Towards a Flexible Experience of Data Provenance Summarization

Jisheng Pei[1], Xiaojun Ye[2]

[1] Department of Computer Science and Technology, Tsinghua University, China
[2] School of Software, Tsinghua University, China
pjs07@mails.thu.edu.cn, yexj@tsinghua.edu.cn

Abstract

In complex data analysis or applications, it is often necessary to collect, aggregate and manipulate large amount of data from multiple sources. Data provenance and their approximated summarizations have been proven to be helpful for recording and understanding user behaviors in these aspects. The growing urgency for providing timely feedbacks and the increasing need to explore more possible summarization options create a much bigger new challenge: exploring more extensive state-space under shorter response time constraints and fulfilling more complex requirements at the same time. In this work, we propose that this can be achieved by relaxing the greedy strategy adopted by existing approaches and by introducing a more flexible optimization strategy based on incremental and adaptive sampling. Our evaluations show that, compared to existing approaches, summarization processes guided by our strategy may produce more flexible and satisfying service data provenance summarization results at smaller temporal and spatial resource costs.

Keywords: Data provenance, Summarization, Urgency, Diversity, Incremental

1 Introduction

Collecting, aggregating and utilizing various sorts of user behavior data to provide helpful information has already become one of the most important missions for many online services. When using or maintaining these services, the generated results would often be much more assuring and easy to understand if users or service providers can know how a specific result generated by the service is derived, which data are contributing to that result and what roles each group of data plays in the derivation process. Data provenance [1-2] plays a centric role in the recording and understanding of such derivation process. However, the huge size of raw data provenance frequently prevents us from getting a quick and insightful understanding of the information they contain. To overcome this obstacle, approximated summarization of data provenance [3] was proposed to reduce the size of provenance expression to be presented, by grouping related raw provenance annotations into abstracted annotations according to some semantic similarity constraints. Such summarization achieves higher clarity and simplicity at the potential costs of possible errors in the results of analytics functions applied to the summarized data provenance expressions. It is because the grouped annotations can no longer provide distinctions between original data annotations in the provenance expression, which are necessary for some analytic functions, e.g. the provisioning use of data provenance expression [1]. The degree of such errors is quantitatively defined as the 'distance' of the summarization.

Existing approaches assume that a size target and distance constraint will be provided by users as input in addition to the set of provenance expressions to be summarized. Given these inputs, the summarization process will try to reach the size goal by merging annotations in the provenance expressions while still meeting the distance constraint.

Unfortunately, given non-trivial distance constraints, it is very hard (i.e. #P-hard) to find the exact smallest possible summarization [3] since the computation of accurate distance from the original provenance is already #P-hard. In [3], the state-of-art data provenance summarization algorithm adopts a greedy strategy to iteratively choose the pair of annotations with smaller summarization size or better distance performance among all candidate pairs for grouping until the target size is reached or any candidate annotation grouping operation would disrupt the distance constraint. However, as the number of provenance elements to be summarized increases, there will be more and more feasible candidate annotation mappings that need to be considered. This might introduce a heavier temporal cost for computing and comparing the error distance. On the other hand, following the greedy choice as the state-space expands also increases the likelihood of falling into the trap of local minima, which may highly

*Corresponding Author: Jisheng Pei; E-mail: pjs07@mails.thu.edu.cn
DOI: 10.3966/160792642018091905027

deteriorate the quality of the summarization.

What is more, however, is that there exists real-world requirements that implore us to explore more summarization possibilities so as to satisfy requirements that go beyond merely satisfying size and distance constraints. When trying to figure out why sometimes some rating results from a movie rating service seem to be abnormally high, providing summarizations of small sizes and small error distances alone may not be enough. Actually, it might be more helpful if the provenance elements that lead to abnormal results might be grouped together as abstracted annotations to describe their amount and common properties. However, this will imply that many more provenance expressions and possible solutions for provenance summarization need to be considered. The dilemma here is that given these additional requirements and workloads, users may want to have a real-time interactive experience with even shorter response delay.

These examples suggest that data provenance summarization processes must evolve to cope with not only the distance and size constraints imposed on the end results, but also allow more flexible and cost-efficient strategies for selecting candidate mappings during the summarization process so as to explore more possible choices and to deal with the enormous size of provenance expressions. On top of these, the summarization process should be capable of producing satisfactory results within short response times. As most existing approaches assume an absolute greedy strategy at each iteration, they become more prone to local minima trap and less suitable for meeting the above requirements as the size of provenance expressions expands and the user requirements get more complex.

In response to these challenges, we propose the strategy of Flexible Annotation Sampling and Testing (FAST). Compared to absolute greedy strategies (which, however, can still be subsumed as special cases of FAST), the key feature of FAST is its flexibility in choosing candidate mappings according to customizable strategies (instead of evaluating all candidates every time) to fulfill a broader range of data provenance summarization functionalities and non-functional goals in various scenarios and contexts. We will show that FAST is more realistic in scenarios where the sizes of provenance expressions are large and user requirements are less explicit but stricter and more complex. Furthermore, it turns out that by adopting a more flexible sampling and selection strategy, better size-distance performance can sometimes be achieved as the local minima trap can be avoided.

To facilitate FAST to be more customizable and practical, we propose a data provenance summarization process framework that adopts an incremental approach to modify and refine the selection criteria of candidate annotation mappings to be analyzed and compared. With this framework, broader range of selection requirements and more urgent time response constraints can be supported to help the summarization process adopt to varying application contexts.

The rest of this paper is organized as follows. After recalling some preliminaries about data provenance and approximated data provenance summarization in Section 2, we will first analyze the drawbacks of adopting an absolute greedy strategy and then introduce the concept of flexible data provenance summarization and the main components of Flexible Annotation Sampling and Testing in Section 3. Based on these components, we propose an incremental approximated data provenance summarization framework in Section 4 and discuss its possible applications in several representative use cases, including agile provenance summarization response and provenance stream summarization. Experiment evaluations are featured in Section 5. Section 6 discusses related work and our contributions. Finally, Section 7 concludes the paper with discussions on possible venues of future work.

2 Preliminaries

We first recall the concepts of provenance semiring and the summarization of provenance semiring expressions [2-3].

2.1 Provenance Semiring, Valuations and Provisioning

In this paper, we assume the use and the notation of semiring provenance model. However, the approach proposed in this paper can be extended easily to other provenance models [4-5]. Semiring provenance model records provenance information with a finite set X of provenance annotations, which can be understood as the basic data items or elements. For example, an annotation may be used to symbolize a row or a field in database, a user of an information system, or a transaction recorded by a system. The provenance information of the operation histories of these items is recorded using the annotations as identifiers and organized as an algebraic structure called provenance semiring. Provenance semiring has been used to capture provenance for positive relational queries. The description format of aggregation functions also supports aggregation operators such as SUM, AVG, and MAX.

Users analyze and interact with data provenance mainly through operations called provisioning, which is the operation of computing the changes to the results after applying certain modifications (specified by users) to the data items in the provenance semiring expressions. For example, when we suspect that some data items in the expression are contributed by malicious users, we may apply the provisioning

operation to remove these items from the expression according to certain scam detection rules and recalculate the end result, from which the influence of malicious users has been cancelled. This operation is implemented by assigning true / false valuations on each element in the expression, so as to control whether they might contribute to the final result or not.

2.2 Summarization through Grouping and Mapping

In order to cope with the increasing complexity and length the provenance expressions, and to help users better understand the key messages brought by these expressions, approximated summarization of data provenance was proposed in [3]. Instead of offering users the raw expressions, provenance expressions are reduced and transformed to highlight the key concept groups in the expressions only. Specifically, data provenance can be summarized by mapping multiple original annotations to abstract but meaningful annotations both to reduce the size of the expression and to convey a higher level message. Given an original provenance expression x, such mapping is denoted as h(x). However, during the summarization process, the distinction between original annotations will also be lost, and this might result in the deterioration of the quality of the provenance expression.

To counterbalance and control the loss of quality, existing works propose size, distance and semantic relatedness to be the three considerations for provenance summarization. Since the number of annotations of a provenance expression largely determines its complexity, it has been taken to be the 'size' of the provenance expression and included as part of the goal to be optimized for the summarization of provenance. To ensure the grouped annotations make sense instead of becoming a meaningless set of random annotations, only similar annotations pairs (e.g. sharing some common attributes or characteristics) will be considered for mapping. The third consideration is defined as the 'distance' between the original and summarized expression, which is in turn dependent on the specific provisioning use intended by the user. Given a set of user-specified intended provisioning uses, which are described fully by the valuation function VAL-FUNC and the combiner function φ (see [3] for detailed definition), the difference of the provisioning result of the original provenance expression and the summarized provenance expression can be evaluated and collected. For various instances of VAL-FUNC and choices of distance measures, we refer readers to [3].

Since it has been proven that computing an optimal summarization with respect to the above considerations is #P-hard (since even computing the accurate distance is already #P-hard), existing work often uses an approximated algorithm for computing the distance between two provenance expressions, and adopts an absolute greedy strategy to search for nearly optimal summarization with respect to the above three considerations (shown in Algorithm 1 below). Algorithm 1 uses greedy criteria at each iteration to select the currently best-ranking candidate annotation mapping for the summarization. As we will show in the next section, this strategy may not be flexible enough when the number of annotations in provenance expressions are large or when the user requirements for the data provenance summarization service become more complex. These serve exactly as the motivation of our study, to seek for a more flexible treatment of provenance summarization.

Algorithm 1. Existing provenance summarization process adapted from [3]

Require: p_0 (original provenance epxression) Ann (annotations in p), φ (the combiner function), V_{Ann} (VAL-FUNC function), TSIZE, TDIST (size and distance bound)

Return: Summary provenance expression p_1

```
1  p' = smallest equivalent from of p0
2  while |p'| > TSIZE or dist(p0, p', V_Ann)
   < TDIST do
3        for every h ∈ CandidateMapping (p') do
4           p_cand := h(p')
5           if CandidateScore (p_cand) is maximal
            then
6              p'_prev := p'
7              p' := p_cand
8  if dist(p0, p', V_Ann) then return p'_prev
9  return p'
```

In Algorithm 1, p_0 is first transformed to its smallest equivalent form (in terms of provisioning distance) [3] before other candidate annotation mappings are considered for summarization. A best-ranking candidate annotation mapping will be chosen at each iteration and applied to reduce the size of the provenance expression, until either TSIZE is reached or the provisioning error distance exceeds TDIST. Here, a CandidateScore function is used to rank the candidate annotation mappings according to their sizes and distances with respect to the original provenance. In [3], a linear combination of these two considerations was proposed to mix them together according to some weights (wDist and wSize).

3 Problem Solution

3.1 Requirements of Flexibility

Three variants of provenance summarization problem have been studied in [3] using the algorithm shown in Section 2. Although the goals (target size or target distance) and strategies (heuristic weights) are specified in different ways, all three forms share the common characteristics of adopting an absolute and fixed greedy strategy throughout the summarization process. However, as the size of provenance expressions increases, adopting an absolute greedy strategy becomes more costly and less effective as there are more and more candidate annotation mappings that need to be considered at each iteration. And an absolute greedy strategy would be more likely to fall into the trap of local minima at an earlier stage. These consequences further lead to the deterioration of service quality of a summarization process. Users might feel restricted about the size of provenance expressions that they might input, and suffer from a longer response time before they may observe the feedback and prepare next requests accordingly.

In order to explore possible alternative strategies for more flexible candidate annotations mapping evaluation and decision, we propose to relax the greedy selection criteria by allowing the summarization process to choose 'non-optimal' directions by some probabilities. By relaxing the greedy criteria, it becomes possible for users to achieve more "flexible" data provenance summarization in the following three ways.

Flexible Search-space. By allowing the summarization process to choose candidates that are not locally-optimal at early iterations of the summarization process, it allows a more extensive exploration of the search space of possible solutions. Thus we may avoid falling into the trap of local minima at an earlier stage. Similar observations and the strategy of "accepting the worse" has been found helpful in simulated annealing [6-7] and genetic algorithms [8].

Flexible workload capability. Once the absolute greedy criteria is removed, we could perform more flexible arrangement of summarization tasks to cope with heavier workload [9-10]. For example, we could reduce the number of candidate annotation mappings to be considered at each iteration by focusing on only some limited parts of data provenance at each iteration. In this way, the overall workload of the summarization process can be lowered significantly. In scenarios involving non-trivial workloads such as provenance repository summarization, which might consume a significant amount of computing and memory resources, such reduction would be very important.

Flexible response time arrangement. As a consequence of the previous re-arrangement of workload, it becomes possible for us to provide users with summarization feedbacks within a shorter period of response time. We may also provide users with ongoing incremental results so that users may have a more continuous and real-time experience. As the response time decreases, users or automatic summarization processes can afford to perform multiple summarization attempts before deciding whether the summarization should be continued or stopped for strategy readjustment.

In this paper, we propose to realize the above ideas of flexible data provenance summarization by modifying the existing greedy algorithm given in Algorithm 1. Instead of always choosing the local optimal candidate annotation mapping, we attempt to both narrow down the scope of candidates to be considered at each iteration and also extend the state-space to be explored by relaxing the greedy criteria.

However, the removal of absolute greedy criteria also results in the possibility of missing out 'good' choices. Therefore, we still need to decide carefully on how we should select and compare which parts of provenance expressions to be focused on and how to refine our searching and optimization strategies adaptively. Admittedly, the best answers or solutions to these issues vary due to the selection of domain and context and there might not be a perfect solution that can be found once and for all. Yet, we believe that a reasonable flexible strategy can be achieved by considering issues in the next section.

3.2 Quality Considerations beyond Distance and Size

In [3], *provenance size*, *semantic constraints* and *distance* have been established as the main considerations of provenance summarization quality. To achieve more flexible and descriptive summarization requirements specification, we further propose to quantify and include the utility of the summarization results to the users. Here utility describes how helpful the content of the summarization will be to the users. For example, if users are interested in knowing more about how data provided by young male users is contributing to the derivation of a piece of query result, then it might be better for the summarization process to group together more annotations with male property and a relatively low average age value. A summary that includes more of this kind of summarized annotations may be of higher utilities to users in this example. Of course, considerations of utility should be combined with the quality constraints specified by the users in terms of the previously identified criteria of semantic relatedness, target size and upper limit of distance. Here, we introduce the notion of utility function to measure the utility of provenance summarization to the users.

Definition 3.1 (*Quality Score).* Given some original

provenance p and summary $p' = h(p)$, we define

$$qual(p, p') = (1-\alpha-\beta)\cdot\mu(p, p') + \frac{\alpha}{|p'|} + \frac{\beta}{dist(p, p')}$$

where $0 \le \alpha, \beta \le 1$ and $0 \le \alpha + \beta \le 1$, to be the overall quality evaluation of summarization p' against original provenance p. Here, μ is a function measuring the utility of summary p' to users according to some application-specific charateristics of the annotations in p and p'. Weight parameters α and β together control how each quality consideration may affect the outcome of quality score.

There are many ways to quantitatively define μ for various scenarios. Here, we give some examples below.

(1) **Semantic Differences** assess how different each summarized annotation is against others by summing up the quantity of their semantic differences. By achieving a higher semantic differences among summarized annotations, we may avoid trivial summarization solutions in which every annotation looks almost the same and does not tell much about how different kinds of items are contributing to the derivation result. Users may specify their preference level of semantic difference.

(2) **Target Attribute Percentage** computes the percentage of occurrences of some user-specified attributes in summarized annotations. This may help users acquire summarized annotations that correspond to the attributes they are interested in.

(3) **Attribute Entropy** measures the amount of uncertainty that the original attributes in each summarized annotations might have when compared to those of the original provenance. As summarized annotations grow larger, the type of entities each annotation represents becomes more ambiguous. Attribute Entropy can be introduced to minimize ambiguity of some attribute dimensions specified by the users.

3.3 Sampling and Updating Policy

Based on the above discussion on flexible summarization requirements and quality considerations, we define the key characteristics of our *Flexible Annotation Sampling and Testing* (FAST) strategy as follows.

1. At each approximation iteration, only those candidate mappings that are related to a selected set of annotations Ann' will be taken into consideration;

2. Annotations in Ann' are sampled from the full set of unmapped annotations Ann according to some sampling distribution F;

3. A best candidate is chosen from the selected candidate annotation mappings and will be included in the current summarization solution at each iteration based on the quality considerations described previously;

4. Sampling distribution F can be updated dynamically in order to fulfill certain requirements.

By adopting FAST, the locally optimal candidate annotation mapping may not be taken into consideration if its related annotations are not selected as members of Ann' (Characteristic 2) in the first place. With FAST, users can control the probability of accepting a worse-than-local-optimal candidate annotation mapping. For example, given an iteration where there are n annotations from which k annotations are selected by equal probabilities, there will be a probability of $P = \frac{c_{n-2}^{k-2}}{c_n^k}$ that a certain pair of annotations will be selected. In other words, suppose we select one optimal pair of annotations to be grouped together at each iteration (as is done in [3]) under the above assumptions, the probability of accepting a worse-than-local-optimal candidate mapping will be $P_{worse} = 1 - P = 1 - \frac{c_{n-2}^{k-2}}{c_n^k}$. From this example, it is clear that as n increases or k decreases, P_{worse} will get greater, which intuitively suggests that the origin greedy strategy is relaxed to a greater degree. Therefore, at the beginning stages of the summarization process, it would be likely to have a more relaxed strategy, as the total number of annotations n is larger. Such strategy may allow us to explore more extensive state-space at the early stages of the summarization process, but with less likelihood to fall into a local optimal trap. On the other hand, as the summarization process approaches towards the end, n decreases and P_{worse} increases, so FAST will become more similar with the original greedy strategy. This is reasonable because at this stage, it will not be worth to risk getting worse results by dropping out of the local optimal trend.

By adopting the sampling strategy, the number of candidate annotation mappings to be considered at each iteration is reduced and consequently there will be an efficiency boost to the summarization process. However, in order to control the loss of incomplete coverage, we may force that a certain percentage of annotations have to be included in the selected set. Of course, we can also include the considerations for summarization quality during the sampling. To do this, we may adaptively raise or lower the probability for certain annotations to be sampled according to user requirements.

Furthermore, the dynamic manipulation of sampling distribution may also be adopted to re-use information produced by previous iterations. We could manipulate the distribution so that those annotations whose related annotation mappings have been observed to be less favorable (e.g. with lower over utility score or overall quality score) to be less likely to be sampled in the next iteration. By adopting these strategies, information

produced by each itereation can be passed onto the successive iterations.

To wrap up all these, we define the concept of sampling policy as follows:

Definition 3.2 (*Sampling Policy*). Given some provenance summarization quality score *qual* and original provenance p, we define

$$\sigma : (qual, p) \rightarrow < count, F, \delta >$$

to be the sampling policy for original provenance p under the requirements described by *qual*. In this definition, count refers to the number of annotations to be sampled in the next iteration, F refers to the sampling distribution or bias for each annotation to be sampled, and δ refers to the strategy for updating the sampling policy.

4 Computing Data Provenance Summarization Using FAST

4.1 Example Summarization Problems and Strategies

Equipped with Flexible Annotation Sampling and Testing technique, the original provenance summarization process can evolve to cope with more diverse and challenging scenarios in real-world applications. Below, we will present some examples where FAST can be applied.

Relaxed Greedy Search As discussed in the last section, we could adopt FAST to relax the original greedy used in [3] to avoid falling into the trap of local minima at the earlier stage of summarzation process. In such cases, we need to carefully control our sampling rate in order to achieve a reasonable rate of accepting worse local solutions.

Rapid Response In cases when a quick reponse is required, we may limit the number of candidate annotation mappings to be considered so as to reduce the computation time of each iteration significantly. As this might result in a more incomplete set of candidate mappings to be considered, it will be important to observe how such limitations may affect the quality and balance between short response time and better summarizatoin quality.

Complex Quality Goal Optimization When trying to achieve a more complex quality goal that is made up of not only size-distance constraints but also some user-specified utility goals, it would be useful if the utility information computed at each iteration of the summarization process could be passed on to the successive iterations by manipulating the sampling policy and probabilities accordingly. In this case, the design of the sampling probability feedback mechanism becomes one of the key issues that determines the quality and efficiency of the summarization outcome.

Stream Compression We argue that FAST can be used to cope with large quantity of stream-like provenance data because we can flexibly choose when to perform compressions and what parts to focus on as the provenance information streams in. For example, we may drastically lower the workload of summarization by grouping only those annotations that arrive at neighbouring periods together, or having some other common attributes. These strategies can be specified conveniently by controlling the sampling policy used in FAST.

From these examples, it is clear that the input of sampling / update policy and quality function description are the two major configurations that users need to provide. Moreover, similar to previous approaches, users may input their target size and distance to serve as the stopping criteria for the summarization process. In next subsection, we will discuss a general provenacne summarization framework based on FAST to support the above scenarios.

4.2 Provenance Summarization Algorithm Using FAST

The above discussion on flexible requirements and our FAST approach lead to Algorithm 2, a generic framework for provenance summarization based on FAST.

Compared to Algorithm 1, Algorithm 2 takes the additional sampling policy σ and quality function *qual* as input. Users may still specify the original quality constraint of size (TSIZE) and distance (TDIST).

At line 7, the algorithm starts the summarization process by selecting the first set of sampled annotations to be considered using the initial sampling policy σ. Starting from this initial set, the algorithm samples and tests candidate annotation mappings to be applied on the summarized provenance expression p' iteratively. At line 5, the iteration condition asserts that the distance of p' from p does not exceed the user-specified TDIST. At line 10, we maintain the size of the set of provenance elements related to the sampled annotation Ann', denoted as Size(p', Ann') such that it does not exceed the user-specified TSIZE. By doing this, it is guaranteed that the final annotation size of the summarized provenance expression fits into TSIZE. Through the loop from line 10 to line 15, p' is summarized iteratively using the available annotation mappings allowed by Ann'. We call an execution of this loop a *sampling-summarization cycle*. Once TSIZE is reached as a sampling-summarization cycle completes, we remove those 'unused' (i.e. not grouped by any annotation mapping) annotations among the latest sampled annotations Ann_{σ} from Ann' and put them back to the sampling pool Ann_{pool}. Furthermore, we update our sampling policy (if necessary) each time

after we complete a sampling-summarization cycle and before the next cycle begins. The summarization process terminates once all annotations have been put into consideration. If the provisioning distance of the summarized provenance expression exceeds the user specified bound TDIST at the end of the iteration, the algorithm returns its last intermediate result that fits the TDIST bound. Otherwise, a successfully summarized provenance expression is returned.

Algorithm 2. FAST Algorithm Framework

Require: The original expression p_0 with annotations Ann, combiner function φ and VAL-FUNC function V_{Ann}, sampling policy σ =<count, F, δ> quality function qual, size bound TSIZE, distance bound TDIST

Return: Summary provenance expression p_1

1 $Ann' := \phi$
2 $Ann_{pool} := Ann$
3 $p' := p_0$
4 $p'_{prev} := p'$
5 **while** dist(p_0, p', V_{Ann}) < TDIST $\wedge Ann_{pool} \neq \phi$ **do**
6 $p'_{prev} := p'$
7 select Annotations Ann_σ from Ann_{pool} according to σ
8 remove Ann_σ from Ann_{pool}
9 Ann':=Ann' $\cup Ann_\sigma$
10 **while** $|Ann'|$>TSIZE **do**
11 **for** every $h \in$ CandidateMapping(p') related to Ann' **do**
12 $p_{cand} := h(p')$
13 **if** *qual* (p_{cand}) is maximal **then**
14 $p' := h(p' \cup p_\sigma)$
15 Update Ann' according to h
16 **if** dist(p_0, p', V_{Ann}) > TDIST then **return** p_{prev}
17 (optional) update σ according to δ
18 **return** p'

Algorithm 2 is certainly not the only way of using FAST approach in data provenance summarization. Many other alternatives exist in terms of how the goal on distance constraint, sizes and user utilities are specified and optimized, how the size of Ann' is controlled instead of reducing $|Ann'|$ to fit TSIZE every time, etc. Nonetheless, we believe that Algorithm 2 can serve as an illustrating example to show how FAST operations can be integrated together with our initial optimization goals. We also notice that Algorithm 2 actually becomes equivalent with the original algorithm when the utility requirement μ measures user utility by target size and the sampling policy σ is to take all annotations at once. Thus, Algorithm 2 can demonstrate the difference in computational complexity before and after adopting FAST.

As has been discussed previously, most of the computational costs of provenance summarization originate from the repetitive execution of the costly distance computing function. In Algorithm 2, this function is included as part of the quality score computation function *qual.* Therefore, we focus our analysis of temporal complexity on the execution of the *qual* function.

Proposition 4.1 *Let qual be the function to compute the quality score of provenance summarization p'=h(p) of some original provenance p. Given a provenance expression p containing n annotations, qual will be executed for*

$$O(n \cdot k \cdot |Ann'|^2)$$

times in Algorithm 2, if the upper limit of the size of the selected set of annotations is O($|Ann'|$) and at most k annotations are sampled according to σ each time.

Proof. Let us first consider the outer loop (line 5 to line 17). Since at each iteration, at least one annotation will be selected and removed from Ann_{pool}, the loop from line 5 to line 17 will be executed for O(n) times. As for the inner while loop, a total number of $O(\frac{|Ann'|(|Ann'|-1)}{2})$ candidate annotation pairs would be evaluated and compared during each iteration. Since it takes at most k iterations before $|Ann'|$ is reduced to TSIZE ($|Ann'|$ is reduced by at least 1 at each iteration), the inner loop will be executed for at most k times. Consequently, the function *qual* will be executed for times.

$$O(n \cdot k \cdot |Ann'|^2)$$

Given Proposition 4.1, we may compare the difference in temporal complexity between Algorithm 2 and Algorithm 1. As k and $|Ann'|$ approach n, Algorithm 2 wil become the absolute greedy algorithm as Algorithm 1 is, and the costly *qual* would have been executed for $O(n^4)$ times. However, if k and $|Ann'|$ are relatively small constants that do not grow in accordance to the growth of n, the temporal complexity of Algorithm 2 can be treated as linear since $O(n \cdot k \cdot |Ann'|^2)$ can be simplified as $O(n \cdot c)$, where $C = k \cdot |Ann'|^2$. In the next section, we will demonstrate by experiments that when n is large, the difference in temporal complexities between Algorithm 2 and Algorithm 1 under natural configurations can be

very significant.

5 Experiments

In this section, we will investigate how FAST based provenance summarization algorithm performs with reference to the greedy provenance summarization algorithm (denoted as GREEDY) proposed in [3]. This investigation will be conducted in two different settings. First, since the GREEDY considers only size and distance as the optimization goal, we will compare FAST and GREEDY for their size-distance performance. We will also demonstrate the superiority of FAST in temporal efficiency when compared to GREEDY. In the second setting, we would like to investigate how cost-effective FAST could achieve user-specified utility requirements by comparing both the size-distance performance and utility score of the summarizations provided by GREEDY and FAST.

The following experiments are all carried out using the widely used provenance repository of MovieLens [11], which is also one of the benchmarks used in [3] for provenance summarization quality evaluation.

5.1 Size-Distance and Temporal Performance

We first evaluate the size-distance performance of FAST by comparing with that of the existing GREEDY approach. The evaluation was performed for provenance expressions of various sizes, ranging from 200 annotations to 2000 annotations. For clearer comparisons, we fixed the target size as 100 annotations and observed the differences in distance when we use the GREEDY stragegy and FAST strategy with different incremental sample size k (number of annotations added at each iteration) configuration.

Furthermore, we present the temporal performance of the FAST algorithm under different configurations and also compare them against those from the GREEDY. For the GREEDY approach, we chose the configuration of wDist = wSize = 0.5 as the heuristic weights when calculating the candidate score of distance and size respectively. This configuration was chosen because it seems to perform better than most other configurations in terms of the size-distance performance. Therefore, it should be a representative and reasonable choice for the GREEDY configuration. Note that in this part of the experiment, no user utility score was computed nor considered during the optimization, as this was not part of the considerations taken by the previous approach.

From Figure 1, we can see that by adopting an incremental strategy, the FAST based algorithm incurs drastically much less temporal cost (one or two magnitudes smaller) than the existing approach (GREEDY). Moreover, it shows that the temporal cost of GREEDY grows polynomially whereas FAST grows linearly. This agrees with our complexity analysis in Proposition 4.1. Figure 2 shows the temporal costs of FAST under different configurations. From this figure, it can be concluded that smaller k (sample size) may result in smaller temporal cost.

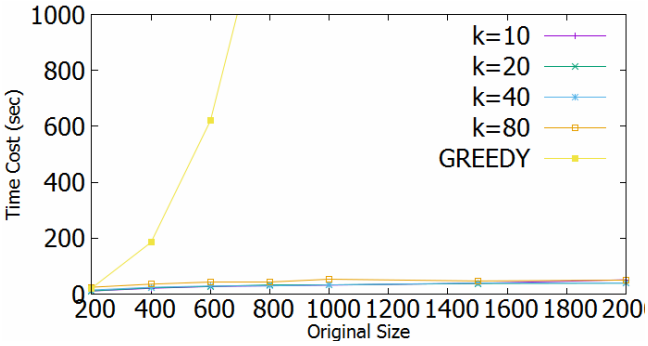

Figure 1. Time cost of FAST v. GREEDY

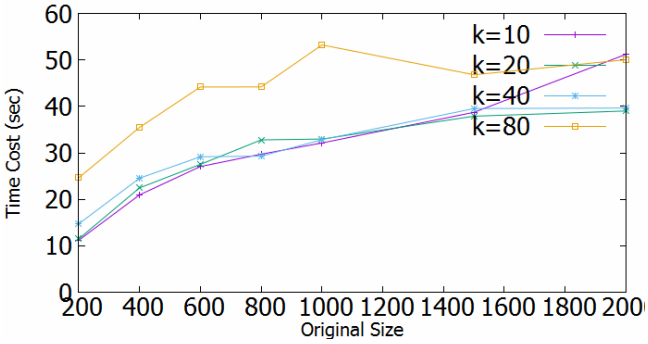

Figure 2. FAST time costs under various sample sizes

In terms of the distance performance, Figure 3 shows that the FAST based approach produces less distance than GREEDY in most cases. This can be explained as the effect of the relaxed-greedy strategy which has been discussed in the previous sections. It is worth mentioning that different configurations of k have impacts on the distance of the outcome. The figure shows that the selection of k should not neither too high nor too low so as to achieve the best performance. This is reasonable since when k is too high, the greedy strategy cannot be relaxed enough to encourage more extensive state-space exploration whereas the risk of missing good candidate mappings will increase when k is too small.

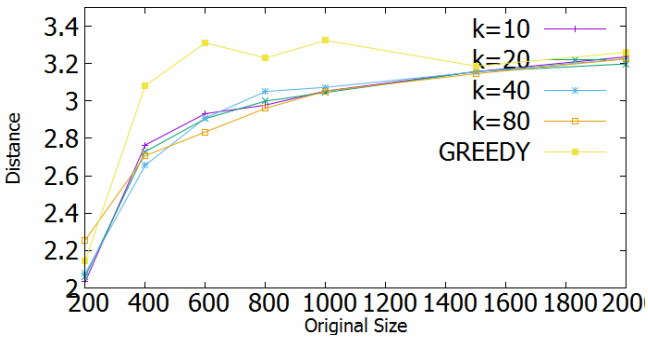

Figure 3. Distance performance of FAST v. GREEDY

5.2 User Utility Case Study: Entropy Reduction

In the second part of the experiments, we use *entropy reduction* as a use case to evaluate how well FAST based provenance summarization can respond to user utility requirements and can take balance between utility score and traditional quality (i.e. distance and size) considerations. Entropy reduction is about controlling the 'purity' of the type of items each annotation contains. Although semantic constraints have already been imposed such that only semantically similar or related annotations may be grouped together, the 'purity' of entities symbolized by each summarized annotations has so far not been quantified and considered as part of the optimization goal during provenance summarization. To quantitatively describe the 'purity' of each summarized annotation, we use the classic definition of entropy [12] to measure the amount of uncertainty in the set of attributes owned by each summarized annotation. Intuitively, when there is a higher variance among the attribute values contained in a grouped annotation, the entropy will become higher; and when the variance is lower (i.e. more 'pure'), the entropy should be lower and sometimes even close to zero.

We have implemented entropy computation and ranking as the utility function μ, and include the ranking of entropy as part of the consideration in the quality function *qual*. The attribute dimension '*Occupation*' of the users is used as the target dimension while computing entropies. In other words, the utility score and in turns the quality score favors summarized annotations with pure *Occupation* attributes. We compare the performances of our proposed algorithm under various FAST configurations, as well as the special case when we sample all candidate annotations once and for all, i.e. when our FAST becomes equivalent with GREEDY except for the fact that additional utility considerations are considered in the optimization. In the following experiments, we use the SUM aggregation function and the "Cancel One Annotation" valuation.

Figure 4 and Figure 5 show that by introducing attribute entropy as a utility consideration, entropy could be significantly reduced. As a trade-off, however, the distances become slightly greater than those produced by FAST when the entropy utility score is not considered. Furthermore, incremental FAST approach demonstrates far better temporal performance (as is shown in Figure 6).

5.3 Discussions

Based on the above observations, we can see that incremental summarization using FAST is a promising technique to reduce the time costs of summarization process. It also allows more extensive range of search possibilities for provenance summarization, which

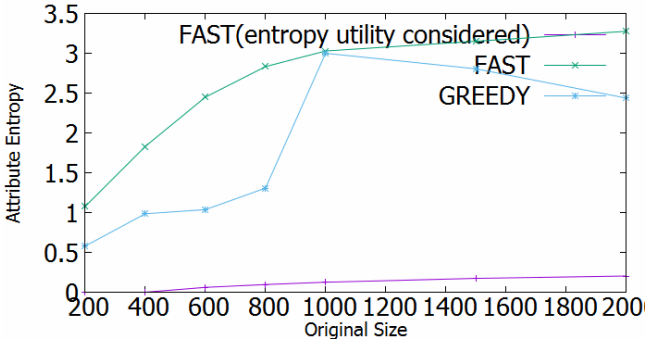

Figure 4. Entropy performance

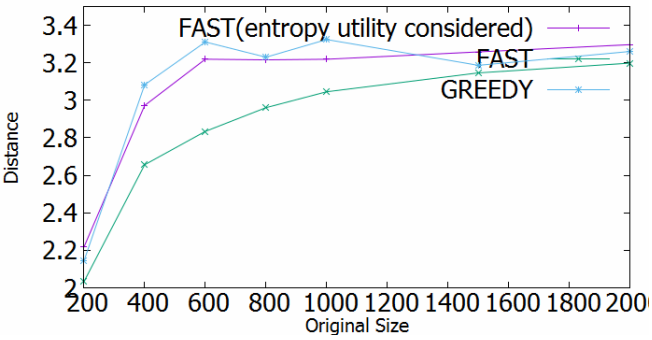

Figure 5. Distance performance

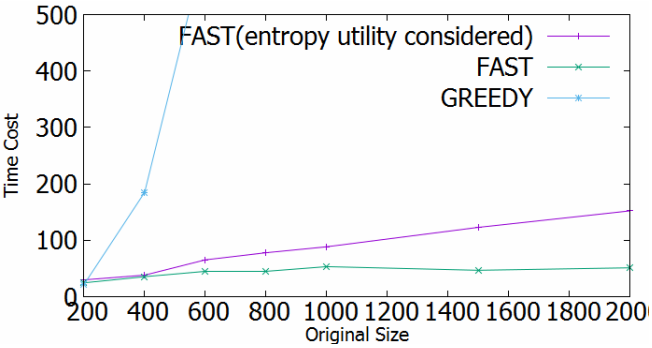

Figure 6. Temporal performance

responds to the flexibility requirements that we have identified at the earlier parts of this paper, i.e. *Flexible Search-space, Flexible Workload Capability* and *Flexible Time-Arrangment*. These new features are important to give nice and flexible user experience, which in turn will allow more user utility requirements and exploration possibilities to be considered in provenance summarization. However, there is still a large potential open space for investigatiom and exploration, e.g. the design of more dedicated sampling policy, the application of various utility score evaluation methods, etc. We believe that these would be important directions for future researches.

6 Related Work

Data provenance is related to various data structures and technologies proposed to record the generation process or the derivation process of a data item or

query results [13-14]. It has been successfully applied in areas such as compliance checking, data access control and estimation of trust [15-17]. A series of challenges have been identified [18-19] for the application of data provenance in real-world applications. In particular, the ever-growing size and complexity of provenance data make it more and more difficult for users to grasp the essential messages in provenance data in real time. In response to this challenge, researchers have looking for various methods to reduce the size or complexity of data provenance. In [20], the authors proposed that provenance data can be explored in an interactive way by presenting a limited part of provenance information at a time. This is quite similar to our incremental approach. However, no summarization or compression of data provenance is involved in [20]. In [21] lossless compression of provenance graph is proposed, but the drawback is that no high-level semantics of the underlying provenance data is considered or presented. Our work is most similar with the work in [3] from the viewpoint of the summarization of annotation mapping.

The optimization techniques in FAST share some common characteristics with existing techniques such as simulated annealing and genetic algorithms. Provenance summarization can also be viewed as a special manifestation of multi-objective optimization problem [22-23]. However, our work differs from the existing optimization techniques in the sense that it is specially designed to satisfy the complex user needs in critical environment where heavy workload, complex requirements and short response time constraints are present [24-25]. In addition, the definition of utility score and sampling policy function, together with the analysis that addresses the service requirements of provenance summarization are the unique contributions of this paper.

7 Conclusion

Complex and in-time requirements from real-world applications require existing provenance summarization approaches to evolve and consider requirements beyond merely size and distance. Additional requirements such as workload handling capability, user utility optimization and agile response time are raised and addressed in this paper. A novel technique called *Flexible Annotation Sampling and Testing* to allow incremental and flexible provenance summarization so as to achieve the above requirements by relaxing the absolute greedy strategy adopted by existing approaches and allowing more flexible sampling and search strategies to be customized by users. Experiment results show that the proposed method is promising to achieve the requirements mentioned above. The design of more dedicated sampling policies and user utility heuristics for domain specific scenarios would be interesting directions of future work.

Acknowledgements

This research was supported by the National Key Research and Development Program of China (No. 2016YFB0800901) and the program of China Scholarship Council (CSC) (No. 201606210384).

References

[1] Y. Amsterdamer, D. Deutch, V. Tannen, Provenance for Aggregate Queries, Proceedings of the *30th ACM Symposium on Principles of Database Systems*, Athens, Greece, 2011, pp. 153-164.

[2] T. Green, G. Karvounarakis, V. Tannen, Provenance Semirings, *Proceedings of the 26th ACM Symposium on Principles of Database Systems*, Beijing, China, 2007, pp. 31-40.

[3] E. Ainy, P. Bourhis, S. Davidson, D. Deutch, T. Milo, Approximated Summarization of Data Provenance, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 483-492.

[4] P. Missier, K. Belhajjame, J. Cheney, The W3C PROV Family of Specifications for Modelling Provenance Metadata, *Proceedings of the 16th International Conference on Extending Database Technology*, Genoa, Italy, 2013, pp. 773-776.

[5] L. Moreau, P. Missier, *PROV-DM: The PROV Data Model*, World Wide Web Consortium, 2013.

[6] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, May, 1983.

[7] P. Van Laarhoven, E. Aarts, *Simulated Annealing: Theory and Applications*, Springer Netherlands, 1987.

[8] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, *International Conference on Parallel Problem Solving from Nature*, Paris, France, 2000, pp. 849-858.

[9] B. Glavic, Big Data Provenance: Challenges and Implications for Benchmarking, in: T. Rabl, M. Poess, C. Baru, H.-A. Jacobsen (Eds.), *Specifying Big Data Benchmarks*, Springer Berlin Heidelberg, Berlin, 2014, pp. 72-80.

[10] K.-K. Muniswamy-Reddy, P. Macko, M. Seltzer, Provenance for the Cloud, *the 8th USENIX Conference on File and Storage Technologies*, San Jose, CA, 2010, pp. 197-210.

[11] F. Harper, J. Konstan, The MovieLens Datasets: History and Context, *ACM Transactions on Interactive Intelligent Systems*, Vol. 5, No. 4, pp. 1904-1919, January, 2016.

[12] J. Lin, Divergence Measures Based on the Shannon Entropy, *IEEE Transactions on Information Theory*, Vol. 37, No. 1, pp. 145-151, January, 1991.

[13] P. Buneman, S. Khanna, W.-C. Tan, Why and Where: A Characterization of Data Provenance, *Proceedings of the 8th*

International Conference on Database Theory, London, UK, 2001, pp. 316-330.

[14] P. Buneman, S. Khanna, W. Tan, Data Provenance: Some Basic Issues, *the International Conference on Foundations of Software Technology and Theoretical Computer Science*, New Delhi, India, 2000, pp. 87-93.

[15] J. Park, D. Nguyen, R. Sandhu, A Provenance-Based Access Control Model, *Proceedings of the 10th IEEE Annual International Conference on Privacy, Security and Trust*, Paris, France, 2012, pp. 137-144.

[16] R. Lu, X. Lin, X. Liang, X. Shen, Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing, *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, Beijing, China, 2010, pp. 282-292.

[17] I. Abbadi, A Framework for Establishing Trust in Cloud Provenance, *International Journal of Information Security*, Vol. 12, No. 2, pp. 111-128, April, 2013.

[18] C. Scheidegger, D. Koop, E. Santos, H. Vo, S. Callahan, J. Freire, C. Silva, Tackling the Provenance Challenge One Layer at a Time, *Concurrency and Computation: Practice and Experience*, Vol. 20, No. 5, pp. 473-483, April, 2008.

[19] D. Holland M. Seltzer, U. Braun, K. Muniswamy-Reddy, PASSing the Provenance Challenge, *Concurrency and Computation: Practice and Experience*, Vol. 20, No. 5, pp. 531-540, April, 2008.

[20] P. Macko, M. Seltzer, Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs, *Proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance*, Crete, Greece, 2011, pp. 1-6.

[21] Y. Xie, K. Muniswamy-Reddy, D. Long, A. Amer, D. Feng, Z. Tan., Compressing Provenance Graphs, *Proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance*, Crete, Greece, 2011, pp. 1-5.

[22] K. Deb, Multi-objective Optimization, in: E. Burke, G. Kendall (Eds.), *Search Methodologies*, Springer, Boston, MA, 2014, pp. 403-449.

[23] A. Konak, D. Coit, A. Smith, Multi-Objective Optimization Using Genetic Algorithms: A Tutorial, *Reliability Engineering & System Safety*, Vol. 91, No. 9, pp. 992-1007, September, 2006.

[24] K. Choo, The Cyber Threat Landscape: Challenges and Future Research Directions, *Computers & Security*, Vol. 30, No. 8, pp. 719-731, November, 2011.

[25] O. Hioual, Z. Boufaïda, S. Hemam, Load Balancing, Cost and Response Time Minimisation Issues in Agent-based Multi Cloud Service Composition, *International Journal of Internet Protocol Technology*, Vol. 10, No. 2, pp. 73-88, June, 2017.

Biographies

Jisheng Pei received the BS degree in Computer Software from Tsinghua University, China. He is now a Ph.D. student at the Department of Computer Science and Technology of Tsinghua University. His research interests include data provenance and process mining.

Xiaojun Ye received the B.S. degree in mechanical engineering from Northwest Polytechnical University, Xi'an, China and the Ph.D. degree in information engineering from INSA Lyon, France. Currently he is a professor at School of Software, Tsinghua University. His research interests include cloud data management, data security and database system testing.