An Online Learning-based Efficient Search System for Sufficient SPARQL Endpoints using Extended Multi-armed Bandit Algorithm

Yoshiaki Kadono and Naoki Fukuta

Graduate School of Informatics, Shizuoka University
{gs14014@s, fukuta@cs}.inf.shizuoka.ac.jp

Abstract. On searching and integrating various kinds of open data, one of crucial issues is to efficiently find sufficient SPARQL endpoints that store sufficient data to be retrieved in the query. In this demonstration, we will present our online learning-based efficient search algorithm and our prototype implementation based on an extended multi-armed bandit approach, which takes into account how the demanded data could be stored in each SPARQL endpoint by the results obtained from a series of test queries.

1 Introduction

At the beginning of 2014, we have over 500 of information sources known as endpoints of Linked Open Data(LOD)*. To utilize such endpoints effectively, we often make some federated queries to some of useful endpoints available in the world. It is not an easy task to accurately predict which endpoints could be helpful to be used for the queries, since there are so many endpoints and also those endpoints are dynamically updating their data but little information are given about how their stored data are useful for the queries[1].

To examine how an endpoint is useful and appropriate for a given query, we need to pre-check how each endpoint has useful data as well as how such data can be effective to retrieve further data stored in other endpoints. However, when we try to examine all the endpoints to be accessed at one time, it will cause serious network congestions due to its heavy network loads, as well as unwanted server loads in those endpoints. Therefore, we need to effectively predict the performance of possible endpoints to be used in a query. In this demonstration, we present a conceptual model of our online learning-based efficient search algorithm and our prototype implementation based on an extended multi-armed bandit approach, which takes into account how the demanded data could be stored in each SPARQL endpoint by the results obtained from a series of test queries.

2 Extending Budget-Limited Multi-Armed Bandits Model

The Multi-Armed Bandits(MAB) Problem classified in decision theory[2] assumes that there is a slot machine with K arms and each arm has its own revenue distribution. In

*http://sparqles.okfn.org/

the situation, the problem seeks how an algorithm can choose the best arm to obtain maximal revenues as well as the cost for estimating the revenue distributions of arms.

The Budget-Limited Multi-Armed Bandits(BLMAB) model[3][4] extends the MAB model to handle limited budgets and costs for pulling arms to estimate revenue distributions. Here, each arm i has its own cost c_i to pull the arm, and the payment will be withdrawn from the total budget B.

The Extended BLMAB model for the end-point search problem(ESP) is an extended BLMAB model, which introduces a two-dimension cost constraint, i.e., the cost for network load and the cost for query execution time. In our extended BLMAB model, the player can choose an arbitrary number of arms from the total K arms to pull in each step. Each arm i has its cost c_i to obtain the revenue based on an unknown distribution μ_i. The player has budget B so that the cost to be payed should be in the budget, and also the maximum steps is limited to be $T \in \mathbb{N}$. The objective of the player is maximizing its revenue by choosing arm i with the limited budget B within the limited T steps. Therefore, we extend the original BLMAB constraints to the following:

$$P\left(\sum_{i}^{K} N_i^A(B)c_i \leq B \cap t \leq T\right) = 1 \tag{1}$$

Here, as described in [4], we consider $N_i^A(B)$ be the random variable that represents the number of pulls of arm i by A, with respect to the budget limit B. Note that, as we mentioned, the goal is to discover the best arm to be used, rather than total revenues obtained by exploration queries.

3 Algorithm for Extended BLMAB

On the end-point search problem, we can assign each endpoint as each arm in the extended BLMAB model. Here, often each endpoint has a limited number of variations in its effective exploration queries. Therefore, the extended BLMAB model should satisfy that, let Q be the number of variations in its exploration queries, an arm i should satisfy $N_i^A(B) \leq Q$.

While the main objective of MAB is to maximize the total revenue, the main objective in end-point search problem is to find the best arm to be used for the final (but not exploration) query. Therefore, let $i(A)$ be the best arm obtained from A, $i(A*)$ be the theoretically best arm, we re-define the regression function for our extended BLMAB model as follows:

$$R(A) = i(A^*) - i(A) \tag{2}$$

4 Implementation and Preliminary Evaluation

Figure 4 shows the results of both ϵ-greedy and an extended version of the KDE algorithm[4] on the prepared test environment.

In the standard MAB problem, a typical value of ϵ is about 0.1. However, from the shown result in Figure 4, around 0.4 seems optimal. This might be caused by the

difference of objectives, since our model is based on an extended BLMAB Model. In Figure 4, the vertical axis shows the regression value defined in Equation 2. Notice that, less regression value is better in this context.

Figure 1 shows the overview of our prototype system, called *SPARQL-MAB*. We are implementing the two algorithms in the system*. Further evaluations and refinements about the models and algorithms are our future work.

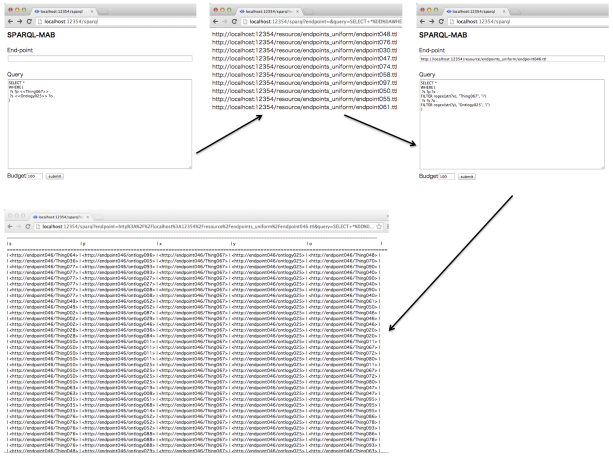

Fig. 1. An overview of our prototype system

For the preparation of the target query to be examined, an abstract query[5] is used which can include some *unknown* nodes, denoted by "<<" and ">>". Figure 2 shows an example of abstract query.

```
SELECT *
WHERE {
  ?x ?y <<Object>>.
  <<Subject>> <<Predicate>> ?x.
}
```

Fig. 2. An example of user's query

In Figure 3, two example queries are shown. Those queries were generated from the user's query shown in Figure 2.

*A demonstration is available at http://whitebear.cs.inf.shizuoka.ac.jp/sparql_mab/

```
SELECT ?a
WHERE {
  ?x ?y ?a.
  FILTER regex(str(?a), "Object", "i")
}
```

```
SELECT ?a ?b
WHERE {
  ?a ?b ?x.
  FILTER regex(str(?a), "Subject", "i")
  FILTER regex(str(?b), "Predicate", "i")
}
```

Fig. 3. Example of queries for endpoint evaluation

In this evaluation, we prepared a hundred of pseudo endpoints each of which stores 10 thousands of triples. Each triple is composed from a randomly selected concepts generated from a hundred kind of nouns or concepts in the given ontologies.

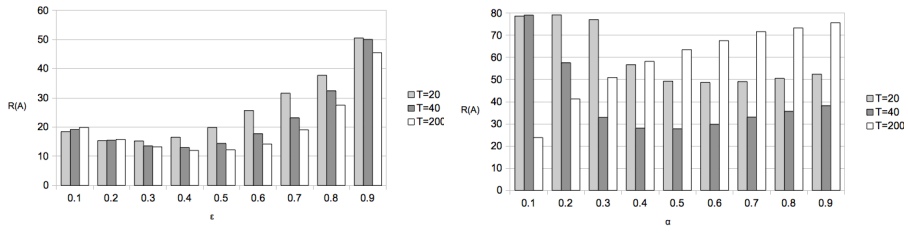

Fig. 4. Performance comparison of ϵ-greedy and KDE algorithm

References

1. T. Fujino and N. Fukuta, *Utilizing Weighted Ontology Mappings on Federated SPARQL Querying*, In, Proc. the 3rd Joint International Semantic Technology Conference, pp.331–347, 2013.
2. H. Robbins, *Some aspects of the sequential design of experiments*, Bulletin of the AMS 55:527-535, 1952.
3. L. Tran-Thanh, A. Chapman, J. Munoz De Cote Flores Luna, A, Rogers and N. Jennings *Epsilon-First Policies for Budget-Limited Multi-Armed Bandits*, In, Proc. Twenty-Fourth AAAI Conference on Artificial Intelligence(AAAI-10), pp.1211-1216, 2010.
4. L. Tran-Thanh, A. Chapman, A, Rogers and N. Jennings *Knapsack based optimal policies for budget-limited multi-armed bandits*, In, Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12), pp.1134-1140, 2012.
5. H. Noguchi, T. Fujino, and N. Fukuta,*On Implementing SPARQLoid and its Query Coding Support Framework – Querying with Weighted Ontology Mappings*, In, Proc. the 3rd Joint International Semantic Technology Conference (JIST2013),2013.(demonstration)