Towards New Data Quality Rules for Modeling Data Change

Nishttha Sharma

Supervised by: Dr. Fei Chiang
McMaster University, Hamilton ON, Canada

Abstract

Data is not static, and attribute value changes often trigger changes in another set of attributes. Traditional methods for analyzing data changes often treat these changes in isolation, failing to consider the broader context in which they occur. This lack of contextual awareness limits the ability to capture relationships between attributes or interpret their significance, especially when distinguishing between normal variations and potential anomalies. In this paper, we discuss the importance of context-awareness and the need to identify normal change behaviour. To achieve this, we introduce a new data quality rule, called change rule, capable of capturing changes in both antecedent and consequent attributes within ordered tuples of a relational instance.

Keywords

Data Dependencies, Dynamic Data Dependencies, Change Exploration, Change Dependency

1. Introduction

In real-world datasets, values rarely remain static as data continuously changes over time. These changes often carry critical information, revealing patterns, trends, and triggers that are essential for understanding environmental conditions, system, user behaviour and trends. Existing database systems have limited functionality to manage changes, and to identify abnormal changes, often relying on triggers to recognize out-of-bound changes. In this work, we consider changes to relational attributes for an entity. To simplify our setting, attribute changes are modeled as a sequence of ordered tuples, implicitly with respect to time. Hence, a tuple represents the value each attribute holds for an entity at a specific point in time.

Data changes occur in numeric and non-numeric attributes. Changes to numeric attributes are often measured using absolute difference, percentage change, rate of change, rolling average [1]. While these metrics are easy to compute, they fail to capture the broader context of the change, such as the influence of related attributes or the significance of the change.

For non-numeric attributes, changes are often measured using edit distances (Levenshtein, Jaro-Winkler, Hamming) [2] or set-based coefficients (Overlap, Jaccard, Dice) [3]. However, these metrics are insufficient because they ignore the semantic meaning of the changes and the context in which they occur. Context is critical because it provides the necessary information to interpret the significance of a change. Without context, changes are reduced to isolated events, which can lead to misleading interpretations of the data change.

Example 1. Table 1 shows two employees (Emp) E1 and E2 and their Position, Salary and number of employees managed (EmpMng) as of a specific Year. Consider the following changes and the need for greater context:

Numeric attribute value changes: As observed in tuples $t_1 - t_3$ of Table 1, after only two years as a Software Developer, E1 was promoted to the position of Senior Software Developer accompanied by a significant increase in salary ($68,400 to $82,000). Whereas tuples $t_9 - t_{13}$ show that E2 spent four years as a Software Developer before being promoted to Senior Software Developer with a similar salary increase as E1's (from $71,500 to $80,000). Changes in salary are typically quantified using percentage change (+19.9% for E1 and +11.9% for E2). While this provides a numerical summary of the change, it fails to account for the broader context. For instance, E1 received a larger raise after a shorter tenure and took on the responsibility of managing four employees, whereas E2 had to wait twice as long for a similar promotion and gained the responsibility of managing two fewer employees compared to E1.

Table 1
Example employee changes in position, salary.

t_{ID}	Year	Emp	Position	Salary	EmpMng
t_1	2012	E1	Software Dev	65,000	0
t_2	2013	E1	Software Dev	68,400	0
t_3	2014	E1	Sr. Software Dev	82,000	4
t_4	2015	E1	Sr. Software Dev	84,100	5
t_5	2016	E1	Sr. Software Dev	86,700	5
t_6	2017	E1	Lead Dev	96,700	25
t_7	2018	E1	Lead Dev	105,000	25
t_8	2019	E1	Manager	130,000	140
t_9	2015	E2	Software Dev	64,500	0
t_{10}	2016	E2	Software Dev	67,000	0
t_{11}	2017	E2	Software Dev	69,200	0
t_{12}	2018	E2	Software Dev	71,500	0
t_{13}	2019	E2	Sr. Software Dev	80,000	2
t_{14}	2020	E2	Sr. Software Dev	82,100	3
t_{15}	2021	E2	Sr. Software Dev	84,000	3
t_{16}	2022	E2	Sr. Software Dev	88,400	4
t_{17}	2023	E2	Lead Dev	96,100	28

Non-numeric changes within and between classes: Traditional edit distance metrics such as Levenshtein distance (LD) quantify changes based on character modifications. The transition from Software Developer to Senior Software Developer has an LD = 7, whereas for Senior Software Developer to Lead Developer, LD = 13. These values suggest that the latter change is almost twice as significant as the former despite both changes being promotions to the next position within the same class (development roles), as shown in Figure 1.

The implications of a change can be much greater between different classes. For instance, the LD between Lead Developer and Manager is 11 which suggests that this transition is smaller than the transition from Senior Software Developer to Lead Developer (LD = 13). However, this inter-

Published in the Proceedings of the Workshops of the EDBT/ICDT 2025 Joint Conference (March 25-28, 2025), Barcelona, Spain
sharmn99@mcmaster.ca (N. Sharma)

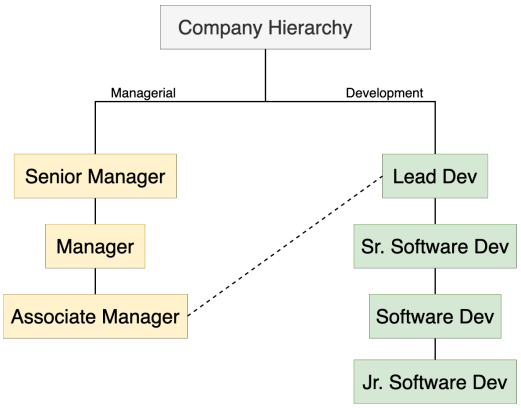

Figure 1: Hierarchy of the company in Table 1

pretation is misleading. The change from Lead Developer to Manager represents a more significant career shift as compare to the change from Senior Software Developer to Lead Developer (where both positions are in the same class), as it involves moving from a development role to a managerial position (2 levels up as per Figure 1) which is accompanied by a significant increase in the number of people managed. Existing distance measures fail to capture semantic interpretations of the data.

Problem 1: The need for context. The example highlights that not all changes are equally significant. Context is often needed to interpret data change, and there is a need to augment existing distance metrics with context.

While identifying (significant) changes is important, it is equally critical to differentiate between normal changes vs abnormal changes. Traditional methods have used declarative methods such as data dependencies of the form $X \rightarrow Y$, where X, Y are attribute sets, representing antecedent and consequent attributes. Order Dependencies (OD) [4], Sequential Dependencies (SD) [5], and Differential Dependencies (DD) [6] specify expected relationships between attribute sets. ODs introduce ordering relationships but do not explicitly quantify changes in attribute values. SDs model consequent attribute changes but do not account for variations in the antecedent attributes. DDs, while addressing changes in both antecedent and consequent attributes, apply to unordered data.

Example 2. Consider a sequential dependency (SD) stating that when ordered by Position, the change in Salary between consecutive tuples should be between 5% and 20%. For E1, the SD is violated between t_3 and t_4 with a salary change of 2.6% falling below the range. It is also violated between t_7 and t_8 , where the salary change (23.8%) exceeds the upper bound. These violations help in identifying abnormal changes. However, we also want to identify patterns where different changes in the antecedent attributes, such as changes within Position will elicit different changes in the consequent (Salary). For instance, with no change in position, salary still changes annually by 2% to 10%. Whenever there is a promotion (change in position > 0), the salary always changes by 10% to 25%. The existing dependencies do not capture relationships of this form.

To address this, we define a data quality rule called change rule. The change rule captures relationships between **changes** in attribute values of an ordered relational instance.

Problem 2: Differentiating normal vs. abnormal data change. Existing dependencies do not capture the dependence between changes from antecedent attributes to consequent attributes on ordered tuples. A declarative specification is needed that models the expected range of value change between attribute sets. We propose *change rules* to address this problem.

1.1. Challenges

- **Context representation:** Context helps to interpret the significance of data changes. Which attributes, and which subset of values are used to provide this context? Is this context time-dependent? How are existing distance measures augmented to consider this context?
- **Efficient rule mining:** Manual specification of change rules is not practically feasible, and automated solutions are needed. Determining dependent sets of attributes is important towards identifying meaningful data change. Exhaustive enumeration of all attribute sets and their values is not feasible, and efficient methods to evaluate the large space of attribute sets are needed.
- **Filtering spurious changes.** Rule mining is known to produce spurious rules. Determining which changes are most relevant and defining (support) measures that filter less meaningful changes is necessary.

1.2. Contributions

We expect to make the following contributions.

- **Context-aware change metric:** A metric for quantifying changes in both numeric and non-numeric attributes, augmenting them with contextual information from related attributes.
- **Change rules:** A new rule that captures the relationship of changes from one attribute set X to another attribute set Y across ordered tuples.
- **Change rule discovery algorithm:** An efficient discovery algorithm for change rules over ordered datasets. The algorithm adapts the FastDD method to handle ordered data [7], and identifies changes in sequential attribute values using context-aware metrics.

2. Related Work

We discuss the relationship of our work to existing metrics, data dependencies, association rules and statistical/ML approaches.

2.1. Similarity, Distance Metrics

Traditional numeric metrics analyze individual attributes in isolation, missing contextual relationships between changes in different attributes. Measures of central tendency (mean, median, mode) summarize values but can be skewed by outliers. Dispersion metrics (variance, standard deviation, IQR) capture data spread, but also prone to outlier sensitivity. Shape distribution measures (skewness, kurtosis, CV) describe asymmetry and variability but can be biased when data is highly skewed or sparse [1].

For non-numeric (categorical, text) data, cosine similarity is commonly used. Cosine similarity measures the cosine of the angle between vectors [8] and is often used with embeddings to capture semantic similarity. Overlap, Jaccard, and Dice Coefficients [3] are used to quantify the similarity and diversity of sets. Edit distance such as Levenshtein, Jaro-Winkler, Hamming quantifies the number of operations needed to transform one string into another [2].

While these metrics are widely used, they do not capture semantic distances. For example, "Software Developer" and "Senior Software Developer" have a high edit distance despite being closely related in meaning. Embedding-based approaches (e.g., BERT) address this by capturing contextual meaning but require pre-trained models and domain-specific tuning. An effective approach for measuring semantic similarity between non-numeric values is to compute cosine similarity on BERT embeddings, which allows for a context-aware representation of the data.

2.2. Data Dependencies

Order Dependencies (ODs) extend functional dependencies by enforcing ordering relationships [4]. They ensure that a positive change in the antecedent corresponds to a positive change in the consequent. However, the semantics of ODs do not declaratively capture the change in any attribute values.

Sequential Dependencies (SDs) declaratively specify the change in consequent attributes [5]. They enforce constraints on how the consequent changes in response to an instance ordered on the antecedent, i.e. when the instance is ordered on X, the changes in the consecutive Y-values will be within a range g. However, they fail to capture the change in the antecedent. Conditional SDs (CSDs) focus on identifying intervals within ordered data that satisfy a given SD. They prefer larger, contiguous intervals that capture a substantial portion of the data satisfying the embedded SD. However, the continuity of these intervals requires a trade-off with the specificity of the bound g, which is not addressed in the paper.

Differential Dependencies (DDs) model differences between any two tuples in a relation independent of the tuple ordering, i.e., if the antecedent attribute differences lie within a range g_x, then the consequence attribute value differences must lie within a range g_y [6]. By not capturing order, DDs miss critical contextual information like trends or patterns across consecutive tuples.

TSDDs [9] designed for time-series data capture temporal relationships by treating data within a given time window as an ordered set and supporting real-valued function operations. However, similar to SDs, they do not account for changes in the antecedent attributes over time. Additionally, selecting an optimal time window remains a challenge, as an overly narrow window may overlook significant trends, while a broader one risks diluting the relevance of dependencies.

2.3. Association Rules

Association rules identify co-occurrences of items within a dataset, typically expressed in the form of $\{A, B\} \rightarrow C$, stating that if items A and B appear together, then C is likely to appear as well [10]. Unlike data dependencies, which enforce constraints that all instances must satisfy, association rules identify probabilistic relationships without guaranteeing consistency. Dependencies ensure structural integrity, while association rules uncover patterns that may not hold universally.

2.4. Statistical and Machine Learning Approaches

Statistical and machine learning approaches leverage patterns in historical data to identify deviations that fall outside expected behavior. Statistical methods rely on predefined thresholds and assumptions about data distribution, while machine learning approaches adapt to complex, high-dimensional datasets.

Statistical and machine learning approaches offer complementary techniques for identifying and differentiating normal and abnormal changes in data. Statistical techniques include rule-based thresholds and hypothesis testing. For example, Z-scores and modified Z-scores are commonly used to detect anomalies by measuring how far a data point deviates from the mean, relative to the standard deviation [1]. For example, if a data point's z-score exceeds a certain threshold (e.g., 3), it may be flagged as abnormal. Similarly, control charts and statistical process control (SPC) methods monitor data streams over time, flagging points that fall outside control limits as potential anomalies [11].

Machine learning provides various techniques for distinguishing normal from abnormal changes in data, particularly through anomaly detection algorithms. Isolation Forest [12] isolates anomalies by partitioning the dataset into smaller subsets. Points that require fewer partitions to be isolated are identified as anomalies. This method works well in high-dimensional data but may struggle with datasets containing overlapping clusters or anomalies that are close to the decision boundary.

While numerous anomaly detection methods exist, our approach specifically targets anomalies in the change of attribute values. We achieve this by defining a change rule that not only identifies abnormal behavior but also captures the relationships between changes across multiple attributes.

3. Preliminaries

Let R be a relational schema on attributes $A_1, A_2, ..., A_N$, and X and Y be sets of attributes such that $X \subseteq R$ and $Y \subseteq R$. Let $I = \{t_1, t_2, ..., t_N\}$ be a relational instance of R with N tuples, ordered on X (implicitly ordered on time). The distance between consecutive tuples in I for an attribute A is given via *a context-aware distance measure*: $dist(t_i[A], t_{i+1}[A])$. We define a permissible range for $dist$ as $g_A = (p, q)$, where p, q are real values, i.e., if $dist(t_i[A], t_{i+1}[A]) \in g_a$, then $p \leq dist(t_i[A], t_{i+1}[A]) \leq q$.

We define a support function $support(\sigma, I))$ that measures the relative strength of a change rule σ in I. Naturally, we seek high-support rules to ensure that they have sufficient evidence in the instance. We introduce change rules in the next section, and focus on their discovery (as part of Problem 2).

Problem Definition: Given a minimum support threshold θ, find all change rules Σ such that I satisfies Σ ($I \models \Sigma$), such that for all $\sigma \in \Sigma$, $support(\sigma, I) \geq \theta$.

4. Current Work: Change Rules

A change rule is a novel data quality rule which describes a relationship between the changes in attributes within I. It states that when the change in the antecedent is within some range $g_x = (g_{xl}, g_{xu})$, then the corresponding change in the consequent will also be within a defined range $g_y = (g_{yl}, g_{yu})$.

DEFINITION 1. *Let π be the permutation of tuples of I increasing on X (that is, $t_{\pi(1)}[X] < t_{\pi(2)}[X] < \ldots < t_{\pi(N)}[X]$). Change rule $\sigma : X_{g_x} \rightarrow Y_{g_y}$ holds over I if for all i such that $1 \leq i \leq N-1$, when $dist(t_{\pi(i)}[X], t_{\pi(i+1)}[X]) \in g_x$ then $dist(t_{\pi(i)}[Y], t_{\pi(i+1)}[Y]) \in g_y$.*

When ordered on X, if the $dist$ between any two consecutive X-values is within the range g_x then the $dist$ between the corresponding Y-values must be within g_y. A change rule with a minimum support threshold θ holds when at least $\theta\%$ pairs of consecutive tuples in the instance satisfy the conditions of the change rule.

Example 3. Consider the change rule over Table 1:

$$\sigma : Position_{(5,15)} \rightarrow Salary_{(0.1,0.25)}$$

This rule states that if the change in Position is between 5 and 15, then the change in Salary will be between 10% to 25%. This holds true for most of the table except when E2 is promoted from Senior Software Developer to Lead Developer. In this case, the salary increase is only 8.7%, which is below the expected 10% to 25% increase. This deviation from the rule highlights that the employee received a smaller-than-normal raise with their promotion.

4.1. Discovery of Change Rules

We build upon the Differential Dependency discovery algorithm, FastDD [7] over unordered data.

- **Diff-Set Construction:** Encodes pairwise differences between all tuples into a diff-set, where each element represents a differential constraint violation (e.g., $t_i[A] - t_j[A] > \phi$), where t_i and t_j are any two tuples in a relational instance and ϕ is a numerical value. For change rules, we modify this step by using a sorted instance I on the antecedent attributes X to compute the $dist$ between consecutive pairs of tuples. This eliminates the redundant comparisons by restricting diff-set construction to adjacent tuple pairs in the sorted instance I.
- **Set Cover Enumeration:** Finds minimal subsets of differential functions (antecedent) that cover all violations of the consequent. For change rules, instead of fixed thresholds, use intervals g_x and g_y for antecedent and consequent gaps. That is, we find the minimal subsets of $dist(t_{\pi(i)}[X], t_{\pi(i+1)}[X]) \in g_x$ that cover all violations of $dist(t_{\pi(i)}[Y], t_{\pi(i+1)}[Y]) \in g_y$.

5. Conclusion and Next Steps

Data changes over time, however, we want to capture relationships between these changes. In this paper, we discussed the importance of context-awareness when capturing these changes and the relevance of identifying normal change behaviour. We introduced a new data rule, called change rule, that captures the relationship between the changes in antecedent and the changes in the consequent.

As next steps, we plan to address the aforementioned problems and challenges:

- Exploring transformer-based embeddings (e.g., BERT), to quantify and accurately capture context-aware changes in numeric and non-numeric data without compromising semantic information.
- Optimize Set Cover Enumeration by developing an efficient method to minimize the search space when identifying minimal subsets of antecedent changes that explain consequent violations.
- Consider the lagged effects of earlier changes on subsequent changes, i.e., the change in an attribute at one time step influences changes at a later time step.

References

[1] N. A. Heckert, J. J. Filliben, C. M. Croarkin, B. Hembree, W. F. Guthrie, P. Tobias, J. Prinz, Handbook 151: Nist/sematech e-handbook of statistical methods (2002).

[2] G. Navarro, A guided tour to approximate string matching, ACM computing surveys (CSUR) 33 (2001) 31–88.

[3] J. S. Cardinal, Similarity measures and graph adjacency with sets (2022). URL: towardsdatascience.com/similarity-measures-and-graph-adjacency-with-sets, [Online; posted 28-Oct-2022].

[4] J. Szlichta, P. Godfrey, L. Golab, M. Kargar, D. Srivastava, Effective and complete discovery of order dependencies via set-based axiomatization, arXiv preprint arXiv:1608.06169 (2016).

[5] L. Golab, H. Karloff, F. Korn, A. Saha, D. Srivastava, Sequential dependencies, Proceedings of the VLDB Endowment 2 (2009) 574–585.

[6] S. Song, L. Chen, Differential dependencies: Reasoning and discovery, ACM Transactions on Database Systems (TODS) 36 (2011) 1–41.

[7] S. Kuang, H. Yang, Z. Tan, S. Ma, Efficient differential dependency discovery, Proceedings of the VLDB Endowment 17 (2024) 1552–1564.

[8] W. H. Gomaa, A. A. Fahmy, A survey of text similarity approaches, international journal of Computer Applications 68 (2013).

[9] X. Ding, Y. Li, H. Wang, C. Wang, Y. Liu, J. Wang, Tsddiscover: Discovering data dependency for time series data, in: 2024 IEEE 40th International Conference on Data Engineering (ICDE), IEEE, 2024, pp. 3668–3681.

[10] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th int. conf. very large data bases, VLDB, volume 1215, Santiago, 1994, pp. 487–499.

[11] P. Qiu, Statistical process control charts as a tool for analyzing big data, Big and Complex Data Analysis: Methodologies and Applications (2017) 123–138.

[12] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.