpreted from information theoretic perspective as measure of data compression provided by a model [4]. The log-likelihood metric can be further extended into metrics like Akaike information criterion (AIC) and Bayesian information criterion (BIC). These metrics penalize large number of model parameters and thus aim to avoid overfitting. In the context of student modeling it is typically much better to address the issue of overfitting by cross-validation. Since AIC and BIC provide a faster way to assess models than cross-validation, they may be useful as heuristics in some algorithms (e.g., learning factor analysis), but they are not serious contenders for proper model comparison.

MAE, RMSE and LL have all the form of "sum of penalties for individual errors" and differ only in the function which specifies the penalty. For RMSE and LL values of penalty functions are quite similar, the main difference is in the interval [0.95, 1], i.e., in cases where the predictor is confident and wrong. These cases are penalized very prohibitively by LL, whereas RMSE is relatively benevolent. In fact the LL metric is unbounded, so single wrong prediction (if it is too confident) can ruin the performance of a model. This property is usually undesirable and an artificial bound is used. This corresponds to basically forcing a possibility of a slip and guess behaviour into a model. After this modification the penalties for RMSE and LL are rather similar. Nevertheless, the LL approach "penalize mainly predictions which are confident and wrong" is reasonable thus it is rather surprising that this metric is used only marginally in evaluation of student models (it is used mostly in connection with AIC or BIC).

2.4 Area Under an ROC Curve

Another popular metric is based on the receiver operating characteristics (ROC) curve. If we want to classify predictions into just two discrete classes (correct, incorrect), we need to select a threshold for the classification. For a fixed threshold we can compute standard metrics like precision, recall, and accuracy. If we do not want to use a fixed threshold, we can use the ROC curve, which summarises the behaviour of the prediction model over all possible thresholds. The curve has "false positive rate" on x-axis and "true positive rate" on the y-axis, each point of the curve corresponds to a choice of a threshold. Area under the ROC curve (AUC) provides a summary performance measure across all possible thresholds. It is equal to the probability that a randomly selected correct answer has higher predicted score than a randomly selected incorrect answer. The area under the curve can be approximated using a A' metric, which is equivalent to the well-studied Wilcoxon statistics [2]. This connection provides ways to study statistical significance of results (but requires attention to assumptions of the tests, e.g., independence).

The ROC curve and AUC metric are successfully used in many different research areas, but their use is sometimes also criticised [3], e.g., because the metric summarises performance over all possible thresholds, even over those for which the classifier would never be used in practice. From the perspective of student modeling the main reservation seems to be that this approach focuses on classification and considers predictions only in relative way – note that if all predictions are divided by 2, the AUC metric stays the same.

In the context of student modeling we are usually not interested in classification, we are often interested directly in absolute values of probabilities and we need these values to be properly calibrated. The probabilities are often compared to a fixed constant (typically 0.95) as an indication of a mastered skill and the specific value is meant to carry a certain meaning. Probabilistic estimates can be also used to guide the behaviour of a system to achieve suitable challenge for students, e.g., by choosing question of right difficulty or modifying difficulty by number of options in multiple choice questions.

Nevertheless, despite this disadvantage, AUC is widely used for evaluation of student models, often as the only metric. It seems that in some cases AUC is used as the only metric for final evaluation, but the parameter fitting procedure uses (implicitly) different metric (RMSE or LL). Particularly in cases of brute force fitting this approach seems strange and should be at least explicitly mentioned.

3. REFERENCES

[1] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[2] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. In *Proc. of Graphics Interface 2005*, pages 129–136, 2005.

[3] J. M. Lobo, A. Jiménez-Valverde, and R. Real. AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008.

[4] M. S. Roulston and L. A. Smith. Evaluating probabilistic forecasts using information theory. *Monthly Weather Review*, 130(6), 2002.

A Comparison of Error Metrics for Learning Model Parameters in Bayesian Knowledge Tracing *

Asif Dhanani† Seung Yeon Lee† Phitchaya Mangpo Phothilimthana† Zachary Pardos
University of California, Berkeley
{asifdhanani, sy.lee, mangpo, pardos}@berkeley.edu

ABSTRACT

In the knowledge-tracing model, error metrics are used to guide parameter estimation towards values that accurately represent students' dynamic cognitive state. We compare several metrics, including log-likelihood (LL), RMSE, and AUC, to evaluate which metric is most suited for this purpose. In order to examine the effectiveness of using each metric, we measure the correlations between the values calculated by each and the distances from the corresponding points to the ground truth. Additionally, we examine how each metric compares to the others. Our findings show that RMSE is significantly better than LL and AUC. With more knowledge of effective error metrics for learning parameters in the knowledge-tracing model, we hope that better parameter searching algorithms can be created.

1. INTRODUCTION

In Bayesian Knowledge Tracing (BKT), one of the essential elements is the error metric that is used for learning model parameters: prior, learn, guess, and slip. Choice of a type of error metric is crucial because the error metric takes a role of guiding the search to the best parameters. The BKT model can be fit to student performance data by using a method which finds a best value calculated from the error metric such as log-likelihood (LL), root-mean-squared error (RMSE), or area under the ROC curve (AUC).

As a modeling method, grid search/brute force [1] is often used to find the set of parameters with optimal values of the error metric, and Expectation Maximization (EM) algorithm [5] is also commonly used to choose parameters maximizing the LL fit to the data. Many studies have compared different modeling approaches [1, 4]. However, the findings are varied across the studies, and it has still been unclear which method is the best at predicting student performance [2].

Pardos and Yudelson compares different error metrics to investigate which one has the most accuracy of estimating the moment of learning [6]. Our work extends this comparison by looking closer into the relationship between three popular error metrics: LL, RMSE, and AUC, and particularly elucidating the relationship to one another closer to the ground truth point.

*For more details of this work, please refer to the full technical report [3].

†Asif Dhanani, Seung Yeon Lee, and Phitchaya Mangpo Phothilimthana contributed equally to this work and are listed alphabetically.

2. METHODOLOGY

To assess whether LL, RMSE, or AUC is the best error metric to use in parameter searching for the BKT model, we needed datasets with known parameter values in order to compare these with the parameter values predicted by using different error metrics. Therefore, we synthesized 26 datasets by simulating student responses based on diverse known ground truth parameter values.

Correlations to the ground truth. For each dataset, we evaluated LL, RMSE, and AUC values on all points over the entire prior/learn/guess/slip parameter space with a 0.05 interval. On each point, we calculated students' predicted responses (probability that students will answer questions correctly). We then used these predicted responses with the actual responses to calculate LL, RMSE, and AUC for all points. To determine which error metric is the best for this purpose, we looked at the correlations between values calculated from error metrics (i.e. LL, RMSE, and AUC) and the euclidean distances from the points to the ground truth. We applied logarithm to all error metrics other than LL in order to compare everything on the same scale. Finally, we tested whether the correlation between the values calculated by any particular error metric and the distances is significantly stronger than the others' by running one-tailed paired t-tests comparing all three metrics against one another.

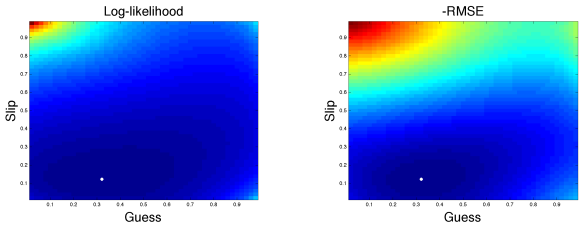
Distributions of values. We visualized the values of LL and -RMSE of all points over the 2 dimensional guess/slip space with a 0.02 interval while fixing prior and learn parameter values to the actual ground truth values. Using the guess and slip parameters as the axes, we visualize LL and -RMSE values by color. The colors range from dark red to dark blue corresponding to the values ranging from low to high.

Direct comparison: LL and RMSE. We plotted LL values and RMSE values of all points against each other in order to observe the behavior of the two metrics in detail. We then labeled each data point by its distance to the ground truth with a color. The range of colors is the same as used in the previous method.

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Comparision	Δ of correlations	t	p-value
RMSE > LL	0.0408	8.9900	<< 0.0001
RMSE > AUC	0.0844	2.7583	0.0054
LL > AUC	0.0436	1.4511	0.0796

Figure 1: T-test statistics



(a) LL Heatmap (b) -RMSE Heatmap

Figure 2: LL and -RMSE values when fixing prior and learn parameter values and varying guess and slip parameter values. Red represents low values, while blue represents high values. The white dots represent the ground truth.

3. RESULTS

Correlations to the ground truth. The average LL, RMSE, and AUC correlations were 0.4419, 0.4827, and 0.3983 respectively. We define that an error metric A is *better* than B if the correlation between values calculated by an error metric A and the distances to the ground truth is higher than that of B. By this definition, RMSE was better than LL on all 26 datasets and better than AUC on 18 of 26 datasets. This is validated by the one-tailed paired t-test shown in Figure 1 revealing RMSE as statistically significantly better than both LL and AUC.

Distributions of values. Figure 2 shows the heat maps of LL and RMSE on a representative dataset. If we follow the gradient from the lowest value to the highest value in the LL heat map, we see that it is very high at the beginning (far from the ground truth) and is very low at the end (close to the ground truth). Conversely, in the -RMSE heat map, the change in the gradient is low. Additionally, notice that the darkest blue region in -RMSE heat map is smaller than that in LL heat map. This suggests that we may be able to refine the proximity of the ground truth better with RMSE.

Direct comparison: LL and RMSE. Figure 3 shows a LL vs -RMSE graph from the most representative dataset. As expected, LL values and RMSE values correlate logarithmically. Additionally, a secondary curve, which we will refer to as the *hook*, is observed in varying sizes among datasets. The hook converges with the main curve when the -RMSE and LL values are both sufficiently high and the points are very close to the ground truth.

Before this point, when we look at a fixed LL value with varied RMSE values, most points in the hook have higher -RMSE values and are closer to the ground truth than do the points in the main curve. However, this same pattern is not seen for a fixed RMSE value with varied LL values. After the curve and hook converge, we can infer that both RMSE and LL will give similar estimates of the ground truth. However, for a portion of the graph before this point, RMSE is a better predictor of ground truth values.

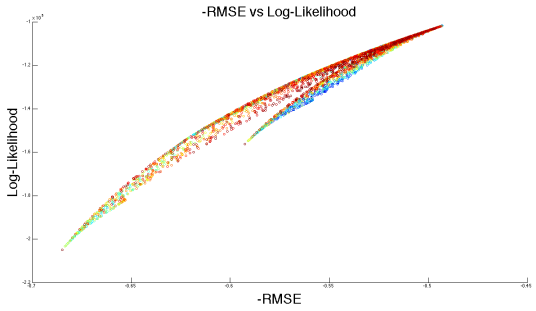


Figure 3: LL vs -RMSE of dataset 25 when prior = 0.564, learn = 0.8, guess = 0.35 , and slip = 0.4

4. CONCLUSION

In our comparison of LL, RMSE, and AUC as metrics for evaluating the closeness of estimated parameters to the true parameters in the knowledge tracing model, we discovered that RMSE serves as the strongest indicator. RMSE has a significantly higher correlation to the distance from the ground truth on average than both LL and AUC, and RMSE is notably better when the estimated parameter value is not very close to the ground truth. The effectiveness of teaching systems without human supervision relies on the ability of the systems to predict the implicit knowledge states of students. We hope that our work can help advance the parameter learning algorithms used in the knowledge tracing model, which in turn can make these teaching systems more effective.

5. REFERENCES

[1] R. Baker, A. Corbett, S. Gowda, A. Wagner, B. MacLaren, L. Kauffman, A. Mitchell, and S. Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *User Modeling, Adaptation, and Personalization*, volume 6075 of *Lecture Notes in Computer Science*. 2010.

[2] R. S. Baker, Z. A. Pardos, S. M. Gowda, B. B. Nooraei, and N. T. Heffernan. Ensembling predictions of student knowledge within intelligent tutoring systems. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, 2011.

[3] A. Dhanani, S. Y. Lee, P. Phothilimthana, and Z. Pardos. A comparison of error metrics for learning model parameters in bayesian knowledge tracing. Technical Report UCB/EECS-2014-131, EECS Department, University of California, Berkeley, May 2014.

[4] Y. Gong, J. Beck, and N. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, volume 6094 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010.

[5] Z. Pardos and N. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *User Modeling, Adaptation, and Personalization*. 2010.

[6] Z. A. Pardos and M. V. Yudelson. Towards moment of learning accuracy. In *Proceedings of the 1st AIED Workshop on Simulated Learners*, 2013.

Prediction of Student Success Using Enrolment Data

Nihat Cengiz
Epoka University
Department of Computer Engineering
Rr. Tiranë-Rinas,Km. 12
1039 Tirana, Albania
ncengiz@epoka.edu.al

Arban Uka
Epoka University
Department of Computer Engineering
Rr. Tiranë-Rinas,Km. 12
1039 Tirana, Albania
auka@epoka.edu.al

ABSTRACT
Predicting the success of students as a function of different predictors has been a topic that has been investigated over the years. This paper explores the socio-demographic variables like gender, region lived and studied, nationality and high school degree that may influence success of students. We examine to what extent these factors help us to predict students' academic achievement and will help to identify the vulnerable students and their need for extra tutoring or similar supportive services at an early time.

We analyzed the data of the Epoka University students that have been enrolled from 2007 to 2013. The sample includes 1211 undergraduate students where 716 did and were supposed to complete the three-year bachelor studies in the past six semesters.

Based on the data mining techniques the most important predictors for student success were the students' high school GPA and gender. For students with high school grades below average, females were found to have a higher percentage of success than boys. No significant correlation was found between the students' success and the demographic information.

Keywords
Academic achievement, influence, classification tree, outcome

1. INTRODUCTION
Increasing the student graduation and decreasing the dropout rates is a long term goal of the higher education institutions. From the students' perspective, a timely and successful graduation is vital as these two factors would strongly affect their employability rate. Employability rate has become an indicator in determining the ranking of higher education institution (HEI), thus HEIs are focusing more on increasing this rate [2].

Many of the students studying at the university face several difficulties during the first year and thus the performance of the first year has been identified as an important predictor of timely graduation rate. In terms of keeping the students in the university, the retention rate is a factor that has been studied extensively. Mallincrodt and Sedlacek (1987) found that freshman class attrition rate were greater than the other academic years with numbers running up to 30%.[3] Therefore most researchers targeted the first year students. An early identification of the students at high risk of failing will enable a timely intervention with the necessary measures by the educators that would increase the graduation rate. Preventing students' failure depends on the identification of the factors affecting success.

Here in this work we will analyze whether the background information has any effect on the success rate of regular students. The only data we collected during the registration period of Epoka University based on the registration form. The content of this form determined by the local authorities and University Administration. In this study we tried to get answers if we can use this data to predict student success. The main objective of our study is to determine the factors that may affect the study outcomes in Epoka University.

2. DATA AND METHODOLOGY
Epoka University student management system does not provide data in the format ready for a direct statistical analysis and modeling. Therefore a data preparation and cleaning were undertaken to prepare database for modeling.

Table Descriptive statistics – Study outcome (716 students)

Descriptive						
		count		%		
	Domain	FAIL	PASS	FAIL	PASS	Total
GENDER	M	221	189	53.9	46.1	57.3
	F	78	228	25.5	74.5	42.7
COUNTRY	ALB	238	372	39.0	61.0	85.2
	TUR	35	14	71.4	28.6	6.8
	KOS	14	17	45.2	54.8	4.3
	OTH	12	14	46.2	53.8	3.6
NATIONALITY	ALB	256	382	40.1	59.9	89.1
	OTH	43	35	55.1	44.9	10.9
REGION	CITY	262	372	41.3	58.7	88.5
	VILL.	37	44	45.7	54.3	11.3
HS_GPA	UPPER	48	224	17.6	82.4	38.0
	INTER.	89	113	44.1	55.9	28.2
	LOWER	160	77	67.5	32.5	33.1

2.1. Data and Methodology
Outcome that we used in our analysis is for the outcome of the student at the end of three-year study. We measured only outcomes, labeled as: Pass and Fail. Students labeled 'Pass' successfully completed the program at the end of three years. Students labeled as 'Fail' include the withdrawn students from the

program voluntarily or by the academic registry for not fulfilling the regulations. Those students who stayed on the program until the end of the study but scored less than the graduation grade (2.00) were also allocated into this category.

The data set with numeric continuous variable such as secondary school grade (HS GPA) was converted into a categorical variable with only three levels A (UPPER), B (INTERMEDIATE) or C (LOWER) denoting grades above 9 out of 10, grades between 8 and 9 and grades less than 8 respectively. Other variables (nationality, citizenship, and region) were classified upon major groups.

In this study we conducted three main types of data mining approaches. Descriptive approach which concerns the nature of the dataset such as the frequency table and the relationship between the attributes obtained using cross tabulation analysis. Predictive approach which is conducted by using four different classification trees and a comparison between these and Logistic regression to confirm the accuracy of the predictors.

Classification tree models can handle a large number of predictor variables, are non-parametric, can capture nonlinear relationships and complex interactions between predictors and dependent variable.[1]

Before generating the classification trees we classified the variables according to the study outcome, i.e. whether students are eligible to be graduated or not. We used attribute selection to rank the variables by their importance for further analysis. Then we generated the classification trees in four different growing methods.

2.2. Summary Data Description

We carried out a cross-tabulation for each variable and the study outcome after cleaning the data as shown in the table above. Table shows that the majority of the successful students are female (over 57%) which is the result of the fact that 74.5% of the female students successfully completed the study. This suggests that female students are more likely to succeed than their male classmates. In terms of country and nationality it is clearly seen that Albanian population is leading the group.

An expected result has been observed in secondary school degrees. We can say that high school degree graduation ratio is directly proportional to the university graduation ratio. While 82% of upper students were able to complete the study on time 56% of intermediate and 32% of lower group students were able to complete.

2.3. Decision Trees

Although the results of the attribute selection suggests continuing analysis with only the subset of predictors, we included all available predictors in our classification trees but only 2 variables were used in the diagrams: HS_GPA and GENDER. Even though some variables may have little significance to the overall prediction outcome, they can be essential to a specific record [1].

Almost all growing methods, (CHAID, exhaustive CHAID, CRT and QUEST) generated exactly the same trees. The largest successful group consists of 272 (38%) students. HS_GPA of this group is over 90%. The largest unsuccessful group contains 237 students (33% of all participants). They have a HS_GPA less than 80%. The next largest group considered also as unsuccessful students are male students having lower HS_GPA.

As the cross-validation estimate of the risk (0.309) indicates that the successful or unsuccessful students are predicted with an error of 30.9% of the cases which means the risk of misclassifying a student is approximately 31%. This result is consistent with the results in the CHAID classification matrix. The Overall percentage shows that the model only classified correctly 70% of students. The classification tables, however, reveal one potential problem with this model: for unsuccessful students, it predicts as successful for only 65.9% of them, which means that 34% of failing students are inaccurately classified with the passing students.

2.4. Logistic regression

The Variables not in the Equation table in block 0 shows that four of the five variables are individually significant predictors of whether a student is successful or not. Region is not a significant predictor. The variables not in the Equation table in block 1 shows that only high school grade point average and gender are significant predictors, but not the other variables. This result also confirms why these two were the only variables used in decision trees

3. CONCLUSIONS

This study examines the background information from enrolment data that impacts upon the study outcome programs at the Epoka University. Based on results, the classification accuracy from the classification trees was significantly high 71% in all tree methods. Although all the variables except the region individually significant predictors as described in attribute selection trees displayed only two variables Gender and secondary school degree. This outcome is also confirmed by the logistic regression. Block 0 classification implied that all except region were good predictors (p<,001) but block 1 classification highlighted that only gender and secondary school degree were significant.

4. REFERENCES

[1]. Kovačić, Z.J. 2010, Early Prediction of Student Success: Mining Students Enrolment Data, proceedings of Informing Science & IT Education Conference (InSITE) 2010, Open Polytechnic, Wellington, New Zealand

[2]. Bratti, M., McKnight, A., Naylor, R., & Smith, J. (2004): Higher Education Out-comes, Graduate Employment and University Performance Indicators. In: Journal of the Royal Statistical Society, 167(3), pp 475-496.

[3]. Mallinckrodt, B., & Sedlacek, W. E. (1987). Student retention and the use of campus facilities by race. NASPA Journal, 24, 28-32.

Expanding Knowledge Tracing to Prediction of Gaming Behaviors

Sarah E Schultz
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA
seschultz@wpi.edu

Ivon Arroyo
Worcester Polytechnic Institute
100 Institute Rd
Worcester, MA
iarroyo@wpi.edu

ABSTRACT
Knowledge tracing has been used to predict students' knowledge and performance for almost twenty years. Recently, researchers have become interested in looking at students' behaviors, especially those considered gaming behaviors. In this work, we attempt to leverage a variation of knowledge tracing to predict gaming behaviors without damaging the prediction of performance. We compare the predictions of this model to those of knowledge tracing and a separate engagement tracing model.

Keywords
Knowledge tracing, affect, engagement, gaming, behavior

1. INTRODUCTION
When Corbett and Anderson first published the knowledge tracing model in 1995, they claimed that their goal was "to implement a simple student modeling process that would allow the tutor to […] tailor the sequence of practice exercises to the student's needs" [1]. While knowledge tracing is generally able to predict students' performance "quite well," it does not take into account the possibility of disengagement. Traditionally, knowledge tracing is used with the probability of transition from a learned to an unlearned state set at 0, so students who become disengaged are not presumed to be forgetting the skill. When the forgetting transition is allowed, models such as knowledge tracing can become confounded, mistaking disengagement for unlearning, as illustrated in Figure 1.

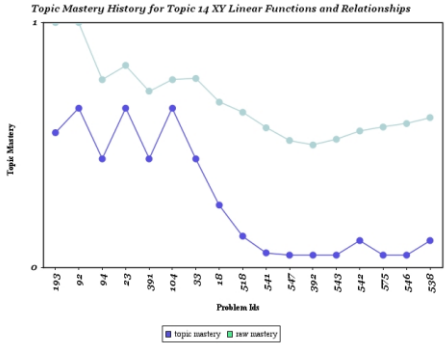


Figure 1- Bayesian Knowledge Estimation of a student on one skill (bottom line)

Figure 1 suggests that this student was un-learning, while after looking at the logs in detail, it was clear that, after the 7th problem, the student was just clicking through all the available multiple-choice answers without attempting to answer correctly.This type of behavior is defined by Baker et al as "gaming the system" [2] and is considered to be an indicator of disengagement or negative affect. Some work has been done in modeling engagement and affect in Intelligent Tutoring Systems [3], but relatively little research has focused on combining these methods with ways of tracking knowledge, such as knowledge tracing, in order to create a model that can predict both student performance and disengaged behavior and intervene appropriately.

2. PREVIOUS WORK

2.1 Bayesian Knowledge Tracing
Corbett and Anderson's Bayesian Knowledge Tracing (BKT) [1] (Figure 2) is a hidden Markov model. At each time step there is a latent node, knowledge, and an observed node, performance. The parameters for this model are $P(L_0)$, the probability that a student already knows the skill; P(T), the probability of learning the skill from one time-step to the next; P(G), the probability that a student who does not know the skill but correctly guesses; and P(S), the probability that a student who does know the skill slips and gets the answer incorrect. As mentioned in the introduction, P(F), forgetting, is traditionally set at 0, however for this work we allow forgetting in order to see if looking at behavior affects the amount of forgetting that students appear to do.

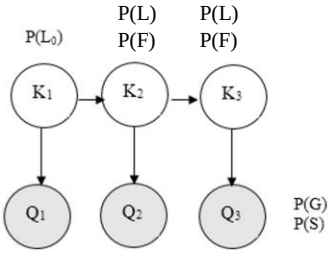


Figure 2- Bayesian Knowledge Tracing

2.2 HMM-IRT
In 2006, Johns and Woolf proposed the Dynamic Mixture Model (DMM) for predicting student knowledge and engagement [4]. They used a hidden Markov model like BKT for tracing engagement, but paired it with an Item Response Theory-like model for predicting knowledge. Rather than predicting knowledge at each time step, there is a single knowledge node for every skill and students' performance relies on that in addition to their engagement state. This allowed more accurate knowledge predictions than IRT alone, as disengagement, indicated by gaming behaviors, could explain away some incorrect attempts, rather than attributing those to knowledge.

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

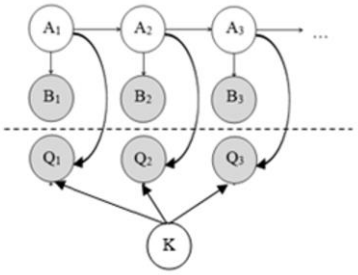


Figure 3- Dynamic Mixture Model

2.3 The KAT Model

In our previous work [5], we proposed the knowledge and affect tracing (KAT) model (Figure 5), which combines two hidden Markov models, BKT and the engagement tracing piece of DMM. As in DMM, affect influences performance. This model was able to predict both performance and behavior better than the dynamic mixture model, but did not predict performance as well as standard BKT, perhaps due to over-parameterization [5].

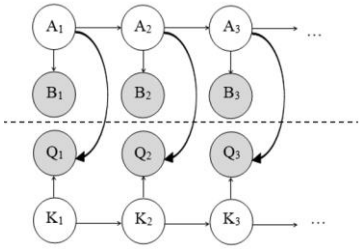


Figure 4- The KAT Model

3. THE KTB MODEL

We propose the "Knowledge Tracing with Behavior" (KTB) model. This model has only one latent node, which we call "knowledge"-- although in reality is a combination of both knowledge and engagement-- and two observables, performance and gaming behaviors. This model is shown in Figure 5.

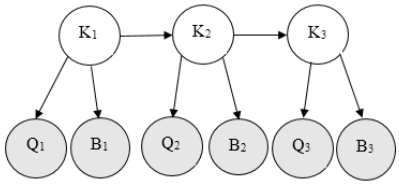


Figure 5- KTB Model

This model has fewer parameters than the dynamic mixture model or KAT model, but still can predict both performance and disengaged behavior of the students.

The variable called Gaming Behavior (B) is defined as either gaming or normal. See our definition for "gaming" in this context in our previous work [5].

4. BAYESIAN ENGAGEMENT TRACING

Since the performance prediction of the KTB model can be compared to that of Bayesian Knowledge Tracing, it is necessary to have a model of engagement tracing to compare the behavior predictions. To that end, we include a model of "Bayesian Engagement Tracing" (BET) in this work, which is the same as the HMM part of Johns and Woolf's model or the engagement piece of the KAT model, but not connected to any other model (top part of figure 4).

5. DATASETS AND METHODS

The data and methods used in this work was the same as that used in [5]. The data came from two tutors for middle and high school mathematics, ASSISTments and Wayang Outpost. For details, please see [5] in the main conference proceedings.

6. RESULTS AND ANALYSIS

While KT and KTB both outperform KAT and DMM in all predictions, in seven of the nine knowledge components, KTB was better able to predict performance than standard knowledge tracing, although the only significant difference between the two was in the ASSISTments skill "Circle Graph" ($p=0.03$). Interestingly, the Bayesian engagement tracing model was better able to predict students' behavior than KTB in eight of the nine knowledge components, although the differences are again not significant, except in two cases, "Box and Whisker," and "Triangles" ($p=0.02$).

7. DISCUSSION

We have proposed a new model, knowledge tracing with behavior, which can predict both student performance and behavior, and have shown that it can do so at least as well as BKT and a separate Bayesian engagement tracing, at predicting future behaviors (correctness at responding math problems and gaming behaviors). KTB seems to stop the false forgetting effect that is recorded by KT when forgetting is not allowed to be zero.

ACKNOWLEDGEMENTS

This research is supported by the Office of Naval Research, STEM Challenge Award, # N0001413C0127US. We also acknowledge funding from NSF (#1316736, 1252297, 1109483, 1031398, 0742503), and IES (# R305A120125 & R305C100024). Any opinions or conclusions expressed are those of the authors, not necessarily of the funders.

REFERENCES

[1] Corbett, A.T., Anderson, J.R., "Knowledge tracing: Modeling the acquisition of procedural knowledge." *User Modeling and User-Adapted Interaction*, 1995, 4, p.253-278.

[2] Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System". In *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.

[3] Beck, J.E. "Engagement tracing: using response times to model student disengagement." *Proceedings of AIED conference*, 2005. p. 88-95. IOS Press

[4] Johns, J. and Woolf, B.P. "A Dynamic Mixture Model to Detect Student Motivation and Proficiency." *Proceedings of AAAI Conference*, 2006, 1, p. 163-168.

[5] Schultz, S. and Arroyo, I. "Tracing Knowledge and Engagement in Parallel in an Intelligent Tutoring System." To appear in *Proceedings of the 7th Annual International Conference on Educational Data Mining*, 2014

Evaluating Student Models

Adaeze Nwaigwe
University of Maryland University College
3501 Unversity Blvd East
Adelphi, MD 207831 412 608 8747
adaeze.nwaigwe@faculty.umuc.edu

ABSTRACT
We use the Additive Factors Model to drive the evaluation of the student model of an Intelligent Tutoring System. Using data from the Andes Physics Tutor, applying the simple location heuristic and implementing the Additive Factors Model tool in the Pittsburgh's Science of Learning Center's DataShop, we discover possible ways to improve the student model of the Andes Intelligent Tutor.

Keywords
Student modeling, learning curves, additive factors model.

1. INTRODUCTION
The quality of student models drive many of the instructional decisions that automated tutoring systems make, whether it is what feedback to provide, when and how to sequence topics and problems in a curriculum, how to adapt pacing to the needs of students and even what problems and instructional materials are necessary [1]. We used the Additive Factors Model (AFM) tool in the Pittsburgh's Science of Learning Center's (PSLC) DataShop to identify areas for improvement in the curriculum for the ANDES Intelligent Tutoring System.

1.1 BACKGROUND
Learning curves derived from student models drive evaluation, revision and improvement of the Intelligent Tutor. The AFM is a statistical algorithm which models learning and performance by using logistical regression performed over the "error rate" learning curve data [1]. If a student is learning the knowledge component (KC) or skill being measured, the learning curve is expected to follow a so-called "power law of practice" [2]. If such a curve exists, it presents evidence that the student is learning the skill being measured or conversely, that the skill represents what the student is learning.

While use of learning curves is now a standard technique for assessing the cognitive models of Intelligent Tutors, the technique requires that a method is instated for attributing blame to skills or KCs. This simply means that each error a student makes must be blamed on a skill or set of skills. Four different heuristics for error attribution have been proposed and tested. These heuristics are guided by whether the method is driven by location – the simple location heuristic (LH), the model-based location heuristic (MLH); or by the temporal order of events – the temporal heuristic (TH), the model-based temporal heuristic (MTH); and whether the choice of the student model is leveraged (MLH, MTH) [3].

2 EVALUATING THE STUDENT MODEL

2.1Adapting the Andes Log data for the AFM Algorithm
The log data used for this work was obtained from the Andes Intelligent Tutor [4] and encompassed four problems in the area of electric field, across 102 students. The data was collected in Spring 2005 at the US Naval Academy during its regular physics class and as part of the PSLC's LearnLab facility that provides researchers, access to run experiments in or perform secondary analyzes of data collected from one of seven available technology-enhanced courses running at multiple high school and college sites (see http://learnlab.org).

Prior to using the AFM tool on the dataset, the simple location heuristic (LH) was applied to error transactions in the Andes log data which had missing KCs. That is, when the Andes failed to assign blame to a KC on an error transaction, the LH will select the first correctly implanted KC in the same location as the error. The LH was applied to about 44% of the original data. Table 1 depicts a summary of the LH data.

2.2 Generating Model Values using AFM
The Datashop's AFM algorithm was used to compute statistical measures of goodness of fit for the model - Akaike Information Criterion (AIC) and Bayesian Information criterion (BIC), as well as to generate learning curves for the Andes log data.

3 RESULTS AND DISCUSSION
We found that there were 5 groups of KCs – "Low and Flat", "No learning", "Still high", "Too Little data" and "Good". The "Low and Flat" group indicated KCs where students likely received too much practice. It appears that although students mastered the KCs they continued to receive tasks for them. It may be better to reduce the required number of tasks or change Andes' knowledge tracing parameters so that students get fewer opportunities with these KCs. The "Still high" group suggests KCs, which students continued to struggle with. Increasing opportunities for practice for these KCs might improve the student model. The "No learning" group indicated KCs where the slope of the predicted learning curve showed no apparent learning. A step towards improving the student model could be to explore whether each of these KCs can be split into multiple KCs. The new KCs may better reflect the variation in difficulty and transfer of learning that may be happening across problem steps, which are currently labeled by each KC. The KCs in the "Too Little data" group seem to be KCs for which students were exposed to insufficient practice opportunities for the data to be meaningful. For these KCs, adding

Published in CEUR-WS:
BKT20y workshop (Yudelson, González-Brenes and Mozer)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

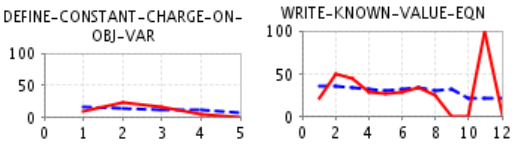
more tasks or merging similar KCs might provide data that is interpretable. The KCs that appeared "Good" may reflect those in which there was substantial student learning. Table 2 shows the different group of KCs, their frequencies and AIC and BIC scores. Figures 1a – 1d show the different groups of KCs. Intercept (logit) and intercept (probability) both indicate KC difficulty. Higher intercept values indicate more difficult KCs. The slope parameter indicates the KC learning rate. Higher values suggest students will learn such KCs faster.

Table 1. LH Data Summary

Number of Students	102
Number of Unique Steps	125
Total Number of Steps	5,857
Total Number of Transactions	71,300
Total Student Hours	107.02
# of Knowledge Component Model	34

Table 2. KC Groups and Statistical Scores

Low and Flat	No Learning	Still High	Too Little data	Good
2	2	4	24	2
# of Knowledge Components				34
AIC				6532.75
BIC				7668.14



KC Name	Intercept (logit)	Intercept (probability)	Slope
define-constant-charge-on-obj-var	1.77	0.85	0.120
write-known-value-eqn	0.63	0.65	0.037

Figure 1a – "Good"

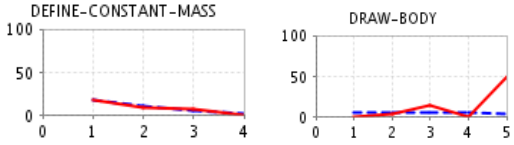
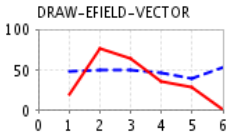
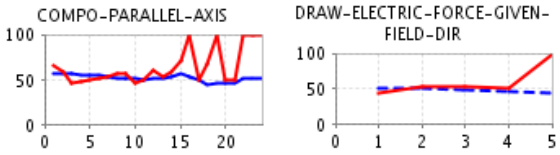


Figure 1b – "Low and Flat"



KC Name	Intercept (logit)	Intercept (probability)	Slope
draw-efield-vector	0.06	0.52	0.000

Figure 1c – "No Learning"



KC Name	Intercept (logit)	Intercept (probability)	Slope
compo-parallel-axis	-0.28	0.43	0.000
draw-electric-force-given-field-dir	-0.01	0.50	0.000

Figure 1d – "Still High"

4 CONCLUSION AND FUTURE WORK

This paper presented how the AFM can be used to evaluate the student model of the Andes Physics Tutor. Refining four of the five groups of KCs identified, might improve the Andes student model. A further approach would to use Learning Factors Analysis [1] algorithm to automatically find better student models by searching through a space of KC models. The next step is to explore these options and measure their effect.

5 ACKNOWLEDGMENTS

Our thanks to the Pittsburgh Science of Learning Center for providing the analysis tool for this work, to Bob Hausmann and Kurt VanLehn for dataset access.

6 REFERENCES

[1] Koedinger, K.R., McLaughlin, E.A., Stamper, J.C. 2012 Automated Student Model Improvement. Proceedings of the 5th International Conference on Educational Data Mining, Chania, Greece, pp. 17–24.

[2] Mathan S. & Koedinger K. 2005. Fostering the Intelligent Novice: Learning From Errors With Metacognitive Tutoring. Educational Psychologist. 40(4), pps. 257–265.

[3] Nwaigwe, A. & Koedinger, K.R. 2011. The Simple Location Heuristic is Better at Predicting Students' Changes in Error Rate Over Time Compared to the Simple Temporal Heuristic. Proceedings of the 4th International Conference on Educational Data Mining. Eindhoven, Netherlands.

[4] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., et al. 2005. The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.

Workshop on Feedback from Multimodal Interactions in Learning Management Systems (FFMI)

Virtually all learning management systems and tutoring systems provide feedback to learners based on their time spent within the system, the number, intensity and type of tasks worked on and past performance with these tasks and corresponding skills. Some systems even use this information to steer the learning process by interventions such as recommending specific next tasks to work on, providing hints etc. Often the analysis of learner / system interactions is limited to these high-level interactions, and does not make good use of all the information available in much richer interaction types such speech and video. In the workshop Feedback from Multimodal Interactions in Learning Management Systems (FFMI@EDM'2014) we wanted to bring together researchers and practitioners who are interested in developing data-driven feedback and intervention mechanisms based on rich, multimodal interactions of learners within learning management systems, and among learners providing mutual advice and help. We aim at discussing all stages of the process, starting from preprocessing raw sensor data, automatic recognition of affective states to learning to identify salient features in these interactions that provide useful cues to steer feedback and intervention strategies and leading to adaptive and personalized learning management systems. The contributions presented in this workshop range from work about affect recognition in intelligent tutoring systems to research questions from online learning and collaborative learning.

We gratefully acknowledge the following members of the workshop program committee:

Carles Sierra, IIIA, Spanish Research Council, University of Technology, Sydney
Arvid Kappas, School of Humanities and Social Sciences, Jacobs University Bremen, Germany
Emanuele Ruffaldi, PERCRO, Scuola Superiore Sant'Anna, Pisa, Italy
Sergio Gutierrez-Santos, Birkbeck, University of London, UK
Mark d'Inverno, Goldsmiths, University of London, UK
Manolis Mavrikis, IOE, University of London, UK
Francois Pachet, Sony Computer Science Laboratory Paris, France
Matthew Yee-King, Goldsmiths, University of London, UK
Helen Hastie, Heriot Watt University, Edinburgh, Scotland
Iolanda Leite, Yale University, Connecticut, United States
Luis de-la-Fuente, International University of La Rioja, Spain
Helen Pain, ILCC, Human Communication Research Centre, University of Edinburgh

The FFMI workshop organizers
Lars Schmidt-Thieme
Ruth Janning

Table of Contents FFMI

Interventions during student multimodal learning activities: which, and why?

Beate Grawemeyer
London Knowledge Lab
Birkbeck College
University of London, UK
beate@dcs.bbk.ac.uk

Manolis Mavrikis
London Knowledge Lab
Institute of Education
University of London, UK
m.mavrikis@ioe.ac.uk

Sergio Gutierrez-Santos
London Knowledge Lab
Birkbeck College
University of London, UK
sergut@dcs.bbk.ac.uk

Alice Hansen
London Knowledge Lab
Institute of Education
University of London, UK
a.hansen@ioe.ac.uk

ABSTRACT

Emotions play a significant role in students' learning behaviour. Positive emotions can enhance learning, whilst negative emotions can inhibit it. This paper describes a Wizard-of-Oz (WoZ) study which investigates the potential of Automatic Speech Recognition (ASR) together with an emotion detector able to classify emotions from speech to support young children in their exploration and reflection whilst working with interactive learning environments. We describe a unique ecologically valid WoZ study in a classroom. During the study the wizards provided support using a script, and followed an iterative methodology which limited their capacity to communicate, in order to simulate the real system we are developing. Our results indicate that there is an effect of emotions on the acceptance of feedback. Additionally, certain types of feedback are more effective than others for particular emotions.

Keywords

Affect, emotions, intelligent support

1. INTRODUCTION

Our aim is to build a learning platform for elementary education which integrates speech recognition for children in order to enable natural communication. This paper reports from on a Wizard-of-Oz study which explores the effect of emotions deduced from speech on different feedback types.

The importance of language as both a psychological and cultural tool that mediates learning has long been recognised; from as early as Vygotsky to modern linguists such as Pinker. From a Human Computer Interaction (HCI) perspective, speech recognition technology has the potential to enable more intuitive interaction with a system, particularly for young learners who reportedly talk aloud while engaged in problem solving (e.g. [11]).

Finally, speech provides an additional cue for drawing inferences on students' emotions and attitude towards the learning situation while they are solving tasks. By paying attention to tone and pitch of speech in conjunction with other auditory signs like sighs, gasps etc., we can provide learners with even more individualized help, by detecting emotions and providing support specifically tailored to the emotional state.

As described in [15] emotions interact with and influence the learning process. While positive emotions such as awe, satisfaction or curiosity contribute towards constructive learning, negative ones including frustration or disillusionment at realising misconceptions can lead to challenges in learning. The learning process includes a range and combination of positive and negative emotions. For example, a student is motivated and expresses curiosity to explore a particular learning goal, however s/he might have some misconceptions and needs to reconsider her/his knowledge. This can evoke frustration and/or disappointment. However, this negative emotion can turn into curiosity again, if the student gets a new idea on how to solve the learning task.

[9] categorised emotions based on facial expressions. These included, joy, anger, surprise, fear, and disgust/contempt. However, these emotions are not specific to learning. [22] classified achievement emotions that arise in a learning situation. Achievement emotions are emotions that are linked to learning, instruction, and achievement. Emotions are classified into prospective, retrospective and activity emotions. They can be positive or negative. For example, a prospective positive emotion is hope for success, while a negative emotion is anxiety about failure. Retrospective emotions are for example, the positive emotion pride or the negative emotion shame, which the student experienced after receiving feedback of an achievement. Activity emotions arise during learning, such as positive emotions like enjoyment, or negative emotions like anger, frustration, or boredom.

We focus on on a subset of emotions identified by Pekrun and Ekman: enjoyment, surprise, frustration, and boredom. We also add confusion as an emotion, which is placed between enjoyment and frustration.

As described in [29] students can become overwhelmed (very confused or frustrated) during learning, which may increase cognitive load for low-ability or novice students. However, appropriate feedback can help to overcome such problems.

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Effective support or feedback needs to answer four main questions: *when*, *what*, *how*, and *why*: (i) *when* to provide the support during learning. (ii) It needs to be decided *what* the support should contain; (iii) *how* it should be presented; and (iv) *why* the feedback needs to be provided.

In this paper we focus on *what* (ii) and *why* (iv) support or feedback should be provided based on the student's emotion. In the area of intelligent tutoring systems or learning environments, the only research we are aware of specifically targeting the question of responding to student affect is [29] and [2]. [29] describes how an embodied pedagogical agent is able to provide different types of interventions, such as praising or mirroring the student's emotional state. [2] looks at the effect of cognitive-affective states on student's learning behaviour. In contrast, in this paper, we investigate the impact of emotions on the effectiveness of different feedback types.

The structure of the paper is as follows: The next section overviews related work on detecting and adapting to emotions in the educational domain. This is followed by a description of the Wizard-of-Oz study, which investigated the effect of emotions on different feedback types. We then discuss the different feedback types. After this, we provide results and discuss the results of the study in respect to adaptive support based on student's emotion. We conclude by outlining directions for future research.

2. BACKGROUND

Different computational approaches have been taken into account in order to detect emotions. These include for example, speech-based approaches (e.g. [6, 27]), using information from facial expressions (e.g. [14]), keystrokes or mouse movements [10], physiological sensors (e.g. [16, 28, 21]), or a combination of these [7].

In the area of education [5] developed a model of emotions (Dynamic Bayesian network) based on students' bodily expressions for an educational game. The system uses six emotional states: joy, distress, pride, shame, admiration and reproach. A pedagogical agent provides support according to the emotional state of the students and the user's personal goal, such as wanting help, having fun, learning maths, or succeeding by oneself.user's personal goal, such as wanting help, having fun, learning maths, or succeeding by oneself.

Another example, is [25] who also used Bayesian Networks to classify students' emotions. Here biophysical signals, such as heart rate, skin conductance, blood pressure, and EEG brainwaves, for the classification of emotions. These include: interest, engagement, confusion, frustration, boredom, hopefulness, satisfaction, and disappointment.

As described earlier, [29] developed an affective pedagogical agent which is able to mirror students' emotional state, or acknowledge a student's emotion if it is negative. They use hardware sensors and facial movements to detect students emotion. The system discriminates between seven emotions: high/low pleasure, frustration, novelty, boredom, anxiety, and confidence. Different machine learning techniques were applied for the classification, including Bayesian Networks and Hidden Markov models.

[17] developed a physics text-based tutoring system called ITSPOKE. It uses spoken dialogue to classify emotions. Acoustic-prosodic and lexical features are used to predict student emotion. They apply boosted decision trees for their classification. Three emotion types are detected: negative, neutral and positive emotions.

Another example is the AutoTutor tutoring system [7], which holds conversations with students in computer literacy and physics courses. The system classifies emotions based on natural language interaction, facial expressions, and gross body movements. The focus is on three emotions, namely frustration, confusion, and boredom. The classification is used to respond to students via a conversation.

Most of the related work in the educational domain focusses on detecting emotions based on different input stimuli, ranging from spoken dialogue to physiological sensors. However, little research has been done on how those detected emotions can be used in a tutoring system to enhance the learning experience. One exception is [29] who describes how an affective pedagogical agent can support students in particular emotional states. Additionally, [2] investigated the impact of student's cognitive-affective states on how they interacted with the learning environment. They found that certain types of emotions, such as boredom, were associated with poor learning and gaming the system. In contrast, we investigate the implications of emotions for different feedback types. We conducted a WoZ study where different kinds of feedback were provided to students in different emotional states. The next section describes the WoZ study in more detail.

2.1 Aims

One of our research aims is to provide adaptive feedback to students during a learning activity which enhances the learning experience by taking into account students' emotion. We were specifically interested in the following questions, which we aimed to address in the WoZ studies:

- Is there an effect of different emotion types upon reaction towards feedback?
- Which interventions were most successful given a particular emotional state?

In order to address these questions we ran an ecologically valid WoZ study which investigated the effect of emotions on different feedback types at different stages of the task.

2.2 Methodology

The studies reported on this paper are part of a methodology referred to as Iterative Communication Capacity Tapering (ICCT). This can be used to inform the design of intelligent support for helping students in interactive educational applications [18]. During the first phase, the facilitator gradually moves from a situation in which the interaction with the student is close, fast, and natural (i.e. face-to-face free interaction) towards a situation in which the interaction is mediated by computer technologies (e.g. voice-over-ip or similar for voice interaction, instant messaging or similar for

textual interaction) and regularised by means of a script. In the second phase, the script is crystallized into a series of intelligent components that produce feedback in the same way that the human facilitator formaly did. The gradual reduction of communication capacity and the iterative nature of the process maximise the probability of the computer-based support being as useful as the facilitator's help. In this paper, we are already starting the second phase, i.e. gradually replacing humans by a computer-based system. Experts ('wizards') are not physically near enough to the students to observe them directly, and therefore must observe them by indirect mediated means: the students' voice was heard by using microphones and headsets and their screen was observed by a mirror screen. The wizards did not have direct access to the students' screens (so e.g. could not point to anything on the screen to make a point), could not see the students' faces (for facial cues), and could not communicate to students by using body language, only by means of the facilities provided by the wizard-of-oz tools that resemble those of the final system.

2.3 Participants and Procedure

After returning informed consent forms signed by their parents 60 Year-5 (9 to 10-year old) students took part in a series of sessions with the learning platform configured for learning fractions through structured tasks from the intelligent tutoring system, together with more open-ended tasks offered by the exploratory learning environment. The sessions were designed to first familiarise all students with the environment, and then to allow them to undertake as many tasks as possible (in a study which has goals outside the scope of this paper). In parallel, we were running the WOZ study by asking two students in each session to work on different computers as described below. In total 12 students took part in the WOZ study but due to data errors we were able to analyse the interaction of only 10 students. At the end of the session the students who participated in the WOZ joined in a focus group discussing their experience with the learning platform. We were particularly interested in students' opinions about the different feedback types provided.

2.4 Classroom setup

The ecological validity of the study was achieved by following the setup depicted in Figure 1, 2 and Figure 3. The classroom where the studies took place is the normal computer lab of the school in which most of the computers are on tables facing the walls in a II-shape, and a few are on a central table. This is the place where the WOZ study took place, while, for ecological validity, the rest of the class was working on the other computers. The students were only told that the computers in the central isle were designed to test the next version of the system and were thus also responding to (rather than just recording as the rest of the computers) their speech. The central isle has two rows of computers, facing opposite directions, and isolated by a small separator for plugs etc. In the central isle the students worked on a console consisting on a keyboard, a mouse, and a screen. Usually, those components are connected to the computer behind the screen; for these studies, they were connected to a laptop on the wizards' side of the table. This allowed the wizard to observe what the students were doing. As the learning platform is a web-based system, and all the students' see is a web browser, the operating system and general look-and-feel of the experience was equivalent to the one that the rest of the students were using. When the wizards wanted to intervene, they used the learning platform's WOZ tools to send messages to the student's machine. These messages were both shown on screen and read aloud by the system to students, who could hear them on their headset.

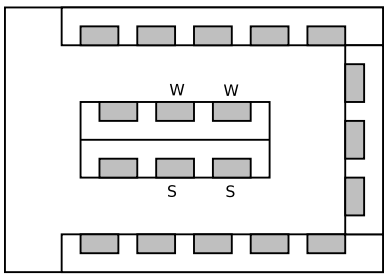


Figure 1: The layout. The Wizard-of-Oz studies took place on the central isle while the rest of the students worked on a version of the system which only sequences tasks and provides minimal support.

Figure 2: The classroom. The children being wizarded in front with wizards at the back.

2.5 The wizard's tools

In line with the ICCT methodology mentioned above, the wizards restricted their 'freedom' in addressing the students by employing a pre-determined agreed script in which the expected interventions had been written. Figure 4 shows a high-level view of this script, the end-points of which require further decisions also agreed in advance in a protocol but not shown here for simplicity. In this study, we limited ourselves to written interventions that could be selected from an online document appropriate for being read aloud by the system. There were no other kinds of interventions (such as sounds, graphical symbols on screen etc.). The intervention had a set of associated conditions that would fire them thus resembling very closely the system under development.

2.6 Feedback types

As outlined in the script (figure 4) different types of feedback were presented to students at different stages of their learning task. The feedback provided was based on interaction via keyboard and mouse, as well as speech.

From an HCI perspective speech production and recognition can provide potentially more intuitive interaction. In

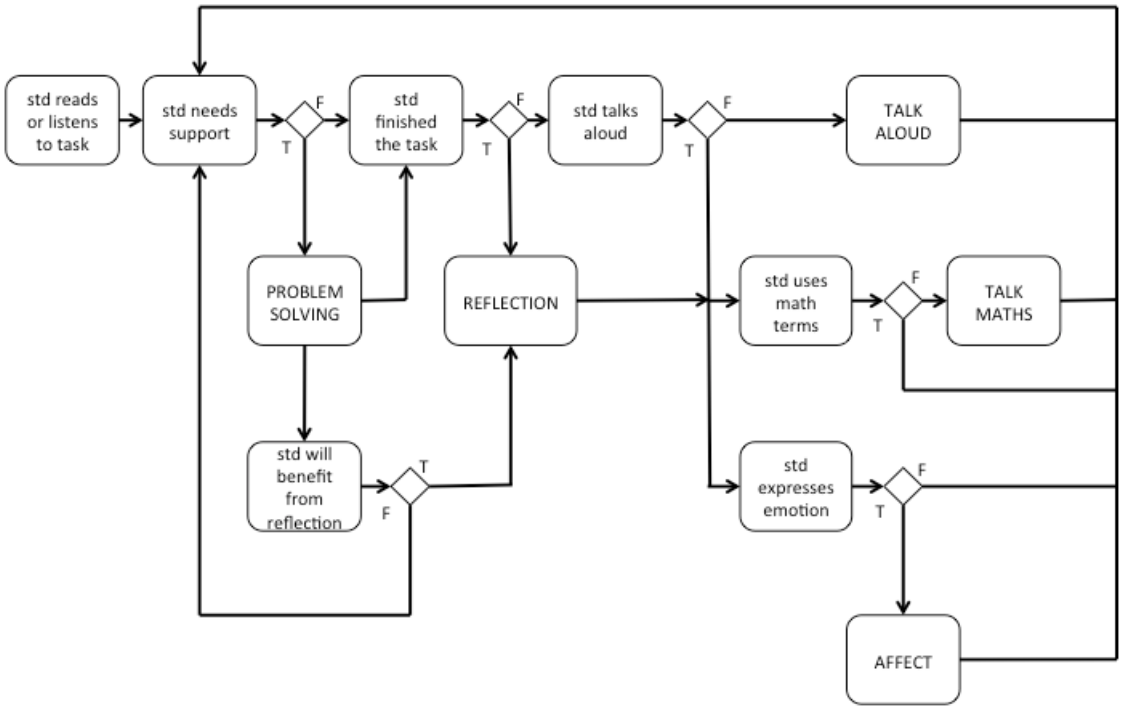


Figure 4: Flowchart representing the wizard's script for support.

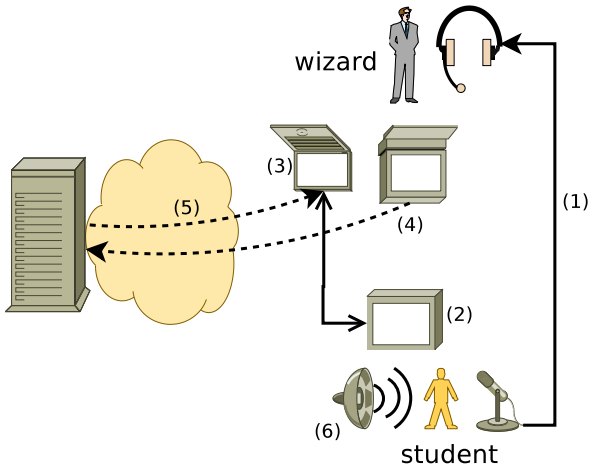


Figure 3: Wizard-of-oz setup. Each student speaks on a headset (mic) which is connected to the wizard's headset (1). The student interacts with a console (i.e. keyboard, mouse, screen) connected to a laptop on the wizard's side (2,3) so that the latter can witness their interaction. The wizard can send messages (4) by using some ad-hoc wizard tools. These messages arrive at the student laptop (5) and are shown on the screen of the student's monitor and read aloud on the student's headset (6).

particular, spoken language input can enable students to communicate verbally with an educational application and thus interact without using human interface devices such as a mouse or keyboard. The following different feedback types were provided:

- **PROBLEM SOLVING - task-dependent feedback**
  This feedback based mainly on the interaction with mouse and keyboard with the learning environment. Here the feedback involved providing support in solving a particular maths problem.
- **TALK MATHS - using particular domain specific maths vocabulary**
  The importance of students' verbal communication in mathematics in particular becomes apparent if we consider that learning mathematics is often like learning a foreign language. Focusing, for example, on learning mathematical vocabulary, [3] encouraged students to talk to a partner about a mathematical text to share confusions and difficulties, make connections, put text into their own words and generate hypotheses. This way, students were able to make their tentative thinking public and continually revise their interpretations.
- **AFFECT - affect boosts**
  As described in [29] affect boosts can help to enhance student's motivation in solving a particular learning task. Higher motivation also implies better performance.
- **TALK ALOUD - talking aloud**
  With respect to learning in particular, the hypothesis

that automatic speech recognition (ASR) can facilitate learning is based mostly on educational research that has shown benefits of verbalization for learning (e.g., [1, 3, 20]).

The possible verbalization effect could be enhanced with ASR since cognitive load theory [26] and cognitive theory of multimedia learning [19] predict that a more natural and efficient form of communication will also have positive learning gains.

The few existing research studies have found mixed results with respect to whether the input modality (speaking vs. typing) has a positive, negative or no effect on learning. In [8], for example, the authors investigated whether student typing or speaking leads to higher computer literacy with the use of AutoTutor. They reported mixed results that highlight individual differences among students and a relationship to personal preferences and motivation.

- **REFLECTION - reflecting on task performance and learning**
  For further consideration is the research about self-explanation; an efficient learning strategy where students are prompted to verbalize their thoughts and explanations about the target domain to make knowledge personally meaningful. Previous research [13] found that the amount of self-explanation that students generated in a computer environment was suppressed by having learners type rather than speaking and the studies. Moreover, some students are natural self-explainers while others can be trained to self-explain [24]. Even when self-explanation is explicitly elicited, it can be beneficial [4] but requires going beyond asking students to talk aloud by using specific reflection prompts [24].

  Self-explanation can be viewed as a tool to address students' own misunderstandings [4] and as a 'window' into students' thinking. While it may be early days for accurate speech recognition to be able to highlight specific errors and misconceptions, undertaking carefully-designed tasks can help identify systematic errors that students make. For example, [12] explores how naming and misnaming involves logic and rules that often aid or hinder students' mathematical learning and relate to misconceptions.

  A lack of mathematical terminology can also be noticed and prompts made to students to use appropriate language as they self-explain.

Table 1 shows examples of the different feedback types. We were interested to explore how emotions impact on the effectiveness of those different feedback types.

3. RESULTS

From the WoZ study we recorded students' screen display and their voices. From this data, we annotated emotions and whether students reacted to feedback.

For the annotation of the emotions and students reactions towards the feedback, we used a similar strategy as described in [23] where dialog between a teacher and a student was annotated according to different feedback types. Also,[2] describe how they coded different cognitive-affective states based on observations of students interacting with a learning environment. Similarly, we annotated student's emotion and if they reacted for each type of feedback provided. Another researcher went through the categories and any discrepancies were discussed and resolved before any analysis took place.

Feedback type	Example
AFFECT	It may be hard, but keep trying. If you find this easy, check your work and change the task.
TALK ALOUD	Remember to talk aloud, what are you thinking? What is the task asking you to do?
TALK MATHS	Can you explain that again using the terms denominator, numerator?
PROBLEM SOLVING	You can't add fractions with different denominators.
REFLECTION	What did you learn from this task? What do you notice about the two fractions?

Table 1: Examples of feedback types

In total 170 messages were sent to 10 students. The raw video data was analysed by a researcher who categorised the emotions and feedback messages. Table 1 shows the different types of messages send to students and the emotions that occurred while the feedback was given. It can be seen that most frequent messages were reminders to talk aloud (66). This was followed by problem-solving feedback (55), and feedback according to students emotions (31). The least frequent messages relates to reflection (13) and using maths terminology (5).

It is not surprising that most of the problem solving feedback was provided when students were confused (35 out of 55). Most of the affect boosts were provided when students enjoyed the activity (15 out of 31), closely followed by students' being confused (11 out of 31). Most of the reflection prompts were given when students enjoyed the activity (10 out of 13). Talk aloud reminders were mainly given when students were confused (30 out of 66). Talk maths prompts were mainly given when students enjoyed the task (3 out of 5) or when they were confused (2 out of 5).

The emotions that were detected by students when feedback was provided and whether students reacted can be seen in figure 5.

Students reacted to all of the feedback when they were bored or surprised (100%).This was followed by reactions to feedback when students were confused (83%) or enjoyed the activity (81%). Students responded the least if they were frustrated (69%).

Looking in more detail at emotions and whether students reacted to the different feedback types, figures 6, 7, and 8 show the percentage of student's reaction towards feedback type for enjoyment, confusion, and frustration.

	emotion					
Feedback type	enjoyment	boredom	confusion	frustration	surprise	total
PROBLEM SOLVING	8	3	35	8	1	55
TALK MATHS	3	0	2	0	0	5
AFFECT	15	2	11	3	0	31
TALK ALOUD	21	1	40	4	0	66
REFLECTION	10	1	1	1	0	13
Total	57	7	89	16	1	170

Table 2: Feedback types, including emotion that occurred while the feedback was provided.

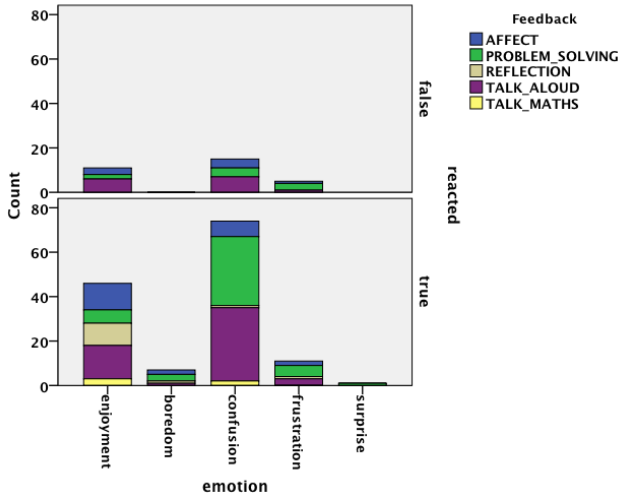


Figure 5: Student's reaction according to feedback types and emotion.

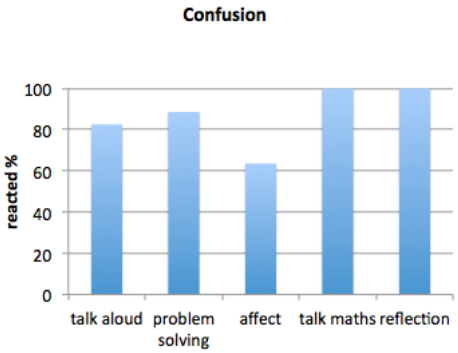


Figure 7: Students' reaction according to feedback types if they were confused.

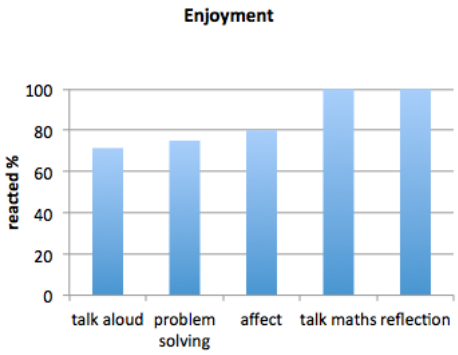


Figure 6: Students' reaction according to feedback types if they enjoyed the activity.

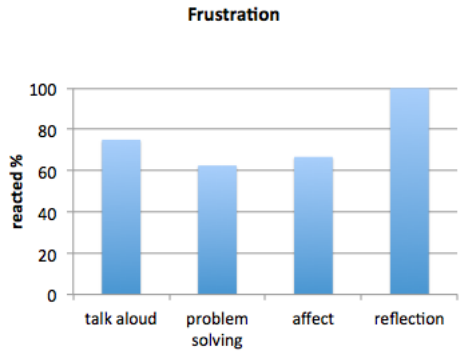


Figure 8: Students' reaction according to feedback types if they were frustrated.

It is interesting to see that while students enjoyed their activity, they responded very well to talk maths (100%) or to reflect on what they have done (100%). The least reaction was given if students were prompted to talk aloud (71%).

If students were confused they responded well again on talk maths (100%) or reflection prompts (100%), followed by problem solving feedback (89%). Surprisingly, least reactions were given when affect boosts were provided (64%).

If students were frustrated most reactions were given for reflection (100%) and prompts to talk aloud (75%). Least responses were given if problem solving feedback was provided (63%).

4. DISCUSSION

The key findings with respect to impact of emotions on the effect of feedback types are listed below in relation to our research aims.

4.1 Is there an effect of different emotion types upon reaction towards feedback?

The results show that for certain types of emotions, such as boredom, any type of feedback is reacted to. This indicates that students may welcome a distraction from their learning and react to feedback if they are bored. As boredom indicates a reduction in learning [2], the feedback provided to students when they are bored should aim to motivate and support the student to continue with the learning task.

Also in most of the cases students reacted to the feedback when they were confused. This implies that students welcome feedback that will help them to get out of their confused state. In designing feedback for learning environments students should be provided with feedback that enables them to overcome their confusion, such as task-dependent problem solving feedback, or feedback to reflect on their learning, which might help to identify and overcome misconceptions.

Additionally, students mainly reacted to feedback when they were enjoying their activity. This is an interesting finding, as in theory this seems to interrupt their learning flow. Here, it seems students' motivation is high and they did not mind being interrupted. Students particularly reacted positively on feedback to reflect.

In contrast, when students were frustrated, they reacted to feedback in only 69% of the cases. This indicates that frustration can reduce motivation and may also increase cognitive load. Here feedback that might help to decrease the frustration, such as reflecting on the difficulty of the learning task might help to motivate the student.

4.2 Which interventions were most successful given a particular emotional state?

The results indicate that for different emotional states, different feedback types are more effective than others.

It is interesting to see that although students enjoyed their activity and reacted to feedback in 81% of the cases, response to talk aloud was only 71%. This was similar when students were frustrated (75%). In contrast when students were confused in 83% of the cases students followed the recommendation to talk aloud. It looks like as if talking aloud might help to identify the problem and might resolve the confusion.

The highest reaction was given to problem solving feedback if students were confused (89%). This is not surprising as students were happy to receive help to perform the task. However, in only 75% of the cases was problem solving feedback reacted to while students enjoyed the activity. This might be because they were interrupted in their learning flow and they needed to switch to a new strategy of answering the learning task based on the problem solving feedback. The number drops even more when students were frustrated (63%). As discussed above, students' motivation might be low when frustrated and also there might be increased cognitive load. Providing problem solving feedback when students are frustrated does not seem to be a very effective strategy.

Providing affect boosts was most effective when students enjoyed their activity (80%). In contrast, students only reacted to affect boosts in 67% of the cases when they were frustrated or 64% when they were confused. From the focus group with the students it emerged that although some students did not react to the emotional boosts when they were confused or frustrated, they liked the encouragement, and that it helped with their motivation to continue to work on the particular learning task.

Providing prompts to talk maths and reflection were very effective across the emotion types. Despite the fact that 5 talk maths prompts and 13 reflection prompt were provided, students seemed to respond to them very well whether confused or frustrated. This implies that reflecting on one's own strategy of solving a task is motivating even if confused or frustrated. We noticed that it may also helped students to identify misconceptions or lead to new ideas on how to solve the learning task.

5. CONCLUSION AND FUTURE WORK

We explored the impact of students' emotional state upon different feedback types. The results indicate that certain types of feedback are more effective then others according to the emotional state of the student. While for some emotional states, such as boredom, a variety of feedback types worked well, for other emotional states, like frustration, only a few types of feedback seem to be effective.

We are now developing and integrating the automatic speech and emotion recognition in our learning platform. Additionally the adaptive support that is able to provide the different feedback types for particular emotional states is under development. At the next stage of our research we are interested to explore how the presentation of the feedback (e.g. high or low intrusive) affects students being interrupted in performing the task and if the presentation has an effect on reaction towards the feedback.

6. ACKNOWLEDGMENTS

This research has been co-funded by the EU in FP7 in the iTalk2Learn project (318051). Thanks to all our iTalk2Learn colleagues for their support and ideas.

7. REFERENCES

[1] M. Askeland. Sound-based strategy training in multiplication. *European Journal of Special Needs Education*, 27(2):201–217, 2012.

[2] R. S. J. d. Baker, S. K. D'Mello, M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *Int. J. Hum.-Comput. Stud.*, 68(4):223–241, apr 2010.

[3] R. Borasi, M. Siegel, J. Fonzi, and C. Smith. Using transactional reading strategies to support sense-making and discussion in mathematics classrooms: An exploratory study. *Journal for Research in Mathematics Education*, 29:275–305, 1998.

[4] M. Chi. Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In R. Glaser, editor, *Advances in instructional psychology*, pages 161–238. Mahwah, NJ: Lawrence Erbaum Associates, 2000.

[5] C. Conati and H. MacLaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 2009.

[6] R. Cowie, E. Douglas-Cowie, B. Apolloni, J. Taylor, A. Romano, and W. Fellenz. What a neural net needs to know about emotion words. *Computational Intelligence and Applications*, pages 109–114, 1999.

[7] S. D'Mello, S. Craig, B. Gholson, S. Franklin, R. Picard, and A. Graesser. Integrating affect sensors in an intelligent tutoring system. In *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International Conference on Intelligent User Interfaces*, pages 7–13, 2005.

[8] S. K. D'Mello, N. Dowell, and A. Graesser. Does it really matter whether student's contributions are spoken versus typed in an intelligent tutoring system with natural language? *Journal of Experimental Psychology; Applied 2011*, 17(1):1–17, 2011.

[9] P. Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.

[10] C. Epp, M. Lippold, and R. Mandryk. Identifying emotional states using keystroke dynamics. In *2011 Annual Conference on Human Factors in Computing Systems*, pages 715–724, 2011.

[11] J. H. Flavell, F. L. Green, E. R. Flavell, and J. B. Grossman. The Development of Children's Knowledge of Inner Speech. *Child Development*, 68(1):39–47, 1997.

[12] H. A. Furani. Misconceiving or misnaming?: Some implications of toddlers' symbolizing for mathematics education. *Philosophy of Mathematics Education Journal*, 17, 2003.

[13] R. G. M. Hausmann and M. T. H. Chi. Can a computer interface support self-explaining? *Cognitive Technology*, 7(1):4–14, 2002.

[14] R. Kaliouby and P. Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[15] B. Kort, R. Reilly, and R. Picard. An affective model of the interplay between emotions and learning. In *IEEE International Conference on Advanced Learning Technologies*, number 43-46, 2001.

[16] P. Lang, M. Greenwald, M. Bradley, and A. Hamm. Look at pictures: Affective, facial, visceral, and behavioral reactions. *Psychophysiology*, 30:261–273, 1993.

[17] D. Litman and K. Forbes-Riley. Predicting student emotions in computer-human tutoring dialogues. In *42nd Annual Meeting on Association for Computational Linguistics (ACL '04), Association for Computational Linguistics*, 2004.

[18] M. Mavrikis and S. Gutierrez-Santos. Not all wizards are from Oz: Iterative design of intelligent learning environments by communication capacity tapering. *Computers & Education*, 54(3):641–651, Apr. 2010.

[19] R. E. Mayer and R. Moreno. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, 38(1):43–52, 2003.

[20] N. Mercer and C. Sams. Tecahing children how to use language to solve maths problems. *Language and Education*, 20(6):507–528, 2007.

[21] F. Nasoz, K. Alvarez, C. Lisetti, and N. Finkelstein. Emotion recognition from physiological signals for presence technologies. *International Journal of Cognition, Technology and Work, Special Issue on Presence*, 6(1):4–14, 2003.

[22] R. Pekrun. The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *J. Edu. Psych. Rev.*, pages 315–341, 2006.

[23] K. Porayska-Pomsta, M. Mavrikis, and H. Pain. Diagnosing and acting on student affect: the tutor's perspective. *User Modeling and User-Adapted Interaction*, 18(1):125–173, Feb. 2008.

[24] M. Roy and M. T. H. Chi. The self-explanation principle in multimedia learning. In R. E. Mayer, editor, *Cambridge handbook of multimedia learning*, pages 271–286. New York: Cambridge University Press, 2005.

[25] L. Shen, M. Wang, and R. Shen. ffective e-learning: Using "emotional" data to improve learning in pervasive learning environment. *Educational Technology & Society*, 12(2):176–189, 2009.

[26] J. Sweller, J. G. van Merrienboer, and G. W. Paas. Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10:251 – 296+, 1998.

[27] T. Vogt and E. André. Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition. In *Multimedia and Expo (ICME05)*, pages 474–477, 2005.

[28] E. Vyzas and R. Picard. Affective pattern classification. In *AAAI Fall Symposium, Emotional and Intelligent: The Tangled Knot of Cognition*, pages 176–182, 1998.

[29] B. Woolf, W. Burleson, I. Arroyo, T. Dragon, D. Cooper, and R. Picard. Affect-aware tutors: recognising and responding to student affect. *Int. J. Learning Technology*, 4(3-4):129–164, 2009.

Multimodal Affect Recognition for Adaptive Intelligent Tutoring Systems

Ruth Janning
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
janning@ismll.uni-hildesheim.de

Carlotta Schatten
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
schatten@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

The performance prediction and task sequencing in traditional adaptive intelligent tutoring systems needs information gained from expert and domain knowledge. In a former work a new efficient task sequencer based on a performance prediction system was presented, which only needs former performance information but not the expensive expert and domain knowledge. In this paper we aim to support this approach by automatically gained multimodal input like for instance speech input from the students. Our proposed approach extracts features from this multimodal input and applies to that features an automatic affect recognition method. The recognised affects shall finally be used to support the mentioned task sequencer and its performance prediction system. Consequently, in this paper we (1) propose a new approach for supporting task sequencing and performance prediction in adaptive intelligent tutoring systems by affect recognition applied to multimodal input, (2) present an analysis of appropriate features for affect recognition extracted from students speech input and show the suitability of the proposed features for affect recognition for adaptive intelligent tutoring systems, and (3) present a tool for data collection and labelling which helps to construct an appropriate data set for training the desired affect recognition approach.

Keywords

multimodal input, affect recognition, feature analysis, speech, adaptive intelligent tutoring systems

1. INTRODUCTION

Learning management systems like intelligent tutoring systems are an important tool for supporting the education of students for instance in learning fractional arithmetic. The main advantages of intelligent tutoring systems are the possibility for a student to practice any time, as well as the possibility of adaptivity and individualisation for a single student. An adaptive intelligent tutoring system possesses an internal model of the student and a task sequencer which decides which tasks in which order are shown to the student. Originally, the task sequencing in adaptive intelligent tutoring systems is done using information gained from expert and domain knowledge and logged information about the performance of students in former exercises. In [12] a new efficient sequencer based on a performance prediction system was presented, which only uses former performance information from the students to sequence the tasks and does not need the expensive expert and domain knowledge. This approach applies the machine learning method matrix factorization (see e.g. [1]) for performance prediction to former performance information. Subsequently, it uses the output of the performance prediction process to sequence the tasks according to the theory of Vygotsky's Zone of Proximal Development [14]. That is the sequencer chooses the next task in order to neither bore nor frustrate the student or in other words, the next task should not be too easy or too hard for the student.

In this paper we propose to support the task sequencer and performance prediction system of the approach in [12] in a new way by further automatically to get and process multimodal information. One part of this multimodal information, which is investigated in this paper, is the speech input from the students interacting with the intelligent tutoring system while solving tasks. A further part will be the typed input or mouse click input from the students, which will be reported in upcoming works. The approach proposed in this paper extracts features from the mentioned multimodal information and applies to that features an automatic affect recognition method. The output of the affect recognition method indicates, if the last task was too easy, too hard or appropriate for the student. This information matches the theory of Vygotsky's Zone of Proximal Development, hence it is obviously suitable for supporting the performance prediction system and task sequencer of the approach in [12]. However, for the proposed approach we need a large amount

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

of labelled data. For this reason we developed a tutoring tool which (a) records students speech input as well as typed input and mouse click input and (b) allows the students to label by themselves how difficult they perceived the shown tasks. This tool is presented in the second part of this paper and will be used to conduct further studies to gain the desired labelled data.

The main contributions of this paper are: (1) presentation of a new approach for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems by affect recognition on multimodal input, (2) identification and analysis of appropriate and statistically significant features for the presented approach, and (3) presentation of a new tutoring tool for multimodal data collection and self-labelling to gain automatically labelled data for training appropriate affect recognition methods.

In the following, first we will present some preliminary considerations along with state-of-the-art in section 2. Subsequently, we will describe in section 3 the real data set used for the feature analysis and investigate in section 4 for the data set the correlation between students affects and their performance. In section 5 we will propose and analyse appropriate features for affect recognition and in section 6 we will explain how to support performance prediction and task sequencing in intelligent tutoring systems by affect recognition applied to multimodal input. Before we conclude, we will describe in section 7 the mentioned tool for multimodal data collection and self-labelling.

2. PREPARATION AND RELATED WORK

Before an automatic affect recognition approach can be applied, one has to clarify three things: (1) What kind of features shall be used, (2) what kind of classes shall be used and (3) which instances shall be mapped to features and labelled with the class labels. After deciding which features, classes and instances shall be considered, one can apply affect recognition methods to these input data. In the following subsections we will present possible features, classes, instances and methods for affect recognition supporting performance prediction and task sequencing in adaptive intelligent tutoring systems along with the state-of-the-art.

2.1 Features

The first step before applying automatic affect recognition is to identify useful features for this process. For the purpose to recognise affect in speech one can use two different kinds of features ([13]): acoustic and linguistic features. Further, one can distinct linguistics (like n-grams and bag-of-words) and disfluencies (like pauses). If linguistics features are used, a transcription or speech recognition process has to be applied to the speech input before affect recognition can be conducted. Subsequently, approaches from the field of sentiment classification or opinion mining (see e.g. [10]) can be applied to the output of this process. However, the methods of this field have to be adjusted to be applicable to speech instead of written statements.

Another possibility for speech features is to use disfluencies features like it was done in [17], [7] and [4] for expert identification. The advantage of using such features is that instead of a full transcription or speech recognition approach only for instance a pause identification has to be applied before. That means that one does not inherit the error of the full speech recognition approach. Furthermore, these features are independent from the need that students use words related to affects. For using this kind of features one has to investigate, which particular features are suitable for the special task of affect classification in adaptive intelligent tutoring systems. Because of the mentioned advantage of disfluencies features in this work we focus on features extracted from information about speech pauses as one part of the multimodal input for affect recognition.

As mentioned in the introduction the other part of the multimodal input will be features which are gained from information about typed input or mouse click input from the students. This kind of features is similar to the keystroke dynamics features used in [2]. In [2] emotional states were identified by analysing the rhythm of the typing patterns of persons on a keyboard.

2.2 Classes

The second step before applying automatic affect recognition is to define the classes corresponding to emotions and affective states, which shall be recognised by the used affect recognition approach. According to [6], [5] and [16] it is possible to recognise in intelligent tutoring systems students affects like for instance confusion, frustration, boredom and flow. As mentioned above, we want to use the students behaviour information gained from speech and from typed input or mouse click input for supporting the performance prediction system and task sequencer of the approach in [12], which is based on the theory of Vygotsky's Zone of Proximal Development [14]. That means that the goal is to neither bore the student with too easy tasks nor to frustrate him with too hard tasks, but to keep him in the Zone of Proximal Development. Accordingly, we want to use the output of the automatic affect recognition to get an answer to the question "Was this task too easy, too hard or appropriate for the student?", or with other words we want to find out if the student felt under-challenged, over-challenged or like to be in a flow. However, the mapping between confusion, frustration, boredom and under-challenged, over-challenged is not unambiguous as one can infer e.g. from the studies mentioned in [16]. Hence, we will use instead of the above mentioned affect classes three other classes for supporting performance prediction and task sequencing by automatic affect recognition: under-challenged, over-challenged and flow. One could summarise these classes as *perceived task-difficulty classes*, as we aim to recognise the individual perceived task-difficulty from the view of the student.

2.3 Instances

The third step before applying automatic affect recognition is deciding which instances shall be mapped to features and labelled with the class labels. If the goal of the affect recognition is to provide a student motivation or hints according to his affective state like e.g. in [16], then instances can be utterances. For supporting performance prediction and task sequencing by affect recognition instead one needs at the end of a task the information, if the task overall was too easy, too hard or appropriate for the student. The reason is that this information shall help to choose the next task shown to the student. Hence, an instance for supporting perfor-

mance prediction and task sequencing by affect recognition has to be instead of an utterance the whole speech input of a student for one task.

2.4 Methods

The possible methods for an automatic affect recognition depend on the kind of the features used as input. As mentioned above, for speech we distinct two kinds of features: linguistics features and disfluencies. Linguistics features are gained by a preceding speech recognition process and can be processed by methods coming from the areas sentiment analysis and opinion mining ([10]). Especially methods from the field of opinion mining on microposts seem to be appropriate if linguistics features are considered. State-of-the-art approaches in opinion mining on microposts use methods for instance based on optimisation approaches ([3]) or Naive Bayes ([11]).

The process of gaining disfluencies like pauses is different to the full speech recognition process. For extracting for instance pauses usually an energy threshold on the decibel scale is used as in [4] or an SVM is applied for pause classification on acoustic features as in [9]. Appropriate state-of-the-art methods for automatic emotion and affect recognition on disfluencies features as well as on features from information about typed input or mouse click input are – as proposed e.g. in [13] and [6] – classification methods like artificial neural networks, SVM, decision trees or ensembles of those.

3. REAL DATA SET

After identifying features, classes, instances and methods for affect recognition for supporting performance prediction and task sequencing like above one can collect data for a concrete feature analysis and a training of the chosen affect classification method. We conducted a study in which the speech and actions of ten 10 to 12 years old German students were recorded and students affective states as well as the perceived task-difficulties were reported. The labelling of these data was done on the one hand concurrently by the tutor and on the other hand retrospectively by a second reviewer. Furthermore, a labelling per exercise (consisting of several subtasks) and an overall labelling per student as an aggregation of the labels per exercise was done. During the study a paper sheet with fraction tasks was shown to the students and they were asked to paint (with the software Paint) and explain their observations and answers. We made a screen recording to record the painting of the students and an acoustic recording to record the speech of the students. The screen recordings were used for the retrospective annotation. The speech recordings shall be used to gain the input for affect recognition. The mentioned typed input or mouse click input information we will collect and investigate in further studies with the self-labelling and multimodal data collection tutoring tool described in section 7.1. In this paper we focus on speech features and hence in section 5 we will propose and analyse possible features extracted from speech pauses. But first we will investigate in the following section 4 the correlation between perceived task-difficulty labels and the performance of the students in the real data set.

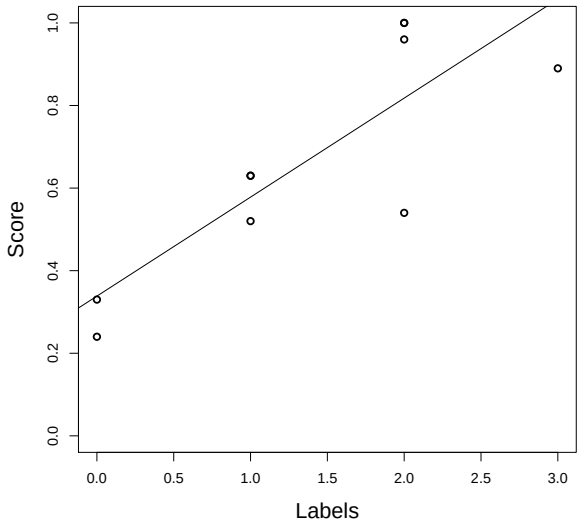


Figure 1: Mapping of the perceived task-difficulty labels to the scores of the students in the real data set.

4. CORRELATION OF PERCEIVED TASK-DIFFICULTY LABELS AND SCORE

Before we present speech features for recognising perceived task-difficulty, we want to show that there is a correlation between the proposed perceived task-difficulty labels and the performance of the students, to underline the suitability of supporting performance prediction and task sequencing by the proposed affect recognition approach. Hence, we mapped the overall perceived task-difficulty labels to the overall score of the students (see figure 1). For this mapping we encoded the different overall perceived task-difficulty class labels as follows:

- 0 = over-challenged
- 1 = over-challenged/flow
- 2 = flow
- 3 = flow/under-challenged
- 4 = under-challenged

The overall score of a student i is computed by

$$\frac{n_{c_i}}{n_{t_i}}\ , \tag{1}$$

where n_{c_i} is the number of correctly solved tasks of student i and n_{t_i} is the number of tasks shown to student i. In figure 1 one can see that there is a clear correlation between perceived task-difficulty labels and score. To substantiate this observation we applied a statistical test by conducting a linear regression and measuring the p-value, indicating the statistical significance, as well as the R^2 and Adjusted R^2 value, indicating how well the regression line can approximate the real data points. This approach delivers a p-value of 0.0027,

Figure 2: Graphic of the decibel scale of an example sound file of a student. The two straight horizontal lines indicate the threshold.

a R^2 value of 0.6966, and an Adjusted R^2 value of 0.6586. The small p-value indicates a strong statistical significance. The significant correlation between perceived task-difficulty labels and scores, which demonstrate the performance, indicates that it makes sense to support performance prediction and task sequencing by perceived task-difficulty classification.

5. SPEECH FEATURE ANALYSIS

The features we propose and analyse in this section are gained from speech pauses. Hence, first one has to identify pauses within the speech input data. The most easy way is to define a threshold on the decibel scale as done e.g. in [4]. For our preliminary study of the data we also used such a threshold, which we adjusted by hand. More explicitly, we extracted the amplitudes of the sound files and computed the decibel values. Subsequently, we investigated which decibel values belong to speech and which ones to pauses (see figure 2). In larger data and in the application phase later on, one has to learn automatically the distinction between speech and pauses by either learn the threshold or train an SVM, which classifies speech and pauses.

5.1 Single Feature Analysis

Before we can introduce the features we want to investigate, we have to define some measurements:

- m: number of students
- p_i: total length of pauses of student i
- s_i: total length of speech of student i
- n_{p_i}: number of pause segments of student i
- n_{s_i}: number of speech segments of student i
- $p_i^{(x)}$: xth pause segment of student i
- $s_i^{(y)}$: yth speech segment of student i
- n_{t_i}: number of tasks shown to student i
- n_{c_i}: number of correctly solved tasks by student i
- Overall score for student i: $\frac{n_{c_i}}{n_{t_i}}$

Table 1: p-value, R^2 and Adjusted R^2 for the feature *Length of maximal pause segment* mapped to score as well as to label.

Mapped to	p-value	R^2	Adjusted R^2
Score	0.1156	0.2802	0.1902
Label	0.0678	0.3577	0.2774

Our data set exists of acoustic recordings from m students, each of which saw n_{t_i} tasks and solved n_{c_i} tasks correctly. The overall score of a student i in this case is the number of correctly solved tasks n_{c_i} divided by the number of seen tasks n_{t_i}. After applying the above mentioned threshold to the data, we get for each student i the total length of pauses p_i and the total length of speech s_i in his acoustic recoding. Furthermore, we can count connected pause and speech segments to get the number of pause segments n_{p_i} and speech segments n_{s_i} of a student i. The xth pause segment is then $p_i^{(x)}$ and the yth speech segment $s_i^{(y)}$. By means of these measurements and their combination we can create a set of features useful for affect recognition supporting performance prediction and task sequencing:

- Ratio between pauses and speech ($\frac{p_i}{s_i}$)
- Frequency of speech pause changes ($\frac{n_{p_i}+n_{s_i}}{\max_j(n_{p_j}+n_{s_j})}$)
- Percentage of pauses of input speech data ($\frac{p_i}{(p_i+s_i)}$)
- Length of maximal pause segment ($\max_x(p_i^{(x)})$)
- Length of average pause segment ($\frac{\sum_x p_i^{(x)}}{n_{p_i}}$)
- Length of maximal speech segment ($\max_y(s_i^{(y)})$)
- Length of average speech segment ($\frac{\sum_y s_i^{(y)}}{n_{s_i}}$)
- Average number of seconds needed per task ($\frac{(p_i+s_i)}{n_{t_i}}$)

The ratio between the total length of pauses and the total length of speech indicates, if one one them is notable larger than the other one, i.e. if the student made much more speech pauses than speaking or vice versa. The frequency of speech and pause segment changes indicates, if there are many short speech and pauses segments or just a few large ones and it is normalised by dividing it by the maximal sum of pause and speech segments over all students. From the percentage of pauses one can see if the total pause length was much larger than the total speech part, i.e. the student did not speak much but was more thinking silently. The length of maximal pause or speech segment indicates if there was e.g. a very long pause segment where the student was thinking silently or a very long speech segment where the student was in a speech flow. The length of average pause or speech segment give us an idea of how much on average the student was in a silent thinking phase or a speech flow. The average number of seconds needed per task indicates how long a student on average needed for solving a task.

To investigate, if these features are suitable to describe perceived task-difficulty as well as performance in our real data

Table 2: p-value, R^2 and Adjusted R^2 for the best combinations of features (with a p-value smaller than 0.05) of a set with 6, 5, 4 or 3 features mapped to the score.

#	Features	p-val.	R^2	Adj. R^2
6	Frequency of changes, seconds per task, max. length of pause, average length of pause, max. length of speech average length of speech	0.0439	0.9516	0.8548
5	Frequency of changes, seconds per task, max. length of pause, average length of pause, average length of speech	0.0105	0.9496	0.8867
4	Frequency of changes, seconds per task, average length of pause, average length of speech	0.0415	0.8207	0.6773
3	Frequency of changes, frequency of changes, average length of speech	0.0431	0.719	0.5786

Table 3: p-value, R^2 and Adjusted R^2 for the best combinations of features (with a p-value smaller than 0.05) of a set with 5, 4, 3 or 2 features mapped to the labels.

#	Features	p-val.	R^2	Adj. R^2
5	Ratio pause speech, frequency of changes, seconds per task, average length of pause, average length of speech	0.0284	0.9158	0.8106
4	Ratio pause speech, frequency of changes, average length of pause, average length of speech	0.0154	0.8818	0.7872
3	Ratio pause speech, frequency of changes, average length of speech	0.0117	0.8207	0.7311
2	Frequency of changes, average length of speech	0.0327	0.6238	0.5163

set, we mapped the values of each feature to the score as well as to the perceived task-difficulty labels. Subsequently, we applied a linear regression to measure the p-value as well as the R^2 and Adjusted R^2 value. However, as expected, single features are not very significant. The feature with the best values for p-value, R^2 and Adjusted R^2 – mapped to score as well as to labels – is the *Length of maximal pause segment*. The statistical values for this feature are shown in table 1. These values are not very satisfactory, as one would desire a p-value smaller than 0.05 and values for R^2 and Adjusted R^2 which are closer to 1. A more reasonable approach is to combine several features instead of considering just one feature. Hence, in the following section we will investigate different combinations of features.

5.2 Feature Combination Analysis

We analysed different combinations of features by applying a multivariate linear regression to them to gain the p-value, R^2 and Adjusted R^2 for these combinations. The investigated combinations are combinations where all features are not strongly correlated, i.e. whenever we had two correlated features we put just one of them into the feature set for that combination. In further steps we removed from the considered feature sets feature by feature. Furthermore, in the multivariate linear regression we mapped the features on the one hand to the score and on the other hand to the labels. The results of the best combinations, i.e. such with a p-value at least smaller than 0.05, are shown in table 2 and 3. For the score there were no combinations with only 2 features with a p-value smaller than 0.05, hence in table 2 we just listed the best combinations with 3 up to 6 features. For the labels instead there were no such combinations, which have a p-value smaller than 0.05, with 6 features, so that in table 3 we only listed the best combinations of 2 up to 5 features. For both (score and labels) there are statistically significant feature combinations. That means that our proposed features are able to describe the score as well as the labels.

6. SUPPORTING PERFORMANCE PREDICTION AND SEQUENCING

As mentioned in the introduction, our goal is to support the performance prediction system and task sequencer of the approach in [12] by affect recognition, or by multimodal input respectively. Hence, in the following we will propose how to realise this support. In figure 3 a block diagram of the approach of supporting performance prediction and task sequencing by means of affect recognition is presented. The approach in [12] is represented in figure 3 by the non-dotted arrows: the performance prediction gets input from former performances and computes by means of the machine learning method matrix factorization predictions for future performances, which are the input for the task sequencer. The task sequencer decides based on the theory of Vygotsky's Zone of Proximal Development from the performance prediction input which task shall be shown next to the student. This process can be supported by the multimodal input as follows:

(1) The additional input for the performance predictor can be the output of the affect recognition, i.e. the perceived task-difficulty labels. In this case the performance predictor can take the perceived task-difficulty of the last task ($T^{(t)}$) to use the following rules for deciding how difficult the next task ($T^{(t+1)}$) should be:
   - If $T^{(t)}$ was too easy (label *under-challenged* or *flow/under-challenged*), then $T^{(t+1)}$ should be harder.
   - If $T^{(t)}$ was appropriate (label *flow*), then $T^{(t+1)}$ should be similar difficult.
   - If $T^{(t)}$ was too hard (label *over-challenged* or *over-challenged/flow*), then $T^{(t+1)}$ should be easier.

(2) The values of the features gained by feature extraction from speech, typed input and mouse click input

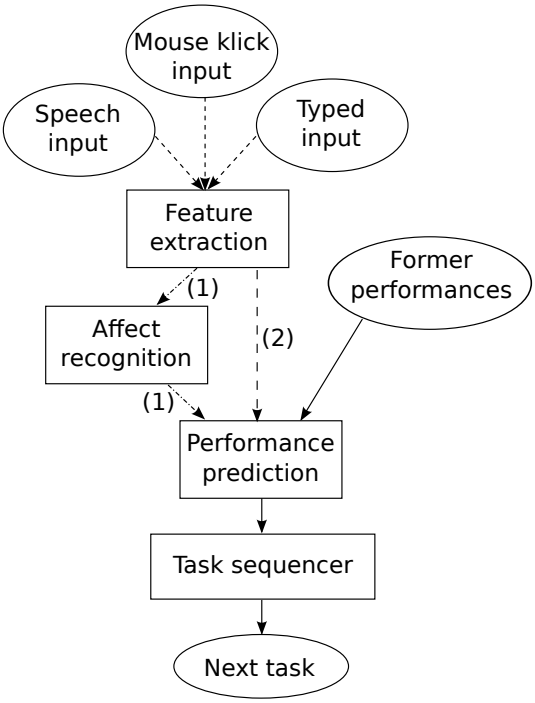


Figure 3: Approach for supporting performance prediction and task sequencing by means of multimodal input and affect recognition.

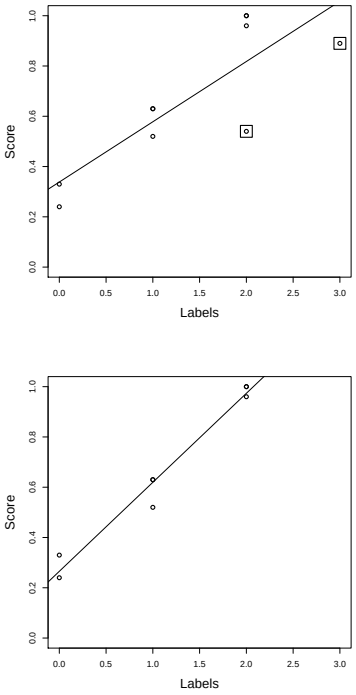


Figure 4: Mapping of the perceived task-difficulty labels to the scores of the students in the real data set (a) with outliers indicated by surrounding rectangles (top) and (b) without outliers (bottom).

can be fed directly into the performance prediction without applying an affect recognition. That means that the features are mapped to scores instead of perceived task-difficulty classes. That this makes sense was shown in section 4 and 5. The performance predictor can then compare e.g. the differences between performances, expressed as *score*, and the scores computed by means of the features ($\widehat{score}$). This difference indicates outliers like if a student felt to be in a flow or under-challenged but his score is worse, i.e. $\widehat{score} > score$. In this case the student may not fully understand the principles of the considered task although he thinks so. Hence, next the system should show the student rather tasks which explain the approach of solving such kind of tasks.

In our studies we observed the behaviour of students described in (2), i.e. the student was labelled as to be in a flow or under-challenged, although he performed worse, as he just thought to understand how the tasks should be solved but he was wrong. In figure 4 this behaviour is indicated by the outliers.

7. LABELLING AND DATA COLLECTION

As mentioned in section 3 the labels of our real data set come from two sources: (a) a concurrent annotation by the tutor and (b) a retrospective annotation by another external reviewer on the basis of the tasks sheet, the sound files and the screen recording. However, in the literature one can find further labelling strategies like self-labelling of the students (see e.g. [5], [6], [8]). The advantage of self-labelling is that one can gain automatically a labelled data set for a subsequent training of an affect recognition method. Furthermore, as we want to recognise the perceived task-difficulty from the view of the student, a label from the student himself seem to be more appropriate than labels from another person only reviewing the behaviour of the student. Hence, for further studies we developed a tool for collecting speech data and typed input and mouse click input data, labelled automatically with the task-difficulty perceived by the student. This tool will be further described in the following section.

7.1 Self-Labelling Fractional Arithmetic Tutor for Multimodal Data Collection

To be able to conduct studies in which the students themselves label the task-difficulty which they perceived, we developed a tutoring tool (*self* - **se**lf-**l**abelling **f**ractional arithmetic tutor for multimodal data collection) written in Java. However, for little children it might be difficult to analyse themselves (see e.g. [8]). Hence, self-labelling is often applied in experiments with at least college students as for instance in [5]. Therefore, we will conduct the experiments with this tool first with older students and more challenging tasks. Later on we will investigate if there is a way to adapt the tool so that a self-labelling is possible also with younger students. Nevertheless, conducting experiments with older students has several advantages besides the possibility of a reasonable self-labelling: older students are able to focus on the tasks longer than young students and the privacy issues are not such strong as for younger students. Both facts lead to more data. Hence, besides investigating the possibility of

adapting *self* for younger students, we have to identify differences and similarities of the data from older and younger students to find out how to exploit older students data to recognise affects from multimodal input from younger students.

In figure 5 one can see the graphical user interface of our self-labelling multimodal data collection tool *self*. To gain more background information, in the beginning *self* asks some information from the students as course of studies, number of terms, age and gender. Subsequently, an instruction with hints how to behave is shown to the students, which they can have a look at also while interacting with the tool (button "Anleitung"). *self* speaks to the students to motivate them to speak with the system and records the speech input of the students. The speech output of *self* is generated by means of *text to speech* realised by the library MARY developed at the DFKI ([18]). While interacting with the system, the student can type in numbers, ask for a hint (button "Hilfe"), skip the task because it is too easy or because it is too hard (left buttons) or submit the solution (button "Endergebnis überprüfen"). Every action of the student, like asking for a hint or submitting the answer, is written – together with a time stamp – into a log file immediately after the action, enabling also the extraction of typed input or mouse click input features. Also a score depending on the number of requested hints h_r and the number of incorrect inputs w is computed according to the approach in [15] and written into the log file. The formula for this score is

$$1 - (\frac{h_r}{h_t} + (w \cdot 0.1)) \,, \tag{2}$$

where h_t is the total number of available hints for the considered task. The meaning behind the formula is that each wrong input $w^{(j)}$ is punished with a factor of 0.1 and every request of a hint $h_r^{(k)}$ is punished with a factor of $\frac{1}{h_t}$, so that if every hint was seen the score will be 0. After the student submitted the correct answer, he is asked to evaluate, if this task was too easy, too hard or appropriate for him (see pop-up window in figure 5). The tasks implemented in *self* for older students cover the following areas:

- Reducing fractions with numbers and variables
- Fraction addition with and without intermediate steps and with numbers and variables
- Fraction subtraction with and without intermediate steps and with numbers and variables
- Fraction multiplication with and without intermediate steps and with numbers and variables
- Fraction division with and without intermediate steps and with numbers and variables
- Distributivity law with and without intermediate steps
- Finite sums of unit fractions
- Rule of Three

After developing *self*, the next step will be to conduct further studies with students to collect an adequate amount of automatically labelled speech input, typed input and mouse click input data for training an affect recognition method and supporting performance prediction and task sequencing. Furthermore, we will investigate if there is a way to adapt *self* so that also younger students can label themselves.

8. CONCLUSIONS

We proposed a new approach for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems by affect recognition on features gained from multimodal input like students speech input. For this approach we proposed and analysed appropriate speech features and showed that there are statistically significant feature combinations which are able to describe students affect, or perceived task-difficulty respectively, as well as the performance of a student. Furthermore, we proved the possibility of supporting performance prediction and task sequencing by perceived task-difficulties by demonstrating that there is a correlation between perceived task-difficulty and performance. Next steps will be to conduct more studies with students by means of the presented self-labelling and multimodal data collection tool to enable a training of an appropriate affect recognition method for supporting performance prediction and task sequencing in adaptive intelligent tutoring systems.

9. ACKNOWLEDGMENTS

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/2007 – 2013) under grant agreement No. 318051 – iTalk2Learn project (www.italk2learn.eu). Furthermore, we thank our project partner Ruhr University Bochum for realising the study and data collection as well as the IMAI of the University of Hildesheim for support for the tutoring tool and preparation for future studies.

10. REFERENCES

[1] Cichocki, A., Zdunek, R., Phan, A. H. and Amari, S.I. 2009. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, Wiley.

[2] Epp, C., Lippold, M., Mandryk, R.L. 2011. Identifying Emotional States Using Keystroke Dynamics. In Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI 2011), Vancouver, BC, Canada, pp. 715–724.

[3] Hu, X., Tang, L., Tang, J. and Liu, H. 2013. Exploiting Social Relations for Sentiment Analysis in Microblogging. In Proceedings of the Sixth ACM WSDM Conference (WSDM '13).

[4] Luz, S. 2013. Automatic Identification of Experts and Performance Prediction in the Multimodal Math Data Corpus through Analysis of Speech Interaction. Second International Workshop on Multimodal Learning Analytics, Sydney Australia, December 2013.

[5] D'Mello, S., Picard, R. and Graesser, A. 2007. Towards An Affect-Sensitive AutoTutor. Intelligent Systems, IEEE, Vol. 22, Issue 4, pp. 53–61.

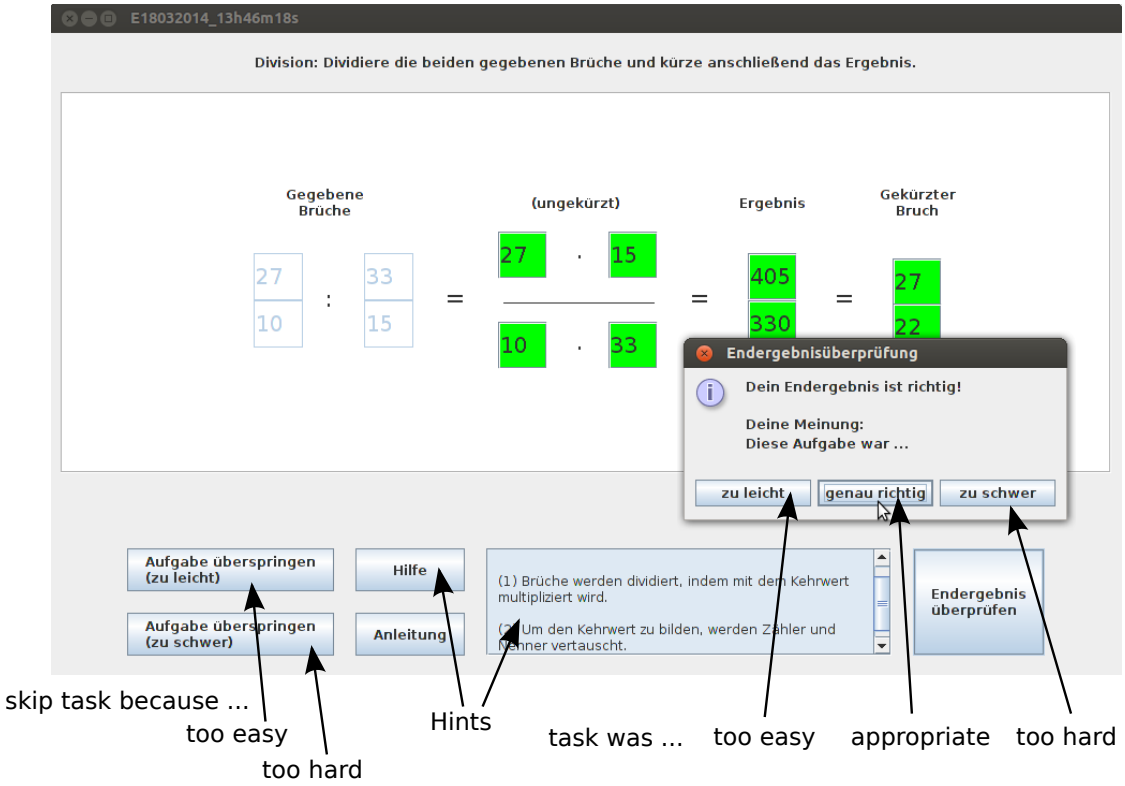


Figure 5: Graphical user interface of the developed fractional arithmetic tutoring tool *self* for self-labelling as well as for speech data and typed input or mouse click input data collection.

[6] D'Mello, S.K., Craig, S.D., Witherspoon, A., McDaniel, B. and Graesser, A. 2008. Automatic detection of learner's affect from conversational cues. User Model User-Adap Inter, DOI 10.1007/s11257-007-9037-6.

[7] Morency, L.P., Oviatt, S., Scherer, S., Weibel, N. and Worsley, M. 2013. ICMI 2013 grand challenge workshop on multimodal learning analytics. In Proceedings of the 15th ACM on International conference on multimodal interaction (ICMI 2013), pp. 373–378.

[8] Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C. and Baker, R.S.J.d. 2013. Knowledge Elicitation Methods for Affect Modelling in Education. International Journal of Artificial Intelligence in Education, ISSN 1560-4292.

[9] Qi, F., Bao, C., Liu, Y. 2004. A novel two-step SVM classifier for voiced/unvoiced/silence classification of speech. International Symposium on Chinese Spoken Language Processing, pp. 77 – 80.

[10] Sadegh, M., Ibrahim, R., Othman, Z.A. 2012. Opinion Mining and Sentiment Analysis: A Survey. International Journal of Computers & Technology, Vol. 2, No. 3.

[11] Saif, H., He, Y. and Alani, H. 2012. Semantic Sentiment Analysis of Twitter. In Proceedings of the 11th International Semantic Web Conference (ISWC 2012).

[12] Schatten, C. and Schmidt-Thieme, L. 2014. Adaptive Content Sequencing without Domain Information. In Proceedings of the Conference on computer supported education (CSEDU 2014).

[13] Schuller, B., Batliner, A., Steidl, S. and Seppi, D. 2011. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. Speech Communication, Elsevier.

[14] Vygotsky, L.L.S. 1978. Mind in society: The development of higher psychological processes. Harvard university press.

[15] Wang, Y. and Heffernan, N. 2011. Extending Knowledge Tracing to allow Partial Credit: Using Continuous versus Binary Nodes. Artificial Intelligence in Education, Lecture Notes in Computer Science, Vol. 7926, pp. 181–188.

[16] Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D. and Picard, R. 2009. Affect-aware tutors: recognising and responding to student affect. Int. J. of Learning Technology, Vol. 4, No. 3/4, pp. 129–164.

[17] Worsley, M. and Blikstein, P. 2011. What's an Expert? Using Learning Analytics to Identify Emergent Markers of Expertise through Automated Speech, Sentiment and Sketch Analysis. In Proceedings of the 4th International Conference on Educational Data Mining (EDM '11), pp. 235–240.

[18] The MARY Text-to-Speech System, http://mary.dfki.de/

Collaborative Assessment

Patricia Gutierrez
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
patricia@iiia.csic.es

Nardine Osman
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
nardine@iiia.csic.es

Carles Sierra
IIIA-CSIC
Campus de la UAB
Barcelona, Spain
sierra@iiia.csic.es

ABSTRACT

In this paper we introduce an automated assessment service for online learning support in the context of communities of learners. The goal is to introduce automatic tools to support the task of assessing massive number of students as needed in Massive Open Online Courses (MOOC). The final assessments are a combination of tutor's assessment and peer assessment. We build a trust graph over the referees and use it to compute weights for the assesments aggregations. The model proposed intends to be a support for intelligent online learning applications that encourage student's interactions within communities of learners and benefits from their feedback to build trust measures and provide automatic marks.

1. INTRODUCTION

Self and peer assessment have clear pedagogical advantages. Students increase their responsibility and autonomy, get a deeper understanding of the subject, become more active in the learning process, reflect on their role in group learning, and improve their judgement skills. Also, it may have the positive side effect of reducing the marking load of tutors. This is specially critical when tutors face the challenge of marking large quantities of students as needed in the increasingly popular Massive Open Online Courses (MOOC).

Online learning communities encourage different types of peer-to-peer interactions along the learning process. These interactions permit students to get more feedback, to be more motivated to improve, and to compare their own work with other students accomplishments. Tutors, on the other hand, benefit from these interactions as they get a clearer perception of the student engagement and learning process.

Previous works have proposed different methods of peer assessment as part of the learning process with the added advantage of helping tutors in the sometimes dauting task of marking large quantities of students [7, 3].

The authors of [7] propose methods to estimate peer reliability and correct peer biases. They present results over real world data from 63,000 peer assessments of two Coursera courses. The models proposed are probabilistic and they are compared to the grade estimation algorithm used on Coursera's platform, which does not take into account individual biases and reliabilities. Differently from them, we place more trust in students who grade like the tutor and do not consider student's biases. When a student is biased its trust measure will be very low and his/her opinion will have a moderate impact over the final marks.

[3] proposes the CrowdGrader framework, which defines a crowdsourcing algorithm for peer evaluation. The accuracy degree (i.e. reputation) of each student is measured as the distance between his/her self assesment and the aggregated opinion of the peers weighted by their accuracy degrees. The algorithm thus implements a reputation system for students, where higher accuracy leads to higher influence on the consensus grades. Differently from this work, we give more weight to those peers that have similar opinions to those of the tutor.

In this paper, and differently from previous works, we want to study the *reliability* of student assessments when compared with tutor assessments. Although part of the learning process is that students participate in the definition of the evalution criteria, tutors want to be certain that the scoring of the students' works is fair and as close as possible to his/her expert opinion.

Our inspiration comes from a use case explored in the EU-funded project PRAISE [1]. PRAISE enables online virtual communities of students with shared interests and goals to come together and share their music practice with each other so the process of learning becomes social. It provides tools for giving and receiving feedback, as feedback is considered an essential part of the learning process. Tutors define *lesson plans* as pedagogical workflows of activities, such as uploading recorded songs, automatic performance analysis, peer feedback, or reflexive pedagogy analysis. The goal of any lesson plan is to improve student skills, for instance, the performance speed competence or the interpretation maturity level. Assessments of students' performances have to evaluate the achievement of these skills. Once a lesson plan is defined, PRAISE's interface tools allow students to navigate through the activities, to upload assignments, to practice, to assess each other, and so on. The tools allow tutors to monitor what students have done and to assess them. In this

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

work we concentrate on the development of a service that can be included as part of a lesson plan and helps tutors in the overall task of assessing the students participating in the lesson plan. This assessment is based on aggregating students' assessments, taking into consideration the trust that tutors have on the students' individual capabilities in judging each others work.

To achieve our objective we propose in this paper an automated assessment method (Section 2) based on *tutor assessments*, aggregations of *peer assessments* and on *trust measures* derived from peer interactions. We experimentaly evaluate (Section 3) the accuracy of the method over different topologies of student interactions (i.e. different types of student grouping). The results obtained are based on simulated data, leaving the validation with real data for future work. We then conclude with a discussion of the results (Section 4).

2. COLLABORATIVE ASSESSMENT

In this section we introduce the formal model of the method and the algorithms for collaborative assessment.

2.1 Notation and preliminaries

We say an online course has a tutor τ, a set of peer students $\mathcal{S}$, and a set of assignments $\mathcal{A}$ that need to be marked by the tutor and/or students with respect to a given set of criteria $\mathcal{C}$.

The automated assessment state S is then defined as the tuple:

$$S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$$

$R = \{\tau\} \cup \mathcal{S}$ defines the set of possible referees (or markers), where a referee could either be the tutor τ or some student $s \in S$. $\mathcal{A}$ is the set of submitted assignments that need to be marked and $\mathcal{C} = \langle c_1, \ldots, c_n \rangle$ is the set of criteria that assignments are marked upon. $\mathcal{L}$ is the set of marks (or assessments) made by referees, such that $\mathcal{L} : R \times A \to [0, \lambda]^n$ (we assume marks to be real numbers between 0 and some maximum value λ). In other words, we define a single assessment as: $\mu_\alpha^\rho = \vec{M}$, where $\alpha \in \mathcal{A}$, $\rho \in R$, and $\vec{M} = \langle m_1, \ldots, m_n \rangle$ describes the marks provided by the referee on the n criteria of $\mathcal{C}$, $m_i \in [0, \lambda]$.

Similarity between marks. We define a similarity function $sim : [0, \lambda]^n \times [0, \lambda]^n \to [0, 1]$ to determine how close two assesments μ_α^ρ and μ_α^η are. We calculate the similarity between assessments $\mu_\alpha^\rho = \{m_1, \ldots, m_n\}$ and $\mu_\alpha^\eta = \{m'_1, \ldots, m'_n\}$ as follows:

$$sim(\mu_\alpha^\rho, \mu_\alpha^\eta) = 1 - \frac{\sum_{i=1}^{n} |m_i - m'_i|}{\sum_{i=1}^{n} \lambda}$$

This measure satisfies the basic properties of a fuzzy similarity [6]. Other similarity measures could be used.

Trust relations between referees. Tutors need to decide up to which point they can believe on the assessments made by peers. We use two different intuitions to make up this belief. First, if the tutor and the student have both assessed some assigments, their similarity gives a hint of how close the judegements of the student and the tutor are. Similarly, we can define the judgement closeness of any two students by looking into the assignments evaluated by both of them. In case there are no assigments evaluated by the tutor and one particular student we could simply not take that student's opinion into account because the tutor would not know how much to trust the judgement of this student, or, as we do in this paper, we approximate that unknown trust by lookig into the chain of trust between the tutor and the student through other students. To model this we define two different types of trust relations:

- *Direct trust*: This is the trust between referees $\rho, \eta \in R$ that have at least one assignement assessed in common. The trust value is the average of similarities on the assessments over the same peers. Let the set $A_{\rho,\eta}$ be the set of all assignments that have been assessed by both referees. That is, $A_{\rho,\eta} = \{\alpha \mid \mu_\alpha^\rho \in \mathcal{L} \text{ and } \mu_\alpha^\eta \in \mathcal{L}\}$. Then,

  $$T_D(\rho, \eta) = \frac{\sum_{\alpha \in A_{\rho,\eta}} sim(\mu_\alpha^\rho, \mu_\alpha^\eta)}{|A_{\rho,\eta}|}$$

  We could also define direct trust as the conjunction of the similarities for all common assignments as:

  $$T_D(\rho, \eta) = \bigwedge_{\alpha \in A_{\rho,\eta}} sim(\mu_\alpha^\rho, \mu_\alpha^\eta)$$

  However, this would not be practical, as a significant difference in just one assessment of those assessed by two referees would make their mutual trust very low.

- *Indirect trust*: This is the trust between referees $\rho, \eta \in R$ without any assignement assessed by both of them. We compute this trust as a transitive measure over chains of referees for which we have pair-wise direct trust values. We define a trust chain as a sequence of referees $q_j = \langle \rho_i, ..., \rho_i, \rho_{i+1}, \ldots, \rho_{m_j} \rangle$ where $\rho_i \in R$, $\rho_1 = \rho$ and $\rho_{m_j} = \eta$ and $T_D(\rho_i, \rho_{i+1})$ is defined for all pairs (ρ_i, ρ_{i+1}) with $i \in [1, m_j - 1]$. We note by $Q(\rho, \eta)$ the set of all trust chains between ρ and η. Thus, indirect trust is defined as a aggregation of the direct trust values over these chains as follows:

  $$T_I(\rho, \eta) = \max_{q_j \in Q(\rho,\eta)} \prod_{i \in [1, m_j - 1]} T_D(\rho_i, \rho_{i+1})$$

  Hence, indirect trust is based in the notion of transitivity.[1]

[1] T_I is based on a fuzzy-based similarity relation sim presented before and fulfilling the $\otimes$-Transitivity property: $sim(u, v) \otimes sim(v, w) \leq sim(u, w)$, $\forall u, v, w \in V$, where $\otimes$ is a t-norm [6].

Ideally, we would like to not overrate the trust of a tutor on a student, that is, we would like that $T_D(a,b) \geq T_I(a,b)$ in all cases. Guaranteeing this in all cases is impossible, but we can decrease the number of overtrusted students by selecting an operator that gives low values to T_I. In particular, we prefer to use the product $\prod$ operator, because this is the t-norm that gives the smallest possible values. Other opertors could be used, for instance the *min* function.

Trust Graph. To provide automated assessments, our proposed method agregates the assessments on a given assignment taking into consideration how much trusted is each marker/referee from the point of view of the tutor (i.e. taking into consideration the trust of the tutor on the referee in marking assignments). The algorithm that computes the student final assessment is based on a graph defined as follows:

$$G = \langle R, E, w \rangle$$

where the set of nodes R is the set of referees in S, $E \subseteq R \times R$ are edges between referees with direct or indirect trust relations, and $w : E \to [0,1]$ provides the trust value. We note by $D \subset E$ the set of edges that link referees with direct trust. That is, $D = \{e \in E | T_D(e) \neq \perp\}$. An similarly, $I \subset E$ for indirect trust, $I = \{e \in E | T_I(e) \neq \perp\} \setminus D$. The w values will be used as weights to combine peer assessments and are defined as:

$$w(e) = \begin{cases} T_D(e) & \text{, if } e \in D \\ T_I(e) & \text{, if } e \in I \end{cases}$$

Figure 1 shows examples of trust graphs with $e \in D$ (in black) and $e \in I$ (in red —light gray) for different sets of assessments $\mathcal{L}$.

2.2 Computing collaborative assessments

Algorithm 1 implements the collaborative assessment method. We keep the notation (ρ, η) to refer to the edge connecting nodes ρ and η in the trust graph and $Q(\rho, \eta)$ to refer the set of trust chains between ρ and η.

The first thing the algorithm does is to build a trust graph from $\mathcal{L}$. Then, the final assessments are computed as follows. If the tutor marks an assignment, then the tutor mark is considered the final mark. Otherwise, a weighted average (μ_α) of the marks of student peers is calculated for this assignment, where the weight of each peer is the trust value between the tutor and that peer. Other forms of aggregation could be considered to calculate μ_α, for instance a peer assessment may be discarded if it is very far from the rest of assessments, or if the referee's trust falls below a certain threshold.

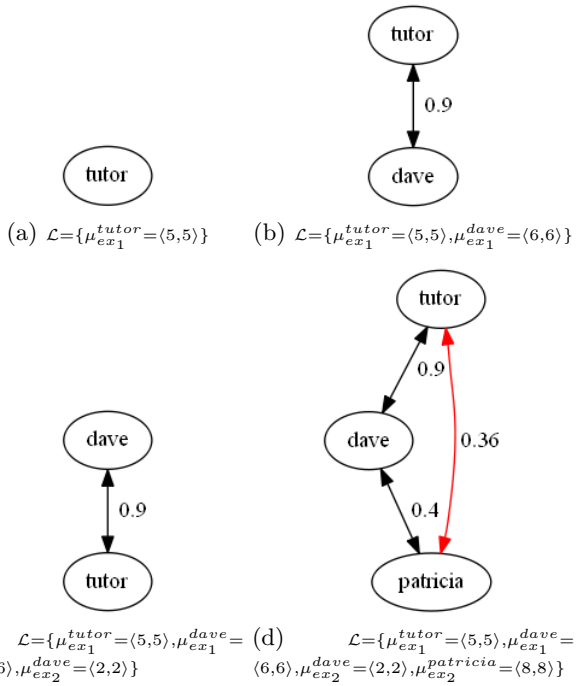
Figure 1 shows four trust graphs built from four assessments histories that corresponds to a chronological sequence of assessments made. The criteria $\mathcal{C}$ in this example are *speed* and *maturity* and the maximum mark value is $\lambda = 10$. For

Algorithm 1: collaborativeAssessments($S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$)

▷ Initial trust between referees is zero
$D = I = \emptyset$;
for $\rho, \eta \in \mathcal{R}, \rho \neq \eta$ **do**
  $w(\rho, \eta) = 0$;
end
▷ Update direct trust and edges
for $\rho, \eta \in \mathcal{R}, \rho \neq \eta$ **do**
  $A_{\rho,\eta} = \{\beta \mid \mu^\rho_\beta \in \mathcal{L} \text{ and } \mu^\eta_\beta \in \mathcal{L}\}$;
  if $|A_{\rho,\eta}| > 0$ **then**
    $D = D \cup (\rho, \eta)$;
    $w(\rho, \eta) = T_D(\rho, \eta)$;
  end
end
▷ Update indirect trust and edges between tutor & students
for $\rho \in \mathcal{R}$ **do**
  if $(\tau, \rho) \notin D$ *and* $Q(\tau, \rho) \neq \emptyset$ **then**
    $I = I \cup (\rho, \eta)$;
    $w(\rho, \eta) = T_I(\tau, \eta)$;
  end
end
▷ Calculate automated assessments
$assessments = \{\}$;
for $\alpha \in A$ **do**
  if $\mu^\tau_\alpha \in \mathcal{L}$ **then**
    ▷ Tutor assessments are preserved
    $assessments = assessments \cup (\alpha, \mu^\tau_\alpha)$
  else
    ▷ Generate automated assessments
    $R' = \{\rho | \mu^\rho_\alpha \in \mathcal{L}\}$;
    if $|R'| > 0$ **then**
      $\mu_\alpha = \frac{\sum_{\rho \in R'} \mu^\rho_\alpha * w(\tau, \rho)}{\sum_{\rho \in R'} w(\tau, \rho)}$;
      $assessments = assessments \cup (\alpha, \mu_\alpha)$;
    end
  end
end
return *assessments*;

simplicity we only represent those referees that have made assessments in $\mathcal{L}$. In Figure 1(a) there is one node representing the tutor who has made the only assessment over the assignment ex_1 and there are no links to other nodes as no one else has assessed anything. In (b) student Dave assesses the same exercise as the tutor and thus a link is created between them. The trust value $w(tutor, Dave) = T_D(tutor, Dave)$ is high since their marks were similar. In (c) a new assessment by Dave is added to $\mathcal{L}$ with no consequences in the graph construction. In (d) student Patricia adds an assessment on ex_2 that allows to build a direct trust between Dave and Patricia and an indirect trust between the tutor and Patricia, through Dave. The automated assessments generated in case (d) are: $\langle 5, 5 \rangle$ for exercise 1 (which preserves the tutor's assessment) and $\langle 3.7, 3.7 \rangle$ for exercise 2 (which uses a weighted aggregation of the peers' assessments).

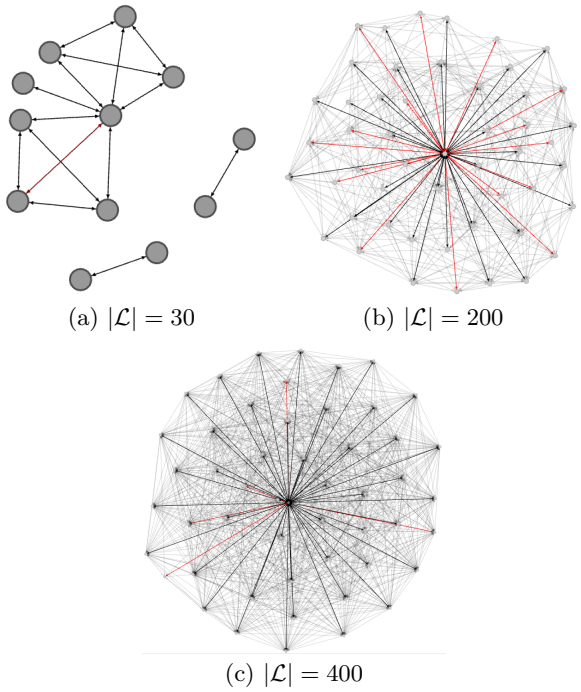
Note that the trust graph built from $\mathcal{L}$ is not necessarily connected. A tutor wants to reach a point in which the graph is totally connected because that means that the collaborative assessment algorithm generates an assessment for every assignment. Figure 2 shows an example of a trust graph of a particular learning community involving 50 peer students and a tutor. When S has a history of 5 tutor assessments and 25 student assessments ($|\mathcal{L}| = 30$) we observe that not all nodes are connected. As the number of assessments in-



(a) $\mathcal{L}=\{\mu_{ex_1}^{tutor}=\langle 5,5\rangle\}$ (b) $\mathcal{L}=\{\mu_{ex_1}^{tutor}=\langle 5,5\rangle,\mu_{ex_1}^{dave}=\langle 6,6\rangle\}$

(c) $\mathcal{L}=\{\mu_{ex_1}^{tutor}=\langle 5,5\rangle,\mu_{ex_1}^{dave}=\langle 6,6\rangle,\mu_{ex_2}^{dave}=\langle 2,2\rangle\}$ (d) $\mathcal{L}=\{\mu_{ex_1}^{tutor}=\langle 5,5\rangle,\mu_{ex_1}^{dave}=\langle 6,6\rangle,\mu_{ex_2}^{dave}=\langle 2,2\rangle,\mu_{ex_2}^{patricia}=\langle 8,8\rangle\}$

Figure 1: Trust graph example 1.



(a) $|\mathcal{L}| = 30$ (b) $|\mathcal{L}| = 200$

(c) $|\mathcal{L}| = 400$

Figure 2: Trust graph example 2

creases, the trust graph becomes denser and eventually it gets completely connected. In (b) and (c) we see a complete graph.

3. EXPERIMENTAL PLATFORM AND EVALUATION

In this Section we describe how we generate simulated social networks, describe our experimental platform, define our benchmarks and discuss experimental results.

3.1 Social Network Generation

Several models for social network generation have been proposed reflecting different characteristics present in real social communities. Topological and structural features of such networks have been explored in order to understand wich generating model resembles best the structure of real communities [5].

A social network can be defined as a graph $\mathcal{N}$ where the set of nodes represent the individuals of the network and the set of edges represent connections or social ties among those individuals. In our case, individuals are the members of the learning community: the tutor and students. Connections represent the social ties and they are usually the result of interactions in the learning community. For instance a social relation will be born between two students if they interact with each other, say by collaboratively working on a project together. In our experimentation, we rely on the social network in order to simulate which student will assess the assignment of which other student. We assume students will assess the assignments of students they know, as opposed to picking random assignments. As such, we clarify that social networks are different from the trust graph of Section 2. While the nodes of both graphs are the same, edges of the social network represent social ties, whereas edges in the trust graph represent how much does one referee trust another in judging others work.

To model social networks where relations represent social ties, we follow three different approaches: the Erdős-Rényi model for random networks [4], the Barabási-Albert model for power law networks[2] and a hierarchical model for cluster networks.

3.1.1 *Random Networks*

The Erdős-Rényi model for random networks consists of a graph containing n nodes connected randomly. Each possible edge between two vertices may be included in the graph with probability p and may not be included with probability $(1-p)$. In addition, in our case there is always an edge between the node representing the tutor and the rest of nodes, as the tutor knows all of its students (and may eventually mark any of those students).

The degree distribution of random graphs follows a Poisson distribution. Figure 3(a) shows an example of a random graph with 51 nodes and $p = 0.5$ and its degree distribution. Note that the point with degree 50 represents the tutor node while the rest of the nodes degree fit a Poisson distribution.

3.1.2 *Power Law Networks*

The Barabási-Albert model for power law networks base their graph generation on the notions of *growth* and *preferential attachment*. The generation scheme is as follows. Nodes are added one at a time. Starting with a small number of initial nodes, at each time step we add a new node with m edges linked to nodes already part of the network. In our experiments, we start with $m+1$ initial nodes. The

edges are not placed uniformly at random but preferentially in proportion to the degree of the network nodes. The probability p that the new node is connected to a node i already in the network depends on the degree k_i of node i, such that: $p = k_i / \sum_{j=1}^{n} k_j$. As above, there is also always an edge between the node representing the tutor and the rest of nodes.

The degree distribution of this network follows a Power Law distribution. Figure 3(b) shows an example of a power law graph with 51 nodes and $m = 16$ and its degree distribution. The point with degree 50 describes the tutor node while the rest of the nodes closely resemble a power law distribution. Recent empirical results on large real-world networks often show, among other features, their degree distribution following a power law [5].

3.1.3 *Cluster Networks*

As our focus is on learning communities, we also experiment with a third type of social network: the cluster network which is based on the notions of *groups* and *hierarchy*. Such networks consists of a graph composed of a number of fully connected clusters (where we believe clusters may represent classrooms or similar pedagogical entities). Additionally, as above, all the nodes are connected with the tutor node. Figure 3(c) shows an example of a cluster graph with 51 nodes, 5 clusters of 10 nodes each and its degree distribution. The point with degree 50 describes the tutor while the rest of the nodes have degree 10, since every student is fully connected with the rest of the classroom.

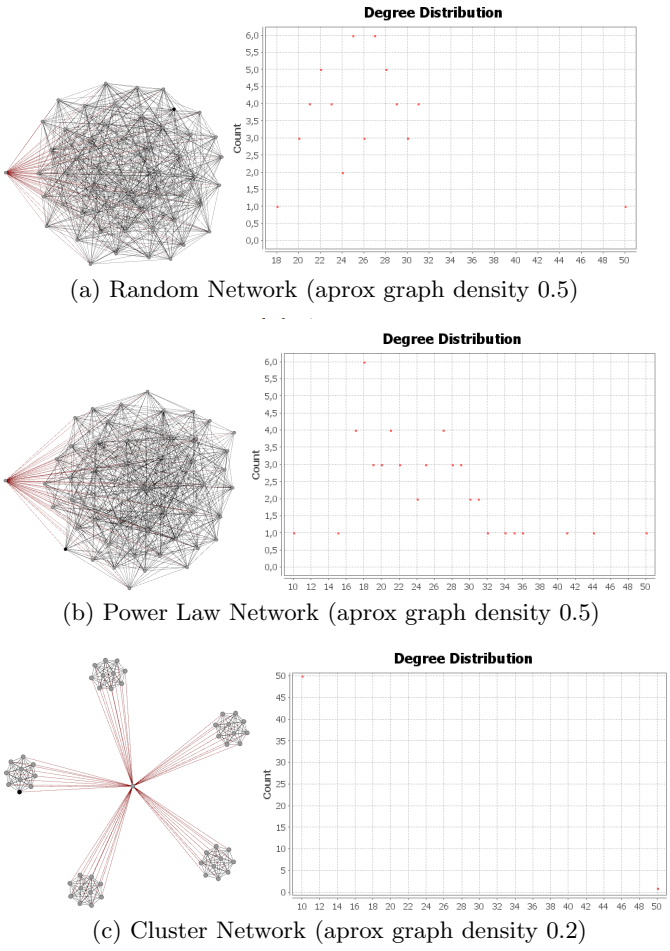
3.2 Experimental Platform

In our experimentation, given an initial automated assessment state $S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$ with an empty set of assessments $\mathcal{L} = \{\}$, we want to simulate tutor and peer assessments so that the collaborative assessment method can eventually generate a reliable and definitive set of assessments for all assignments.

To simulate assessments, we say each students is defined by its profile that describes how good its assessments are. The profile is essentially defined by the measure, or distance, $d_\rho \in [0, 1]$ that specifies how close are the student's assessments to that of the tutor.

We then assume the simulator knows how the tutor and each student would assess an assignment. This becomes necessary in our simulation, since we generate student assessments in terms of their distance to that of the tutor's, even if the tutor does not choose to actually assess the assignment in question. This simulator's knowledge of the values of all possible assessments is generated accordingly:

- For every assignment $\alpha \in \mathcal{A}$, we calculate the tutor's assessment, which is randomly generated according to the function $f_\tau : \mathcal{A} \to [0, \lambda]^n$. This assessment essentially describes what mark would the tutor give α, if it decided to assess it.
- For every assignment $\alpha \in \mathcal{A}$, we also calculate the assessment of each student $\rho \in \mathcal{S}$. This is calculated according to the function $f_\rho : A \to [0, \lambda]^n$, such that: $sim(f_\rho(\alpha), f_\tau(\alpha)) \geq d_\rho$ We note that we only need to calculate ρ's assessment of α if the student who submitted the assignment α is a neighbour of ρ in $\mathcal{N}$.



(a) Random Network (aprox graph density 0.5)

(b) Power Law Network (aprox graph density 0.5)

(c) Cluster Network (aprox graph density 0.2)

Figure 3: Social Network generation examples

We note that the above only calculates what the assessments would be, if referees where to assess assignments.

3.3 Benchmark

Given an initial automated assessment state $S = \langle R, \mathcal{A}, \mathcal{C}, \mathcal{L} \rangle$ with an empty set of assessments $\mathcal{L} = \{\}$, a set of student profiles $Pr = \{d_s\}_{\forall s \in \mathcal{S}}$, and a social network $\mathcal{N}$ (whose nodes is the set R), we simulate individual tutor and students' assessments. When does a referee in R assess an assignment in $\mathcal{A}$ is explained shortly. However we note here that the value of each generated assessment is equivalent to that calculated for the simulator's knowledge (see Section 3.2 above).

In our benchmark, we consider the three types of social networks introduced earlier: random social networks (with 51 nodes, $p = 0.5$, and approximate density of 0.5), power law networks (with 51 nodes, $m = 16$, and approximate density of 0.5), and cluster networks (with 51 nodes, 5 clusters of 10 nodes each, and approximate density of 0.2). Examples of these generated networks are shown in Figure 3.

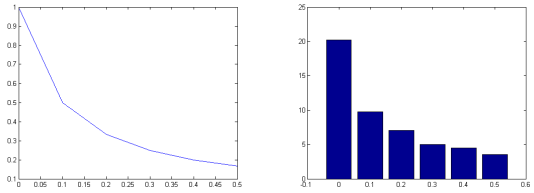
We say one assignment is submitted by each student, resulting in $|\mathcal{S}| = 50$ and $|\mathcal{A}| = 50$. The range that a referee (tutor or student) may mark a given assignment with respect to a given criteria is [0,10]. And the set of criteria is $\mathcal{C} = \langle speed, maturity \rangle$. The criteria essentially measure the *speed* of playing a musical piece, and the *maturity level* of the student's performance.

An assessment profile is generated for each student ρ at the beginning of the execution, resulting in a set of student profiles $Pr = \{d_s\}_{\forall s \in \mathcal{S}}$, where $d \in [0, 0.5]$. We consider here two cases for generating the set of student profiles Pr. A first case where d is picked randomly following a power law distribution (Figure 4(a)) and a second case where d is picked randomly following a uniform distribution (Figure 4(b)).

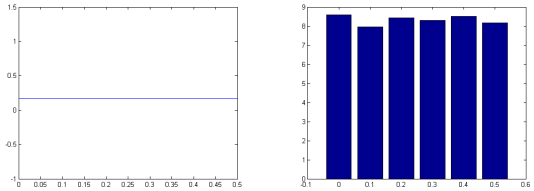
With simulated individual assessments, we then run the collaborative assessment method in order to compute an automated assessment. We also compute the 'error' of the collaborative assessment method, whose range is $[0, 1]$, over the set of assignments $\mathcal{A}$ accordingly:

$$\frac{\sum_{\alpha \in \mathcal{A}} sim(f_\tau(\alpha), \phi(\alpha))}{|\mathcal{A}|}$$

, where $\phi(\alpha)$ describes the automated assessment for a given assignment $\alpha \in \mathcal{A}$



(a) Power law profile generation



(b) Uniform profile generation

Figure 4: Example of the profile distributions (left) and of d counting averaged over 50 instances (right)

With the settings presented above, we run two different experiments. The results presented are an average over 50 executions. The two experiments are presented next.

In experiment 1, students provide their assessments before the tutor. Each student ρ provides assessments for a randomly chosen a_ρ number of peer assigments (of course, where assignments are those of their neighboring peers in $\mathcal{N}$). We run the experiment for 5 different values of $a_\rho = \{3, 4, 5, 6, 7\}$. After the students provide their assessments, the tutor starts assessing assignments incrementally. After every tutor assessment, the error over the set of automated assessment is calculated. Notice that the collaborative assessment method takes the tutor assessment, when it exists, to be the final assessment. As such, the number of automated assessments calculated based on aggregating students' assessments is reduced over time. Finally, when the tutor has assessed all 50 students, the resulting error is 0.

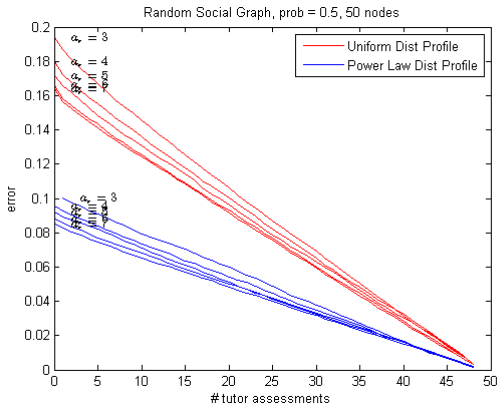
In experiment 2, the tutor provides its assessments before the students. The tutor in this experiment will assess a randomly chosen number of assignments, where this number is based on the percentage a_τ of the total number of assignments. We run the experiment for 4 different values of $a_\tau = \{5, 10, 15, 20\}$. After the tutor provides their assessments, students' assessments are performed. In every iteration, a student ρ randomly selects a neighbor in $\mathcal{N}$ and assesses his assignment (in case it has not been assessed before by ρ, otherwise another connected peer is chosen). We note that in the case of random and power law networks (denser networks), a total number of 1000 student assessments are performed. Whereas in the case of cluster networks (looser network), a total of 400 student assessments are performed. We note that initially, the trust graph is not fully connected, so the service is not able to provide automated assessments for all assignments. When the grap gets fully connected, the service generates automated assessments for all assignments and we start measuring the error after every new iteration.

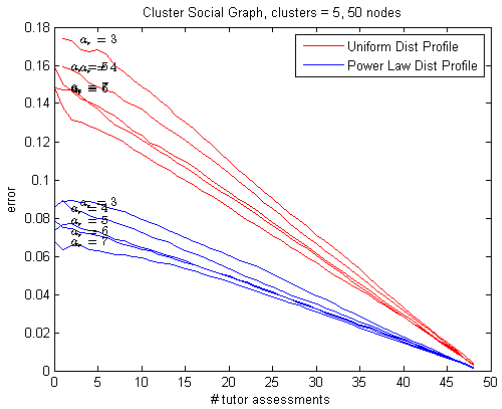
3.4 Evaluation

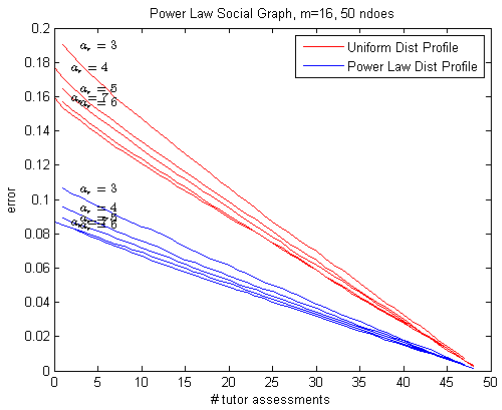
In experiment 1, we observe (Figure 5) that the error decreases when the number of tutor assessments increase, as expected, until it reaches 0 when the tutor has assessed all 50 students. This decrement is quite stable and we do not observe abrupt error variations or important error increments from one iteration to the next. More variations are observed in the initial iterations since the service has only a few assessments to deduce the weights of the trust graph and to calculate the final outcome.

In the case of experiment 2 (Figure 6), the error diminishes slowly as the number of student assessments increase, although it never reaches 0. Since the number of tutor assessments is fixed in this experiment, we have an error threshold (a lower bound) which is linked to the students' assessment profile: the closest to the tutor's the lower this threshold will be. In fact, in both experiments we observe that when using a power law distribution profile (Figure 4(a)) the automated assessment error is lower than when using a uniform distribution profile (Figure 4(b)). This is because when using a power law distribution, more student profiles are generated whose assessments are closer to the tutors'.

In general, the error trend observed in all experiments comparing different social network scenarios (random, cluster or power law) show a similar behavior. Taking a closer look at experiment 2, cluster social graphs have the lowest error and we observe that assessments on all assignments are achieved earlier (this is, the trust graph gets connected earlier). We attribute this to the topology of the fully connected clusters which favors the generations of indirect edges earlier in the graph between the tutor and the nodes of each cluster. Power law social graphs have lower error than random networks in most cases. This can be attributed to the criteria of preferential attachment in their network generation,







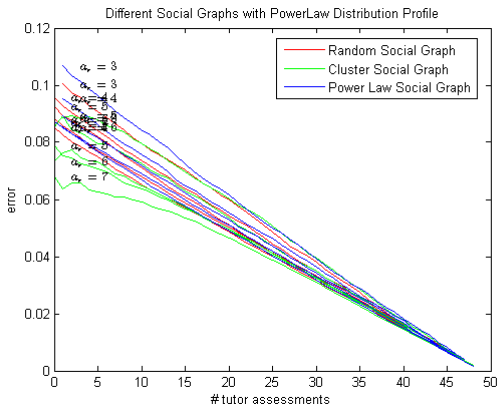
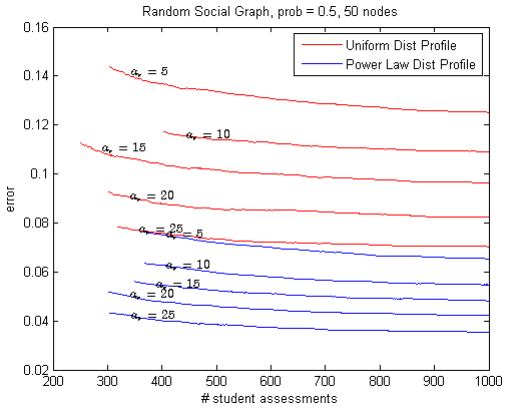
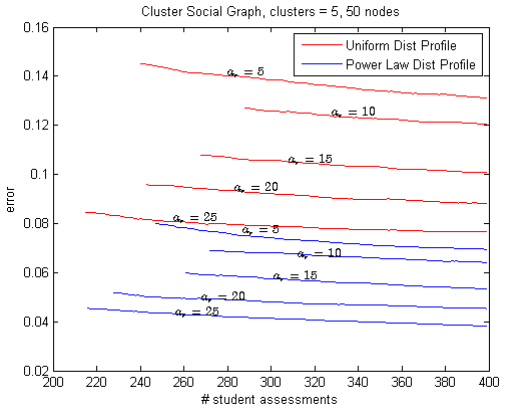
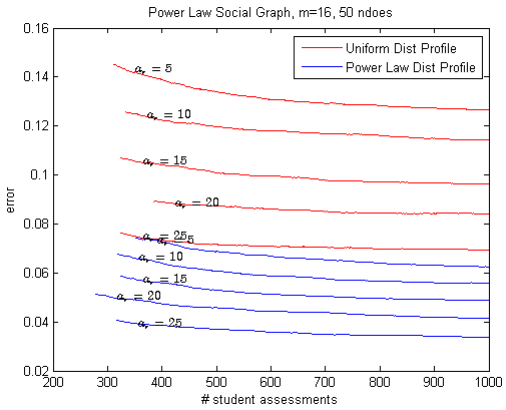


Figure 5: Eperiment 1







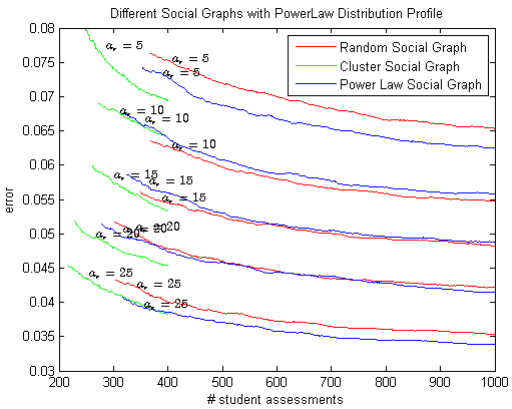


Figure 6: Experiment 2

which favors the creation of some highly connected nodes. Such nodes are likely to be assessed more frequently since more peers are connected to them. Then, the automated assessments of these higly connected peers are performed with more available information which could lead to more accurate outcomes.

4. DISCUSSION

The collaborative assessment model proposed in this paper is thought of as a support in the creation of intelligent online learning applications that encourage student interactions within communities of learners. It goes beyond current tutor-student online learning tools by making students participate in the learning process of the whole group, providing mutual assessment and making the overall learning process much more collaborative.

The use of AI techniques is key for the future of online learning communities. The application presented in this paper is specially useful in the context of MOOC: with a low number of tutor assessments and encouraging students to interact and provide assessments among each other, direct and indirect trust measures can be calculated among peers and automated assessments can be generated.

Several error indicators can be designed and displayed to the tutor managing the course, which we leave for future work. For example the error indicators may inform the tutor which assignments have not received any assessments yet, or which deduced marks are considered unreliable. For example, a deduced mark on a given assignment may be considered unreliable if all the peer assessments that have been provided for that assignment are considered not to be trusted by the tutor as they fall below a preselected acceptable trust threshold. Alternatively, a reliability measure may also be assigned to the computed trust measure T_D. For instance, if there is only one assignment that has been assessed by τ and ρ, then the computed $T_D(\tau, \rho)$ will not be as reliable as having a number of assignments assessed by τ and ρ. As such, some reliability threshold may be used that defines what is the minimum number of assignments that both τ and ρ need to assess for $T_D(\tau, \rho)$ to be considered reliable. Observing such error indicators, the tutor can decide to assess more assignments and as a result the error may improve or the set of deduced assessments may increase. Finally, if the error reaches a level of acceptance, the tutor can decide to endorse and publish the marks generated by the collaborative assessment method.

Another interesting question for future work is presented next. Missing connections might be detected in the trust graph that would improve its connectivity or maximize the number of direct edges. The question that follows then is, what assignments should be suggested to which peers such that the trust graph and the overall assessment outcome would improve?

Additionally, future work may also study different approaches for calculating the indirect trust value between two referees. In this paper, we use the product operator. We suggest to study a number of operators, and run an experiment to test which is most suitable. To do such a test, we may calculate the indirect trust values for edges that do have a direct trust measure, and then see which approach for calculating indirect trust gets closest to the direct trust measures.

Acknowledgements

This work is supported by the Agreement Technologies project (CONSOLIDER CSD2007-0022, INGENIO 2010) and the PRAISE project (EU FP7 grant number 388770).

5. REFERENCES

[1] Praise project: http://www.iiia.csic.es/praise/.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 1999.

[3] L. de Alfaro and M. Shavlovsky. Thecnical report 1308.5273, arxiv.org. *Crowdgrader: Crowdsourcing the evaluation of homework assignments*, 2013.

[4] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 1959.

[5] E. Ferrara and G. Fiumara. Topological features of online social networks. *Communications in Applied and Industrial Mathematics*, 2011.

[6] L. Godo and R. Rodríguez. Logical approaches to fuzzy similarity-based reasoning: an overview. *Preferences and Similarities*, 2008.

[7] C. Piech, J. Huang, Z. Chen, C. Do, A. Ng, and D. Koller. Tuned models of peer assessment in moocs. *Proc. of the 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.

Mining for Evidence of Collaborative Learning in Question & Answering Systems

Johan Loeckx
Artificial Intelligence Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussel
jloeckx@ai.vub.ac.be

ABSTRACT

Question and Answering systems and *crowd learning* are becoming an increasingly popular way of organising and exchanging expert knowledge in specific domains. Since they are expected to have a significant impact on online education [14], we will investigate to which degree the necessary conditions for collaborative learning emerge in open Q&A platforms like Stack Exchange, in which communities grow organically and learning is not guided by a central authority or curriculum, unlike MOOCs. Starting from a pedagogical perspective, this paper mines for circumstantial evidence to support or contradict the pedagogical criteria for collaborative learning. It is observed that although there are *technically no hindrances towards true collaborative learning*, the nature and dynamics of the communities are not favourable for collaborative learning.

The findings in this paper illustrate how the collaborative nature of feedback can be measured in online platforms, and how users can be identified that need to be encouraged to participate in collaborative activities. In this context, remarks and suggestions are formulated to pave the way for a more collaborative and pedagogically sound platform of knowledge sharing.

1. INTRODUCTION

Computer-assisted instruction (CAI) is one of the hottest topics in education research [9] and often claimed to revolutionise how we teach and learn [6]. Massive Open Online Courses or MOOCs are the newest manifestation of this phenomenon. However, while 2012 was being praised as "the year of the MOOC", more and more critical voices were heard during the last year and MOOCs are under increasing pressure to finally live up to their promise. Spoken in terms of of *Gartner's Hype Cycle* [8], we could say that we're either at the peak of inflated expectations, or already entering the *through of disillusionment* [3, 15, 10].

This however does not mean that online learning isn't advancing in many interesting directions: Kahn's academy emerged more or less organically when Salman Kahn started teaching his cousin mathematics using short videos. When Salman realized a lot more children could benefit from these lessons, he started distributing them on YouTube. Today, Kahn Academy reaches 10 million students per month, according to Wikipedia. Wikipedia itself has become an integral part of traditional education too. Some researchers expect that learning in general will evolve from an individual task centred around the teacher-student dichotomy, to a collaborative social activity, in which online knowledge bases like Wikipedia, forums, social networks and Question & Answering systems are playing an ever more important role [4]. In this paper, we will try to find evidence of the claimed collaborative properties of Q&A systems, more in particular the music forum site of Stack Exchange[1]. Though the analysis is based on text-based feedback, it is expected that the dynamics of feedback in collaborative activities also hold in multi-modal situations.

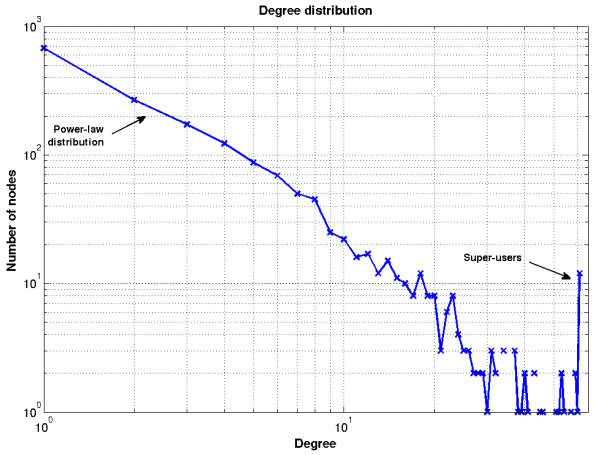


Figure 1: The degree distribution shows that the network of user-interaction is scale-free, which supports the hypothesis that there is no symmetry of knowledge.

This paper is structured as follows. First, the pedagogical background of collaborative learning is set out, based upon the work of Dillenbourg [7] and conditions for and indicators of collaborative learners are introduced. Next,

[1]http://music.stackexchange.com

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

educational data mining techniques are applied [12] to find evidence of collaborative learning in *crowd learning* systems, more specifically Question and Answering systems like Stack Exchange. Lastly, a critical discussion is performed and suggestions towards more collaborative Q&A systems are proposed, to end with conclusions.

2. COLLABORATIVE LEARNING

2.1 Pedagogical approach

Existing definitions of collaborative learning in the academic fields of psychology, education and computer science, differ significantly and are often vague or subject to interpretation. We thus needed a theory that unified the different theories and was applicable to the online, computerised world as well. Not the least, it had to be easily operationalisable. A review of the literature brought us to the work done by Pierre Dillenbourg [7] that perfectly suited our requirements. Dillenbourg takes a broad view on the subject and argues that collaborative learning is a *situation* in which two or more people *learn* through *interactions*.

This means that *collaborative learning can not be reduced to one single mechanism*: just like people do not learn because they are individual but rather because the activities they perform trigger learning mechanisms, people don't learn collaboratively because they are together. Rather, the interactions between the peers create activities (explanation, mutual regulation,...) that trigger cognitive learning mechanisms (elicitation, internalisation, ...) [7].

For these processes to be effective, some requirements need to be fulfilled. A subset was extracted that could be measured numerically, albeit indirectly, using the information available in our data set (summarized in Table 1). In the next section we will have a closer look at these indicators.

2.2 Indicators

Dillenbourg discriminates three important aspects for collaborative learning to be effective and characterises situations, interactions and processes as *collaborative* if they fulfil the following criteria:

- Peers are more or less at the *same level*, have a *common goal* and *work together*;
- Peers *communicate interactively*, in a synchronous and *negotiable manner*;
- Peers apply mechanisms like *internalisation, appropriation and mutual modelling.*

These high-level criteria have been refined by Dillenbourg into more detailed conditions for collaborative learning, of which a subset has been summarised in Table 1. Each corresponding indicator provides indirect circumstantial evidence for each criterion, as our analysis was limited by the data available in the Stack Exchange. Nevertheless, as we will see, they give useful insight in the formation and dynamics of open online collaborative communities for learning.

The research in this paper can be seen as an extension of previous research in Educational Data Mining, that measured participation and interaction between students [11] and the successful formation of learner's communities [1, 13].

3. QUANTITATIVE ANALYSIS

Stack Exchange can be considered as a distant-learning autodidact platform in which communities are formed organically and learning is not guided by a curriculum or some central authority, but exclusively by the members of the community, in contrast with MOOCs. This paper aims at answering the question whether the necessary conditions for collaborative learning emerge spontaneously in these platforms. As the work is done in the context of the PRAISE project[2], a social media platform for music learning, the Music Stack Exchange data set was chosen.

Stack Exchange provides an open API, from which all data can be exported. The data set consisted of 2400 questions, 1500 active members and 1.7 million page views The platform is basically a forum in which anyone can ask and reply to questions. As a means of quality control, users can give up- and down votes to questions, and answers. People can also comment on questions and answers which is actually some kind of meta-discussion in which feedback on relevance, terminology, etc... is given. In the following paragraphs, the criteria listed in Table 1 will be studied in more detail.

3.1 Symmetry of action

Symmetry of action expresses the extent to which the same range of actions is allowed by the different users. Stack Exchange employs a system of so-called *privileges*, attributed according to your reputation[3]. These privileges are generally connected to *moderation rights*, rather than with the actions of asking and replying to questions – unless you have a negative reputation. The fact that users can exert the same actions, does not imply that this also actually the case. An analysis of the distribution of the ratio of answers over the number of questions, reveals that we can roughly discriminate *three kinds of users*, based upon their activity profile:

- *Silent users (62% of the registered users)* that never answer, e.g. users that don't register or register but do not ask questions nor reply to them;
- *Regular users (37% of registered users)* that give roughly as much as answers as they ask questions, that is, two on average;
- *Super-users (<1% of the registered users)*, these are 'hubs' that give at least 40x more answers than they ask questions.

The largest part (96%) of *regular users*, ask less than five questions, and 76% even asks only one question: there are *no 'parasite' users between the regular users that ask question but do not answer.* From the other side, only 8 'expert' super-users (0.5% of the community) were responsible for answering 25% of the questions. Above findings indicate that **the symmetry in action is highly skewed because of a small group of 'super-users' and a large group of 'silent users'.**

[2]http://www.iiia.csic.es/praise/
[3]http://stackoverflow.com/help/privileges

Aspect	Criterion	Indicator
Situation	Symmetry of action	Ratio of answers and questions per user
	Symmetry of knowledge	Scale-freeness of the user interaction graph
	Symmetry of status	Distribution of reputation within the community
Interactions	Synchronous	Response times of answering to questions
	Division of labour	Distribution of questions and answers in the community

Table 1: Criteria of collaborative learning according to Dillenbourg, with corresponding indicators. The indirect nature of the indicators stems from the fact that only meta data was available from the Stack Exchange data set, and that the criteria in general are very hard to measure quantitatively.

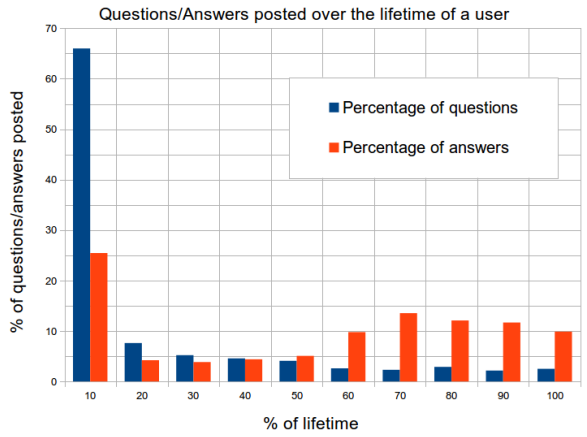


Figure 2: Users tend to ask more questions in the beginning when signing up, and start answering as they have been around some time.

3.2 Symmetry of status

Stack Exchange employs a *reputation system* by which members get rewarded or punished if a peer up- or down votes your answer or question, when your answer gets 'accepted', etc...

We would expect a "healthy" collaborative community to have a strong correlation between reputation and the time a user has been around on the platform: as users spend more time on the platform, their reputation builds up. An inquiry into the Stack Exchange music data set, however, reveals only a correlation of 0.23 between reputation and "time around". We could thus conclude that there is some **odd kind of symmetry, in the sense that no one really builds up reputation**.

3.3 Symmetry of knowledge

Traditionally, these reputation systems are believed to make a good indicator for the knowledge a user possesses. However, there are some problems with this reasoning:

- Knowledge is not a uni-dimensional measure, but is connected to a (sub) domain of expertise;
- Someone's reputation keeps on increasing, even without activity: there is a bias towards old posts and members;
- There is a bias towards "easy answerable questions".

Figuring the knowledge of the members directly is quite an impossible task to perform, especially in a broad and open-ended domain like music. To assess symmetry of knowledge, however, one could argue that *if* everyone in the Stack Exchange music learner's community has more or less the same expertise, *then*, on average, anyone would answer questions asked by anyone.

In other words, there would be no particular hierarchy in answering, rather the network of interaction would be "random" and *not scale-free*. Another way to put this, is to state that *no hubs of people would exist that answer significantly more questions than others*. A network is called *scale-free* if the degree distribution follows a power law[2]:

$$P(k) \sim k^{-\gamma} \tag{1}$$

with $P(k)$ being the fraction of nodes that have a degree k, and γ a constant typically between 2 and 3. Figure 1 reveals a power-law relationship, with exception this special group of "super-users". Above findings therefore suggest that **symmetry of knowledge is not observed.**

3.4 Division of labour

As pointed out before, a small group of super users answer vastly more questions than they ask: a group of 21 users answered half the questions. This is clearly not a balanced situation in which the total labour of answering questions, is equally distributed. Figure 2 shows the relative timing of when users ask and respond to questions over their lifetime.

Users tend to ask questions in the beginning (a visit to the site probably triggered by an urgent need to get a question resolved), but start answering more uniformly after a while. The graph also indicates that engagement is largest in the beginning. This information is relevant when developing platforms with a pedagogical purposes: **users probably need to be "bootstrapped", allowing them to give lesser answers and ask more questions in the beginning, so they get "locked into" the platform**.

Note that a relative plot was preferred, in which the x-axis indicates the % of the lifetime, 0% being the moment of signing up, and 100% the date the data set was obtained. It allowed us to grasp the details of both users that had just signed up, as well as users that have been active for a long time (especially as the rate of signing up is probably not constant but increases with time).

3.5 Synchronous feedback

To keep people engaged in an activity, according to the "theory of flow" [5], immediate feedback is necessary. In the case

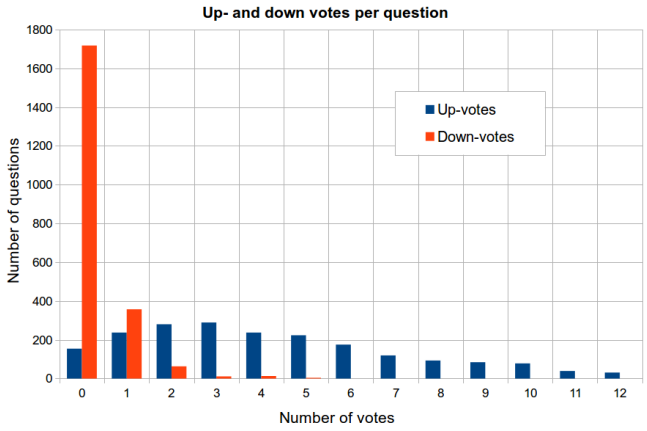


Figure 3: Users tend to give much more up-votes than down-votes to questions. Generally speaking, down-voting is only used to remove off-topic, duplicate questions or questions that are either too specific or broad.

of the music Stack Exchange platform, 68% of the questions received an answer within the day, and 20% even within the hour. This may seem odd, but closer inspection reveals that – once again – this is due to the small-group of "super-users" that are very engaged.

4. CRITICAL DISCUSSION

Based upon the analysis done in the previous section, some critical remarks and suggestions are offered to improve the pedagogical nature and collaborative learning

4.1 Remarks

4.1.1 Limited to no instructional design

The data set on Stack Exchange music's forum, is an amalgam of questions (1) with different levels of granularity, typically with a small scope, (2) on a wide range of topics, for learners (3) with different learning goals and (4) different levels of expertise. The activities are not designed to elicit collaborative learning, and as the data is unstructured, without sufficient scaffolding of the learning content (e.g. through hyper-linking), it is no natural fit for learning but rather provides **ad-hoc answers to appease short-term narrow personal learning goals**.

4.1.2 A heterogeneous community

Above remarks wouldn't be so *problematic for collaborative learning*, if proficient communities existed within the Stack Exchange platform that had more or less the same goals, expertise and engagement. In the current case, there's a risk of frustration and boredom in expert users that don't see their questions answered and who have to answer straightforward questions. For novice members, on the other hand, their learning remains limited because they do not get sufficient guidance and do not really construct knowledge.

Although *the group of super-users* makes sure that questions get answered quickly and perform the largest part of moderation, they are *potentially harmful to collaborative learning* as they distort the natural formation and dynamics of collaborative communities. From the other side, their interventions may bootstrap "young" forums.

4.1.3 Strong preference for "liking"

The dataset revealed a *very strong preference for voting up rather than down*: only two users gave more down votes than up votes and of all the people that have ever cast a down vote (72 users out of the roughly 1500 active users), 80% gave more than five times as much up-votes in return. 80% of the questions had *no down vote*, compared to less than 10% without up-vote. Figure 3 shows the distribution of up- and down-votes. This effect was even more pronounced in the answers: the number of down-votes is typically zero or very small, whereas the up-votes reach a maximum at about 3 up-votes, then slowly attenuates. A further analysis of questions with more down than up-votes, revealed that these questions where either off-topic (40%), too vague, broad or specific (35%), not real questions (10%) or Duplicate questions (8%).

4.2 Suggestions

4.2.1 Sub-communities

Allowing users to organise themselves in smaller active sub-communities with common or similar learning goals, may prove an elegant solution to manage or exploit the variety in expertise of the users. Also, the concept of reputation would make more sense. A similar idea was proposed by Santos [13].

4.2.2 Knowledge construction

Good feedback should provoke critical thinking by asking sensible questions, provide a clue to "what's next" and allow to construct knowledge through scaffolding and coupling back to acquired knowledge. Though the concept of freely asking questions is very accessible, the content stays rather ad-hoc and unstructured. A way to organise and link different questions in order to guide learners would be very useful.

4.2.3 Collaborative interfaces

In the modern ages of web technology, users could benefit from a collaborative interface in which knowledge is constructed together, in a way similar to for example Google Docs where one single entity is shared by all users. So, rather than preserving the strict question/answer or learner/teacher dichotomy, one would go for a situation in which knowledge – not only answers but also questions – is constructed live in an interactive way.

5. CONCLUSIONS

In this paper, the case for collaborative learning in open-ended auto-didact Q&A environments like Stack Exchange is investigated. Based upon the criteria put forward by Dillenbourg, we can state that though there are *technically no hindrances towards collaborative learning, the nature and dynamics of the community that organically form on Stack Exchange, do not support the case for collaborative learning.*

It was observed that the *symmetry of action was distorted* due to a small group of "super-users" that answered the majority of questions and a large group of "silent users" that do not really interact with the platform. Inspection of the

degree distribution of the user interactions reveals that the community network is scale-free, which means that *symmetry of knowledge is very unlikely*. The reputation system seems insufficient as a measure of expertise and a strange kind of symmetry of status is observed, in the sense that *no one really builds up reputation*, except for a small group of users.

Lastly, the limited possibilities to instructional design, elicits *short-term narrow and personal learning goals*. Also, the very heterogeneous nature of the community is not favourable for learning. Suggestions were made to adapt these interesting and popular platforms to learning, like *creating subcommunities with common learning goals*, extend the possibilities for *organising and structuring the content* and employ *collaborative interfaces*.

As future work, these results should be validated by means of other communities on Stack Exchange as well, and on different modes of feedback, rather than only text-based.

6. ACKNOWLEDGEMENTS

This research has been supported by the EU FP7 PRAISE project #318770.

7. REFERENCES

[1] A. R. Anaya and J. G. Boticario. A data mining approach to reveal representative collaboration indicators in open collaboration frameworks. *International Working Group on Educational Data Mining*, 2009.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] C. G. Brinton, C. Mung, S. Jain, H. Lam, Z. Liu, and F. Ming Fai Wong. Learning about social learning in MOOCs: From statistical analysis to generative model. *arxiv.org*, abs/1312.2159, 2013.

[4] M. A. Chatti, M. Jarke, and D. Frosch-Wilke. The future of e-learning: a shift to knowledge networking and social software. *International journal of knowledge and learning*, 3(4):404–420, 2007.

[5] M. Csikszentmihalyi. *The Evolving Self: A Psychology for the Third Millennium*. Harper Collins, New York, 1993.

[6] L. Cuban and L. Cuban. *Oversold and underused: Computers in the classroom*. Harvard University Press, 2009.

[7] P. Dillenbourg et al. Collaborative-learning: Cognitive and computational approaches. Technical report, Elsevier, 1999.

[8] J. Fenn and M. Raskino. *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time*. Harvard Business Press, 2008.

[9] J. Hattie. *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. Routledge, 2009.

[10] T. Lewin. After setbacks, online courses are rethought. *New York Times*, (December 11), 2013.

[11] T. P. Padilha, L. M. Almeida, and J. B. Alves. Mining techniques for models of collaborative learning. In *Designing Computational Models of Collaborative Learning Interaction, workshop at. ITS*, pages 89–94, 2004.

[12] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, 2007.

[13] O. C. Santos, A. Rodríguez, E. Gaudioso, and J. G. Boticario. Helping the tutor to manage a collaborative task in a web-based learning environment. In *AIED2003 Supplementary Proceedings*, volume 4, pages 153–162, 2003.

[14] M. Sharples, P. McAndrew, M. Weller, R. Ferguson, E. FitzGerald, T. Hirst, and M. Gaved. Open university: Innovating pedagogy. 2013.

[15] V. Strauss. Are MOOCs already over? *http://www.washingtonpost.com/blogs/answer-sheet/wp/2013/12/12/are-moocs-already-over/*, (December 12), 2013.

Creative Feedback: a manifesto for social learning

Mark d'Inverno
Department of Computing
Goldsmiths, University of London
+44 207 919 7701
dinverno@gold.ac.uk

Arthur Still
Department of Computing
Goldsmiths, University of London
+44 207 919 7701
awstill@btinternet.com

ABSTRACT

Arguably one of the most important activities of a university is to provide environments where students develop the wide variety of social and intellectual skills necessary for giving and receiving feedback. We are not talking here about the kinds of activity typically associated with the term "feedback" - such as that which occurs through individual course evaluation questionnaires or more universal systems such as the National Student Survey, but the profoundly creative and human act of giving and receiving feedback in order validate, challenge and inspire. So as to emphasise we are talking about this kind of feedback, we coin the term "creative feedback" to distinguish it from the pre-conceived rather dreary compliance-inflected notions of feedback and set out in this paper to characterise its qualities. In order to ground and motivate our definition and use of "creative feedback" we take a historical look at the two concepts of creativity/creative and feedback. Our intention is to use this rich history to motivate both the choice two words, and the reason to bring them together. In doing so we wish to emphasise the characteristics of an educational philosophy underpinned by social interaction. By describing those qualities necessary to characterise creative feedback this paper sets out an educational philosophy for how schools, communities and universities could develop their learning environments. What we present here serves not only as a manifesto for designing learning environments generally but as a driver for designing technologies to support online social learning. Technology not only provides us with new opportunities to support such learning but also to investigate and evidence the way in which we learn and the most effective learning environments.

Keywords: Feedback, creative, creativity, learning, technology

1. INTRODUCTION

When the word feedback is mentioned in universities - as happens now with increasing frequency - there are usually one or two winces around the room. The problem it is a word that has become associated with compliance, with checking competency, with measurement and judgement, with having to go through the motions of various government or funding body processes and, perhaps too, with feeling beholden to open up channels of communication so as to hear things that we would rather not have to hear. This is a pity, and especially so at universities, because feedback is central to learning. Not just to learn a discipline, but to learn about the way we are, to learn about the way we think, to learn about the way we interact and about the way in which we produce and value our work. Whether that work is an analytical or interpretive essay, whether it is a poem or a composition, whether it is a new performance or a new artwork, it is only through actively seeking feedback both from others and from ourselves that we learn.

At one level it is clear that without the on-going feedback that we sense and perceive from our environment we could not operate or survive. Without basic perceptual acts such as seeing, hearing and touching we couldn't function for very long. However, feedback is also necessary to experience ourselves as social beings, and especially to understand and investigate the process of social interaction between individuals. Sometimes the communication from one human to another is like an experiment whose result is evidenced by the feedback perceived from the other [22]. For example, shouting "hello?" to check whether anyone is at home, the result might be the perception of a response like "I'm in the kitchen!" or complete silence. This is an example of a simple feedback loop at work providing evidence for a model of the world. At the other extreme feedback loops can be continuous and extremely complex, and often below conscious awareness such as when two jazz musicians are improvising together [54]. In all cases feedback is the way in which we understand the world we are in, and learn about our physical and social place within it.

Suppose you are a learning to play music, for example. If you play a piece of music then the only way you can know how it was heard and experienced by others is to get their feedback on your performance. This feedback will be absolutely critical if you want to understand how you can improve yourself as a performer. Of course in any performance sustained self-feedback is critical too and musicians are skilled enough to give themselves this on-going and continuous feedback as they play. In addition to this, musicians have the option of recording performances and listening to them later in order to provide an entirely new perspective. The distance created in time and space, and moving from performer to listener, provides new opportunities for fresh insights on how to improve ones own performance. In addition, through an understanding of how we come across to others, we can often best advance the quality and precision of the feedback we give ourselves.

If we accept the need for building communities of feedback the issue then becomes how to build the right kinds of learning environments. If students can develop their own skills in giving and receiving feedback at school and university, then they will gain confidence in giving and receiving feedback from friends, colleagues, press and audiences too. Education environments should enable an exploration of how peers and tutors perceive essays, performances, software and artworks and in turn, how we all learn to be open to the feedback from others.

This philosophy is very strong in the Art department at Goldsmiths, where the emphasis is very much focused on developing communities of feedback. This department is especially interesting because of its reputation for producing world-class artists that have become important cultural and creative pioneers in the UK.[1] In our observations, first, second and third year undergraduates come

[1] (Damien Hirst, Malcolm McClaren, Mary Quant, Lucien Freud and Anthony Gormley are all alumni of the Art department. Other alumni include Laurie Provoust who currently holds the Turner prize and Steve McQueen who won a Bafta and Oscar for best film with "12 years a slave". The question to us is whether developing communities of creative feedback is the key to the Art department's success.)

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

together weekly in order to give feedback on a small selection of undergraduates work. The students clearly worked as a group in balancing praise and criticism, combining the emotional and analytical, and moving from the sociological to the political. In all these open conversations students are learning about how to give and receive feedback to each other and understanding the ever present gap between any intention behind an artwork, and the perception by others. One of the most fascinating aspects observed in these sessions was the ability of students to take a sufficient emotional distance in order to be open to feedback, and to experience it freely without personalising anything. This ability is not only key in terms of learning how others experience their work but becomes an important skill for artists moving into a professional sphere with the free-for-all comment and criticism that social media now encourages.

Arguably then, a learning institution's key objective is to provide the kind of supportive and trusting environments where students can develop their ability to give and receive feedback in a culturally-aware, sensitive, mindful, critical and challenging way. We certainly think so, and would like a label to describe the kind of feedback we have in mind, and for this we choose the term "creative feedback". In this paper we provide a historical account of the notions of creative and creativity in order to justify the use of this term in an educational context. Moreover, by using this term explicitly the hope is we can rescue the concept of feedback from its often rather dreary compliance-inflected interpretation.

In what follows we will call upon our experience as educators spanning mathematics, psychology, psychotherapy, music and computer science, to try to explain what we mean by creative feedback and to justify our use of this term. To do this we need to take a brief historical look at the concepts of "creative" (and the related "creativity") and "feedback" – particularly though not exclusively in an education context - in order to explain exactly what we mean by these terms and why we are bringing them together specifically. The aim of the historical analysis is to give currency to the use of the term and the underlying manifesto for learning. We clearly need to be mindful of using the word "creative" when it is used so loosely, and for so many different educational, marketing and political reasons. We not only have creative writing and creative learning but now we have creative musicianship, creative computing and creative financing, not to mention the growing importance given to "creative industries" and economic arguments about why they are such an important part of our future. The word is in danger of being no more than what is approved of, and we wish to recover an older and fuller meaning for our purposes.

Aims. In this paper we set out to characterise creative feedback as the basis of an educational philosophy that is inspired by the American psychologist, philosopher, and educationalist John Dewey. The idea that follows naturally from this is that we structure schools, learning groups and universities as "communities of discovery". There are a number of motivating factors for the work in this paper described next.

The first is the desire to build educational environments (which include online environments) that give more people access to developing "creative feedback" skills. Creative feedback belongs to what Dewey called "creative intelligence" which is a part of all human thinking and is available to everyone. A strong part of our individual learning journey is gaining an understanding how others see us. The way we think, the way we behave, what we produce. This understanding is such a crucial part of learning that we want to build environments that encourage students to be aware of how others see them. As George Herbert Mead wrote, *"the individual mind can exist only in relation to other minds with shared meanings"* [42: p5]. If this is true, the relation to other people is grounded within a framework of feedback and the individual mind can only exist within such a framework.

Next, we want to emphasise that "creativity" depends on feedback from the world rather than being something that is an intrinsic quality that resides within individuals. It depends on feedback both in the act of creation itself, and also the social feedback that is received once it is made available to others (which may or may not amount to acclamation as great art).

As stated above feedback is not often seen as a creative endeavour but rather as being quite mechanical (tick boxes and scores) and about compliance (such as is often the case when making module feedback forms available to students). The impact of this notion of feedback on tutor/tutee relationships can often be dire. We explicitly introduce "creative feedback" to mitigate against this commonly held view of feedback and, in addition, to move away from another commonly held conception about feedback that it only exists in terms of praise and punishment. Furthermore, we want to emphasise how we are immersed in feedback as biological and social beings and we wish any definition to encompass this.

Most educationalists like us want to promote effective education as available to everyone rather than a middle-class luxury and technology clearly has an important role here. However, technology also provides opportunity to bring communities of learners together and, moreover, serve as a test-bed from which we can start to evidence the benefits of social learning over the individual, rote-learning and exam-based methodology that so dominates current political thinking. It also provides us with exciting new possibilities for understanding the way in which we learn. One of the drivers in our own research, for example, is to develop learning analytics and methodologies that can enable us to correlate creative feedback with learning.

The ability to use technology to understand and support social learning depends on whether we can construct systems that encourage humans to give and receive creative feedback. In order to achieve this we need participatory design methods working with a variety of user groups in order to design software that can support creative feedback across a whole range of disciplines (e.g. poetry, music, design, digital art). We believe a historical and educational underpinning is necessary to drive the principled design of such systems that not only support creative feedback but also allow mixed human and computational societies. One of the practical questions that we are addressing in the design of novel education systems that enable social learning is how to build autonomous artificial systems that can help exemplify creative feedback in a learning community.

2. A HISTORY OF CREATIVITY AND FEEDBACK

The Education Wars. Ever since people started arguing about education, there has been an angry debate that is still not resolved, and is especially marked today in England. On the one hand the Secretary of State for Education crusades for even more frequent and stringent examinations and inspections in the State-based schools, creating what his critics call "*exam factories*" [12], designed to compete with the dauntingly efficient exam factories of the Far East.[2] And on the other hand the popular educationalist

[2] "*Tougher GCSE marks pegged to China scores*". Guardian headline, 3.4.14

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

Sir Ken Robinson speaks for many when he condemns such an approach for undermining creativity, which is the true goal of democratic education. It may be hard to define creativity, but everyone agrees that it is a good thing, and that it is not fostered by an exclusive focus on training students for success in exams. The emphasis on exam factories may even be self-defeating, since there are studies showing that the success of children in China and Japan depends more on the early nurturance of sociality, than on forced study and rigorous examinations [35] More like what Coffield called "*communities of discovery*" than "*exam factories*", so perhaps Gove is taking us "*ever faster down the wrong road*" [11].

Background to the Conflict. This quarrel occurs at every level of education, from toddlers to adults, and it reflects different views on the nature of children. At one extreme is the active child, full of wonder and curiosity at the world, who needs only skilled guidance from the teacher to flower into a civilized and creative adult. At the other is the resistant child, lazy and easily distracted, whose motivation and attentiveness require firm moulding and sometimes medication in order to learn lessons and become a good citizen. Around 1900 these extremes were given psychological and educational form by two prominent American thinkers [61], and this set the scene for many of the debates on education during the coming century. In the active, curious child camp sat the philosopher, educationalist and psychologist, John Dewey, the great champion of American pragmatism, which is a philosophy based on doing rather than thinking; in the other camp sat Edward Thorndike, famous throughout the 20th century for his puzzle box experiments with cats published in 1898 [56] in which he claimed to show that cats are incapable of reason and learn only through trial and error. During the second half of the 20th century both camps contributed to the new interest in creativity, which has now become a massive and well-funded research industry in Europe especially in relation to technology.

In this paper we aim to show how technology can contribute to the fostering of creativity in education in a way that can satisfy both the jeremiads of Professor Robinson and the ministerial anxieties of Michael Gove. But first we need to be clear about what kind of learner we have in mind, Dewey's or Thorndike's, since this determines what we mean by creative and creativity, and the deployment of these terms has provided a map of the hidden agendas of Psychology and Educational Theory during the 20th century.

E. L. Thorndike: Connectionism, Stimulus-Response And The Importance Of Measurement. In 1911 Thorndike published his puzzle box experiments in *Animal Intelligence*, and developed the theory that learning is initially guided by random trial and error learning, rather than rational intelligence. For Thorndike and later many Behaviourists, the unit of behaviour was the stimulus response (S-R) connection, treated as a kind of reflex. Thorndike's view was that learning takes place by establishing connections in the brain and these connections are stamped in through a system of reward and punishment. Applied to education it was argued that the randomness of the trials in initial learning showed that little is to be gained by relying on the prior capacities of the novice learner.

Connections were treated as "atoms of the mind", and Thorndike speculated that "*the vague gross feelings of the animal sort might turn into the well-defined particular ideas of the human sort, by the aid of a multitude of delicate associations*" [58: p289]. This is Thorndike's Connectionism, and it has been one of the main models guiding studies of learning throughout the 20th century, though it was quickly found that the S-R scheme needed to be extended to S-O-R [68]. In this extended scheme O refers to the state of the organism, which is made up of many variables or factors, including prior knowledge (the multitude of delicate associations), motivation, attentiveness, intelligence and many other variables.

During the second half of the 20th century computers became the new model of the mind, and the language for describing "a multitude of delicate associations" became increasingly sophisticated, eventually leading to a new brand of Connectionism as a model for perception and learning [3]. But even in its most sophisticated form, it is still about the selection of successful acts and the "stamping out" of "profitless" [58: p283] acts by reward and punishment. Nowadays we speak of input and output of information rather than S-R, but whatever the cognitive complexity of what goes on in between, a basic linear structure remains, with the environment operating on the organism, rather than the organism on the environment.

But Thorndike was not only one of the founders of S-R theory, he was also a pioneer of mental testing as a way of classifying individuals for social control, and therefore for assigning numbers to the "O" variables in the S-O-R scheme. Thorndike greatly admired the work of Darwin's cousin Francis Galton (1822-1911) who spent much of his life studying and measuring human variation and its genetic basis after reading *Origin of Species*. As part of this interest Galton became the first to use questionnaires and statistics for the measurement of human differences and Thorndike in turn became a champion of measurement in Psychology and Education. In 1904 he published *An Introduction to the Theory of Mental and Social Measurements* [57] which introduced students to the new statistical methods that were to dominate the scientific practice of Psychology

Deweyan Inquiry. The contrasting philosophy was that of John Dewey, who was one of the first to acknowledge the value of Galton's statistical discoveries [16] but had little faith in the value of measuring the worth of individual human beings [36]. He believed effective education is powered by the child's spontaneous curiosity about the world and is social, taking place in "a community held together by participation in common activities" [20: 55]. This social setting generates inquiry, a process as natural as breathing in all animals. Inquiry is an ongoing process that reveals novelty, which in turn becomes the spur to further inquiry.

In 1896 Dewey had made the revolutionary step of taking the basic S-R reflex studied in the laboratory by physiologists, not as the simple arc of Thorndike, but as a circular structure with neither stimulus nor response being dominant over the other. He argued that the S-R reflex is not an isolable molecule of behaviour, but is inseparable from an ongoing process involving what 50 years later would be called feedback.[3] Dewey was not a laboratory psychologist, and unlike Thorndike's S-R, his scheme did not lend itself to precise control, since it required freedom of action for optimal learning to take place.

The main concern for the teacher therefore is to guide this action toward educational goals, and to avoid stifling freedom through the indiscriminate "stamping out" of what Thorndike referred to as "profitless" acts. For Dewey these "profitless" acts are part of what

[3] Thorndike's S-R connectionism also involved a rudimentary form of feedback. Reward and punishment applied to isolated S-R connections are feedback. But Dewey seemed to have in mind what we now think of as a self-organising system, in which the parts, which we may for convenience label stimulus, response, feedback, etc., cannot usefully be isolated and studied as "laboratory preparations" outside the system. The knowledge gained by an inquiring child involves, not a changing array of S-R connections, but an evolving place within a system that includes its social and physical environment.

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

he called inquiry and to stamp them out is to suppress inquiry and to stunt human development.

Who Has Won? In Psychology and in Education, Thorndike has won hands down:

One cannot understand the history of education in the United States during the twentieth century unless one realises that Edward L. Thorndike won and John Dewey lost [33: p185].

But as Lagemann goes on to point out, Dewey paradoxically remains a significant figure in education, dominating discussion in schools of education, and pointing to an ideal, even if it is Thorndike who prevails in practice. But occasionally an indirect Deweyan light shines through. A possible example of this was the dramatic reception in the West of Vygotsky's Zone of Proximal Development (ZPD). Dewey had a strong influence on Russian education in the 1920's when Vygotsky was developing his ideas, [39]. Vygotsky had certainly read Dewey's work [63: p53], and there is a close affinity with Dewey's ideal of "a community held together by participation in common activities" [20: p55]. ZPD contrasted the child's developmental level when measured by conventional tests, with the level shown under adult or peer guidance [63: p86] where the ability to follow and imitate comes into play: "*using imitation, children are capable of doing much more in collective activity or under the guidance of adults*" [61: 88]. This presupposes "*a specific social nature and a process by which children grow into the intellectual life of those around them*" [63: p88], which comes close to the collective learning through inquiry described by Dewey. In 1966 Bruner [7] introduced the word "scaffolding" to describe what is going on in ZPD, but this has been often been limited to the capacity to benefit from adult help [67], rather than from the more general sociality of "collective activity", which leads to a form of "social constructivism" [69]. Like an education based on Deweyan inquiry, ZPD in our interpretation goes very deep, and its effects, unlike those of scaffolding (if we take the metaphor literally), cannot be removed once the construction is complete.

In Psychology too, Dewey has been lurking in the background, and his influence became more apparent once the notion of feedback spread after the publication of Norbert Wiener's *Cybernetics* [66]. Later, in 1960, *Plans and the Structure of Behavior* [46] appeared, and brought together feedback of information (rather than reward and punishment) with some of the early influences on Artificial Intelligence. These included Chomsky's generative grammar [9] and Newell, Shaw and Simon on problem solving in computers [47]. The result was the TOTE (test operate, test exit), introduced as a unit of behaviour to replace the S-R model, and the authors were quick to recognise that this was similar to what Dewey had proposed in his 1896 reflex arc paper [46: p30, 43].

More generally, affinity with the Dewey scheme rather than Thorndike's shows itself when the organism, animal or human, is treated as essentially in the world, active and subject to continuous feedback as it acts, rather than a static processor of information. Examples of this Deweyan scheme are Gibson's sensori-motor systems as a model for perception [25]; the move in Robotology from cognitive representions to a focus on sensori-motor activity [6]; Jean Lave's Situated Learning [34]; and more recent work in Psychology and Philosophy on Situated Cognition [48].

Formative Assessment and Feedback. In one respect - through the notion of formative assessment - the Deweyan influence penetrated deep into the heartlands of Thorndikean territory, measurement and educational testing.

The psychologist L.L.Thurstone studied at Chicago with a close colleague of Dewey's, George Henry Mead, and spent most of his career there. Early on in his career he proposed a Deweyan model of ongoing behaviour as an alternative to the S-R scheme [59]. But his main achievements were in test theory and a more careful analysis than was usual of what is typically meant by measurement in Psychology [60]. Lee Cronbach, whose PhD was also from Chicago, continued this critical tradition within psychological measurement. His work with Meehl on Construct Validity [14] showed the limitations of psychological testing, since it measures constructs rather than reality. And he recommended that assessment be part of the learning process, rather than a test given after the learning is over [13]. Later this was labelled "formative" by contrast with the conventional "summative" assessment [50]. Summative assessment was by tests after the course had ended, whereas formative assessment was assessment during the course, designed as part of the learning process. It is closer therefore to a Deweyan rather than a Thorndikian philosophy of education, and the formative assessor joins "a community held together by participation in common activities" [20: p55]. Formative assessment involves what came to be called formative feedback. In formative feedback the student is given ongoing information about performance, and the term has replaced the concepts of reward, punishment and reinforcement. But the old S-R scheme dies hard, and many of the experiments reported on formative feedback seem quite similar to those by Thorndike and others of 80 years ago [51]. They are a long way from the feedback of a sensori-motor system that is the necessary vehicle for Deweyan inquiry. This same pattern - an apparent massive victory by the Thorndike camp, yet a persistent critical or subversive presence from the Deweyans - exists in the field of creativity, where the difference between the two viewpoints is especially marked and important given that the concept of creativity is so dominant in educational discourse.

Creative Intelligence. In literature on Creativity, which spans many disciplines and is now remarkably large and increasing every year, two distinct points of view about its nature have remained unchanged. The first is that it is a puzzling and wonderful property of the human mind that has given rise to all great human achievements.[4] The second is that it is a perfectly ordinary and basic property of all human and perhaps even animal behaviour. The reason for this strange contradiction between the two meanings, which seems to have gone largely unnoticed, may be because the modern word "Creativity" derives from two distinct ways of thinking about novelty and innovation in the world. The first of these, which sees creativity as the basic process of every mind, belongs to the Deweyan view. The second, which came later, sees creativity as a marvellous addition to the mechanical processes of ordinary thinking; this belongs to the Thorndikean view.

Figure 1. Creative and Creativity in Google's nGram

As the diagram above suggests, the popularity of words like "creative" and "creativity" is only quite recent. Originally both words were the prerogative of God, who was unique in being able to make something (the world) out of nothing. This is what

[4] "*Creativity is consensually viewed as one of the most remarkable characteristics of the human mind*." Cardosa (8:147). Creativity "*is the humble human counterpart of God's creation*" Arieti [1: 4].

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

creation meant, making something out of nothing. With this in mind, "Creative" (though not creativity) was occasionally extended to women giving birth and in the 19th century to refer to the divine and mysterious work of poets and artists[5]. This can be seen clearly in the diagram above.

But after the widespread acceptance of the Theory of Evolution by the end of the 19th century, the world itself could be seen as creative through variation and selection, with no help from God. This is how it is used in the title of Bergson's *Creative Evolution* [4] which was first published in French in 1907, and then translated into English four years later[6]. This was a book that was widely discussed, especially in the pragmatist circles around William James in Harvard and John Dewey in Chicago.

Dewey's *Creative Intelligence* was published later in 1917, and the word "creative" in the title was not being used to pick out one kind of intelligence amongst others, but to emphasise that human intelligence is inherently creative through a natural process of deliberate variation and invention. This could be the herald of a new beginning for education, since according to the traditional philosophies, "*If ever there was creation it all took place at a remote period. Since then the world has only recited lessons."* [21: p23]. Dewey thought that reciting lessons is a way of suppressing the variation that is necessary for creative intelligence to flourish. There was nothing divine about Dewey's view of creative thought, and he made little use of the popular concept of genius, instead seeing art and creativity as present in the most mundane activities: "*The sources of art in human experience will be learned by him who sees how the tense grace of the ball-player infects the onlooking crowd; who notes the delight of the housewife in tending her plants, and the intent interest of her goodman in tending the patch of green in front of the house"* [18: p3].

In this philosophy, education involves social control, but not via rules dictated by authority. Instead Dewey took as a benign paradigm of social control that of children playing games, in which the control is not from on high, but is naturally social from "a community held together by participation in common activities" [20: p55]. This underlies his practical experiments in education in the experimental schools he set up first in Chicago, later at Columbia University.

Creativity. The modern word "Creativity" came into play a little later than "creative," in the mid 1920's [45]. In 1924, around seven years after Dewey's Creative Intelligence was published, the mathematician and philosopher Alfred North Whitehead was invited to Harvard, where he developed the process philosophy for which he is best known. At the centre of this philosophy was his concept of creativity, a term he coined from the Medieval Latin "creare". [63: p208]. This was his word for the evolution of forms or species. Darwin had shown how this could be a property of organic evolution, and Whitehead applied the same basic structure (variation, and a means of fixing change) to the universe as a whole. It was his metaphysical principle through which entities are created out of flow ("*all things flow*" [65: p208]) which is more basic than the things that we experience. New forms (the solar system, new species) emerge and creativity is the power that enables this to happen. Dewey read this as a universal generalisation of his own views of human invention, managed by creative intelligence out of variation, and wrote approvingly about Whitehead and his ideas of creativity in 1937 [19]. On this view, there is nothing special about creativity. It is a basic principle of the world, and human creativity is no more than a reflection of this.

From Creativity to Social Creativity. Dewey's friend and colleague the social psychologist G.H. Mead had contributed one of the chapters in Dewey's *Creative Intelligence* of 1917 writing, "*The individual in his experiences is continuously creating a world which becomes real through his discovery"*. [41: p210] After reading Whitehead, he used the word "creativity" in his lectures during the 1920's, [41: p325], and it appeared in his best known book "*Mind, Self and Society*" [40] which was widely read.

There Mead described how any individual self is constituted by the social and physical environment it inhabits, but at the same time affects the environment in which the it is situated. More generally, the organism is partly determined by its environment, but also *"is determinative of its environment*" a more general version of the circular process described by Dewey [17]. Thus the word "creativity" is will have been familiar to the many readers of Mead and Dewey, and they would have had a common understanding that there was nothing special about it, not linked to genius but essential for the thinking of every human being and animal.[7]

Creativity as Faculty. But when creativity re-emerged in 1950 [26] it had a different meaning, and came from a different tradition of Psychology, that of Psychological measurement, therefore closer to Thorndike than to Dewey. It was not about creativity as the generation of change and novelty in the world, but referred instead to a personality characteristic. Launched by J.P. Guilford in 1950 in a presidential address to the American Psychological Association, he started by expressing astonishment at the lack of work on Creativity. He made no mention of Whitehead, Dewey or Mead, and based his concept of creativity on Factor Analysis, discovered by Charles Spearman [52]. Spearman had actually written a book called *Creative Mind* in 1930 [53], in which the word "creativity" appears, but it is not referred to by Guilford though he is likely to have known it. Spearman was a colleague of Whitehead's at UCL for several years before Whitehead left for Harvard, and may have picked the word up from him.

By partitioning similar correlations in tables from a large number of tests, Spearman had shown how to extract distinct factors of the mind, like intelligence, perseverance, memory and so on, and now creativity, which can be used to form part of the O in the S-O-R scheme. By 1950 Factor Analysis had reached a high level of sophistication, and Guilford had isolated a factor he called Creativity, based on his test of Convergent and Divergent thinking. Convergent thinking is conventional problem solving, converging on the correct solution, divergent is open ended and was thought to allow the free play of imagination, with questions like "in what different ways can you make use of a brick?" Later many other tests of creativity were devised including Torrance's Incomplete Figure Test [62] tests of insight, similar to Duncker's classic candle problem [23] and of "remote associations" Mednick et al [44].

The Creativity Bandwagon. The vastness of the bandwagon launched by Guilford has been extraordinary, and cannot be

[5] "*But this I know; the writer who possesses the creative gift owns something of which he is not always master--something that at times strangely wills and works for itself*." Charlotte Brontë in editorial preface to 1850 edition of Wuthering Heights [5, p 1iii].

[6] Translation of Bergson's *L'Évolution créatrice* from 1907 as *Creative Evolution* in 1911 [4].

[7] Vygotsky had a similar view: "*just as electricity is equally present in a storm with deafening thunder and blinding lightning and in the operation of a pocket flashlight, in the same way, creativity is present, in actuality, not only when great historical works are born but also whenever a person imagines, combines, alters, and creates something new, no matter how small a drop in the bucket this new thing appears compared to the works of geniuses.*" [64: p10-11]

Published in CEUR-WS:
FFMI workshop (Schmidt-Thieme and Janning)
In EDM 2014 Extended Proceedings (Gutierrez-Santos and Santos)

explained only by the happy Utopian vision offered by the definition that runs throughout the literature: "a creative response is novel, good, and relevant." [32: xiii]. From a comfortable seat on board in 1966, Liam Hudson wrote:

'Creativity' . . . applies to all those qualities of which psychologists approve. And like so many other virtues . . . it is as difficult to disapprove of as to say what it means. As a topic for research, 'creativity' is a bandwagon; one which all of us sufficiently hale and healthy have leapt athletically abroad [29: p100-101].

But why, what are the reasons for the astonishing success of the Creativity bandwagon, which continues to gain speed, and has left in its wake a whole set of often quite unrelated "creative industries" (media, advertising, TV, film, design, games). Even banking is given the epithet creative without a trace of irony, as well as the great entrepreneurs, led by Richard Branson. Here are just a few of the possible reasons for this remarkable juggernaut.

A. It is held together by the scientific armour of Factor Analysis, a way of constructing smooth curves from the uncertain data of questionnaires.

B. Protected by this show of rigour, it was able to break away from the aridities of Behaviourism, which had given Psychology its needed scientific respectability but had bored students for years.

C. The giants of Humanistic Psychology got on board, each with a mouth-watering trade mark to draw students to Creativity 101: Carl Rogers' self-actualization in 1954 [49], Csikszentmihalyi's flow in 1975 [15], and Maslow's peak experiences in 1968 [37]. Charles Tart was there with altered states of consciousness in 1969 [55], and Frank Barron, veteran of LSD experiments in 1963 [2]. And even Buddhism, offering an endless stream of books with titles beginning "Zen and Art of . . . ." to say nothing of Kabat-Zinn's introduction mindfulness as an essential component of creativity in 1990 [31]. It all added much needed glamour to Psychology.

D. Artificial Intelligence hitched a lift. As early as 1958 Newell et al [47], had raised the problem of creativity for computers and described a programme on ILLIAC that composed music. Computational creativity has progressed independently (there are remarkably few cross references between the two disciplines) but in parallel with Psychology's version, and has probably added a further bit of hard-nosed scientific respectability to the whole endeavour.

E. Last but not least, there has been massive funding from military and industry. As Guilford wrote in 1959, soon after the launch of Sputnik by the USSR "*The preservation of our way of life and our future security depend upon our most important national resources: our intellectual abilities and, more particularly, our creative abilities. It is time, then, that we learn all we can about those resources*" [27: p469]. The economy and safety of the West is thought to depend on the practical benefits of making things that work, from nuclear weapons to the stylish artefacts of Steve Jobs, and the secret is creativity.

3. CREATIVE FEEDBACK

But in the midst of all this razzmatazz, there was a quiet Deweyan revolution. Some of it took place on the bandwagon itself, where there are researchers who stress that Creativity is an everyday matter, and that we all possess it in our capacity for flow and mindfulness. More recently there are those who have turned away from creativity with a capital C, and looked at how a more modest Deweyan creative intelligence can be encouraged throughout education [10, 24, 30]. Dewey believed that creative intelligence is necessary for democracy to prosper, and it is fostered by what we call creative feedback.

This is the goal of MusicCircle Software project at Goldsmiths; to design an online environment to support communities of creative feedback for learning to play music. It includes the ability to upload performances, share them with others, and then seek and provide creative feedback. It is developed through a process of participatory design, working with students and other users to ensure we build what people want. Through systems such as ours perhaps we can begin to reconcile the conflicting demands of Michael Gove and Ken Robinson through evidencing clearly how learning takes place through creative feedback.

In order to understand how to design learning environments, we now set out to characterise creative feedback in more detail. We do so by describing its qualities along a number of dimensions drawing both upon our historical analysis and our combined backgrounds: teaching, programme development and management in higher education; performance and composition in music; design and implementation in software; and mindfulness and psychotherapy in practice. These qualities of creative feedback are offered in hope of receiving creative feedback to inspire the next steps.

1. CF is social. It comes from one social agent who has perceived the feedback object in some way (whether that is an output or a process of an individual) to another (the originator of the feedback object). Note this definition does not preclude students giving creative feedback to their own work.

2. CF is mindful. This incorporates at least two aspects. a) That the person giving the CF is aware of the cultural and individual context of the receiver (such as an understanding of the individual's artistic or scientific goals/methods/audiences etc.) and b) That individuals are aware of any personal judgments that are being made and can articulate these if required.

3. CF contains a degree of community awareness. a) That CF embodies an awareness of what creative feedback has occurred previously but also that it features as part of a complex and developing system b) That giving and receiving CF should be embraced equally for the community to sustain itself. It would be difficult for communities to thrive if everyone wanted to give more CF than they wanted to receive of course. CF creates a self-sustaining self-organising system where flexibility and robustness need to be balanced. Whilst each learner may have more or less knowledge about what is required to maintain such a system it is clear that it can only exist if individuals in the learning environment actively encourages engagement in CF.

4. CF is clear, the language used being unambiguous and terms used mutually understood.

5. CF is democratic. Being a tutor or student bestows no special right to giving or receiving CF (though of course one might hope that tutors have more experience and skills in giving it).

6. CF is challenging. Underpinning any creative partnership is the notion of the challenge that the each brings to the other. CF that provides the right level of challenge is arguably the most sought after feedback. To do so involves "skill in means", a Buddhist concept meaning that feedback is geared to the level and character of the student, and is always open to the student's needs.

7. CF incorporates generosity of spirit and compassion. It is an act of giving and enabling, itself an essential aspect of skill in means.

8. CF is always open to discussion and further explanation.

9. CF is comparative rather than absolute. No absolute judgment about a feedback object can be made. Comparisons (explicit or implicit) of the feedback object to other existing objects is a mindful tactic in many cases and involves skill in means. (For example, CF to a jazz piano student from a tutor could simply say

how close the student's playing is to another well-known jazz pianist and how they may want to take a listen.)

We believe the key to successful education is about providing the right kinds of environments where skills in creative feedback can develop. The role of technology is both to build new kinds of learning environments but critically to start to evidence how the creative feedback ability is correlated with learning and artistic development more generally. This may have ramifications for the way in which we think about structuring learning in schools, universities and any other kind of learning community.

4. CONCLUDING THOUGHTS

We are designing a new technology at Goldsmiths called Music Circle as part of a European Project (Practice and Performance Analysis Inspiring Social Education) through the technology-enhanced learning Programme. It is designed to allow students to upload and share performances and compositions within learning communities and then by inviting feedback from others. In order to identify the kind of feedback we wish to encourage in our system (which currently operates in a blended learning context at Goldsmiths) we have identified the term "creative feedback" which embodies a range of characteristics including clarity, mindfulness, generosity, challenge and democracy.

At the heart of the motivation for designing this system is the idea that students can learn a huge amount from the creative feedback given by others. Not only that, but that the students can develop their own abilities as musicians through the ability to give creative feedback to others. And there is little doubt that the ability to receive feedback well, to depersonalise it as much as possible and respond to it appropriately, will stand students in good stead for the world of professional musicianship. Moreover, outside the professional music world, employers will be seeking students who have the skills to work in communities that have skills in giving and receiving creative feedback. Indeed one can easily imagine a world where an employer is much more interested in the way in which a student has contributed to and benefitted from being in a community. So our manifesto and agenda for change may result in students leaving universities not with a transcript of module marks but with a detailed account of their sustained engagement with creative feedback in a community of learners.

As part of the design of the system, we are designing "creative feedback agents" that are software systems that can start to provide some aspects of creative feedback on uploaded performances and compositions. With the development of techniques from audio analysis, gesture analysis, and style analysis combined with building models of learners we are looking to build systems that can start to embody some of the CF characteristics we have identified in this paper. What is important to us is that the design of our software is underpinned by a strong educational philosophy that comes from an understanding of the historical precedents and discoveries of many before us. We want to move away from the idea that technologies are designed and built by technologists and we embrace a multi-disciplinary approach where learners, educators, designers, sociologists, philosophers, historians, psychologists and computer scientists come together to build systems but with a clear understanding of the work that has come before. Perhaps more than anything this paper is a call to arms to revive and embed a Deweyian educational philosophy that can now be both supported and evidenced through technology.

5. ACKNOWLEDGEMENTS

Our thanks to Goldsmiths, Harry Brenton, Roger Burrows, Rosie Shepperd, Matthew Yee-King, Francois Pachet, Jon McCormack, Andreu Grimalt-Reynes, Melly, Maisie and Maureen Still, Sarah Khan, Jonathan James, Chris Kiefer, Carles Sierra and Robert Zimmer. This research was supported by the FP7 Technology Enhanced Learning Program Project: Practice and Performance Analysis Inspiring Social Education (PRAISE) which includes Goldsmiths, Sony Computer Science Laboratories in Paris, the Institute of Artificial Intelligence in Barcelona and VUB, Brussels.

6. REFERENCES

[1] Arieti, S. (1976). Creativity: The Magic Synthesis. New York, Basic Books.

[2] Barron, F. (1963). Creativity and Psychological Health. Oxford, Van Nostrand.

[3] Bechtel, W. and A. Abrahamsen (1991). Connectionism and the Mind: an introduction to parallel processing in networks. Oxford, Blackwell.

[4] Bergson, H. (1911). Creative Evolution. London, Macmillan.

[5] Brontë, E. (1995). Wuthering Heights. London, Penguin.

[6] Brooks, R. (1991). "Intelligence without representation." Artificial Intelligence 47: 139-159.

[7] Bruner, J. (1966). Toward a Theory of Instruction. Cambridge, MA, Harvard University Press.

[8] Cardoso, A., et al. (2000). An Architecture for hybrid creative reasoning. Soft Computing in Case Based Reasoning. S. K. Pal, T. S. Dillon and D. S. Yeung, Springer: 147-178.

[9] Chomsky, N. (1957). Syntactic Structures. The Hague, Mouton.

[10] Claxton, G., et al. (2006). "Cultivating creative mentalities: a framework for education." Thinking Skills and Creativity 1(2): 57-61.

[11] Coffield, F. (2007). Running ever faster down the wrong road: An alternative future for education and skills. London, Institute of Education.

[12] Coffield, F. and B. Williamson (2011). From Exam Factories to Communities of Discovery: The democratic route. London, Institute of Education

[13] Cronbach, L. J. (1957). "The two disciplines of scientific psychology." American Psychologist. 12(11): 671-684.

[14] Cronbach, L. J. and P. E. Meehl (1955). "Construct validity in psychological tests." Psychological Bulletin 52: 281-302.

[15] Csikszentmihalyi, M. (1975). Beyond Boredom and Anxiety: Experiencing Flow in Work and Play. San Francisco Jossey-Bass.

[16] Dewey, J. (1889). "Review of Natural Inheritance by Francis Galton." Publications of the American Statistical Association 1(7): 331-334.

[17] Dewey, J. (1896). "The reflex arc concept in psychology." Psychological Review 3: 357-370.

[18] Dewey, J. (1934 (1980)). Art as Experience. New York, Perigree Books.

[19] Dewey, J. (1937). "Whitehead's Philosophy " The Philosophical Review 46(2): 170-177.

[20] Dewey, J. (1938 (1963)). Experience and Education. New York, Collier Books.

[21] Dewey, J. et al (1917). Creative intelligence. New York, Henry Holt.

[22] d'Inverno, M. and M. Luck (2012). "Creativity through Autonomy and Interaction." Cognitive Science, 4(3): 332-346.

[23] Duncker, K. (1945). "On problem solving." Psychological Monographs 58(5, Whole No. 270).

[24] Gauntlett, D. (2011). Making is Connecting: The Social Meaning of Creativity, from DIY and Knitting to YouTube and Web 2.0 London, Polity Press.

[25] Gibson, J. J. (1966). The Senses considered as Perceptual Systems. Boston, Houghton-Mifflin.

[26] Guilford, J. P. (1950). "Creativity." American Psychologist. 5(9): 444-454.

[27] Guilford, J. P. (1959). "Three faces of intellect." American Psychologist. 14(8): 469-479.

[28] Hargreaves, D. J., et al., Eds. (2012). Musical Imaginations. Oxford, Oxford University Press.

[29] Hudson, L. (1966). Contrary Imaginations. London, Methuen.

[30] Johnston, J. S. (2006). Inquiry and Education: John Dewey and the quest for democracy. Albany.

[31] Kabat-Zinn, J. (1990). Full Catastrophe Living. New York, Delacorte.

[32] Kaufman, J. C. and R. J. Sternberg, Eds. (2010). The Cambridge Handbook of Creativity. Cambridge, Cambridge University Press.

[33] Lagemann, E. C. (1989). "The plural worlds of educational research." History of Education Quarterly 29(2): 185-214.

[34] Lave, J. (1988). Cognition in Practice. Cambridge, Cambridge University Press.

[35] Lewis, C. C. (1995). Educating Hearts and Minds. Cambridge, Cambridge University Press.

[36] Manicas, P.T. (2002). "John Dewey and American Psychology." Journal for the Theory of Social Behaviour 33(2): 267-294.

[37] Maslow, A. H. (1968). Toward a Psychology of Being. New York, Wiley.

[38] McCormack, J. and M. d'Inverno (2014). "On the Future of Computers and Creativity", AISB 2014 Symposium on Computational Creativity, London.

[39] Mchitarjan, I. (2000). "John Dewey and the development of education in Russia." Studies in Philosophy and Education 19(1-2): 109-131.

[40] Mead, G. H. (1934). Mind, Self and Society. Chicago, University of Chicago Press.

[41] Mead, G. H. (1936). Movements of Thought in the Nineteenth Century. Chicago, University of Chicago Press.

[42] Mead, G. H. (1964). Selected Writings. Chicago, University of Chicago Press.

[43] Mead, G. H., Ed. (1982). The Individual and the Social Self. Chicago, University of Chicago Press.

[44] Mednick, M. T., et al. (1964). "Incubation of creative performance and specific associative priming." Journal of Abnormal and Social Psychology 69: 84-88.

[45] Meyer, S. (2005). "Introduction: Whitehead Now." Configurations 13(1): 1-33.

[46] Miller, G. A., et al. (1960). Plans and the Structure of Behavior. New York, Holt, Rinehart and Winston.

[47] Newell, A., et al. (1958). The processes of creative thnking. Presented before a symposium at the University of Colorado, May 14, 1958.

[48] Robbins, P. and M. Aydede, Eds. (2009). Situated Cognition. Cambridge, Cambridge University Press.

[49] Rogers, C. R. (1954). "Towards a theory of creativity." ETC: A Review of General Semantics 11: 249-260.

[50] Scriven, M. (1967). The methodology of evaluation. Perspectives of Curriculum Evaluation. R. W. R. M. Tyler, R. M. Gagné and M. Scriven. Chicago, Rand McNally.

[51] Shute, V. J. (2008). "Focus on Formative Feedback." Review of Educational Research 78(1): 153-189.

[52] Spearman, C. (1904). ""General Intelligence", Objectively Determined and Measured." The American Journal of Psychology 15(2): 201-292.

[53] Spearman, C. (1930). The Creative Mind. London, Nisbet & Co.

[54] Sudnow, D. (1978). Ways of the Hand. London, Rooutledge & Kegan Paul.

[55] Tart, C., Ed. (1969). Altered States of Consciousness. New York, Wiley.

[56] Thorndike, E. L. (1898). "Animal Intelligence: an experimental study of the associative processes in animals." Psychological Review Monograph, No 8.

[57] Thorndike, E. L. (1904). An Introduction to the Theory of Mental and Social Measurements. New York, The Science Press.

[58] Thorndike, E. L. (1911). Animal Intelligence. New York, Macmillan.

[59] Thurstone, L. L. (1923). "The Stimulus-Response Fallacy in Psychology." Psychological Review 30: 354-369

[60] Thurstone, L. L. (1927). "A law of comparative judgement." Psychological Review 34(4): 278-286

[61] Tomlinson, S. (1997). "Edward Lee Thorndike and John Dewey on the Science of Education." Oxford Review of Education 23(3): 365-383.

[62] Torrance, E. P. (1962). Guiding Creative Talent. New York, Prentice-Hall.

[63] Vygotsky, L. (1978). Mind in Society. Cambridge, MS, Harvard University Press

[64] Vygotsky, L. (2004). "Imagination and creativity in childhood." Journal of Russian and East European Psychology 42(1): 7-97.

[65] Whitehead, A. N. (1929 (1978)). Process and Reality. New York, The Free Press.

[66] Wiener, N. (1948). Cybernetics. New York, Wiley.

[67] Wood, H. and D. Wood (1999). "Help seeking, learning and contingent tutoring." Computers & Education 33: 153-169.

[68] Woodworth, R. S. (1918). Dynamic Psychology. New York, Columbia University Press.

[69] Young, M. F. D. (2008). Bringing Knowledge Back In: from social constructivism to social realism in the sociology of education. Abingdon, Oxfordshire, Routledge.