HHS Public Access

Author manuscript

Adv Database Technol. Author manuscript; available in PMC 2016 August 26.

Published in final edited form as:

Adv Database Technol. 2016 March ; 2016: 620–623.

PROX: Approximated Summarization of Data Provenance

Eleanor Ainy,
Tel Aviv University

Pierre Bourhis,
CNRS CRIsTAL UMR 9189

Susan B. Davidson,
University of Pennsylvania

Daniel Deutch, and
Tel Aviv University

Tova Milo
Tel Aviv University

Eleanor Ainy: eleanora@mail.tau.ac.il; Pierre Bourhis: pierre.bourhis@univlille1.fr; Susan B. Davidson: susan@cis.upenn.edu; Daniel Deutch: danielde@post.tau.ac.il; Tova Milo: milo@post.tau.ac.il

Abstract

Many modern applications involve collecting large amounts of data from multiple sources, and then aggregating and manipulating it in intricate ways. The complexity of such applications, combined with the size of the collected data, makes it difficult to understand the application logic and how information was derived. *Data provenance* has been proven helpful in this respect in different contexts; however, maintaining and presenting the full and exact provenance may be infeasible, due to its size and complex structure. For that reason, we introduce the notion of approximated summarized provenance, where we seek a compact representation of the provenance at the possible cost of information loss. Based on this notion, we have developed PROX, a system for the management, presentation and use of data provenance for complex applications. We propose to demonstrate PROX in the context of a movies rating crowd-sourcing system, letting participants view provenance summarization and use it to gain insights on the application and its underlying data.

1. INTRODUCTION

Complex applications that collect, store and aggregate large-scale data, and interact with a large number of users, are commonly found in a wide range of domains. Notable examples are *crowd-sourcing applications* such as Wikipedia, social tagging systems for images, traffic information aggregators such as Waze, or recommendation sites such as TripAdvisor and IMDb. In the context of such applications, several questions arise relating to *how data was derived*. As a user, what is the basis for trusting the presented information? How do crowd contributions vary among the crowd members, based on their characteristics? If some

contribution seems wrong, how does the information change if we discard it? These questions are of fundamental importance for better understanding the application and its results.

At its core, the answer to these questions is based on the *provenance* of the collected data and resulting information, that is, *who* provided the information, in *what* context and *how* the information was manipulated. As shown in e.g. [10, 4], provenance is much more powerful than simply a log of the application execution. In particular, the algebraic model of provenance (based on semirings) has been shown to allow to correlate input data with output data; to track important details of the computational process that took place; and to further ([8]) *provision* the computation result with respect to hypothetical scenarios, namely to observe changes to the result based on changes to the input (without actually rerunning the process). Detailed tracking of provenance was thus proved to be a suitable (and necessary) vehicle for the applications that we have mentioned above.

Consider a crowd-sourcing application for movie reviews. The number of movies may be quite large and so is the number of reviewers for every movie; the *aggregated score* for the movie is computed by combining the scores of multiple users, possibly accounting for their previous reviews and for their preferences. These features and the way in which they are used in the computation should all be reflected in the provenance. In turn, provenance may be presented to *explain* results (computed ranking of movies), or to *provision* them (e.g. “how would the average movie rating change if we ignore ratings by some (group of) users?”).

However, the complexity of processing and the large scale of data also mean that detailed semiring provenance information is extremely intricate; and so presenting it in full, as an explanation to the computation, would be extremely difficult to understand. To this end, we present PROX, a system that provides approximated summarization of provenance information for complex applications. The summarization is based in part on the semantics of the underlying data (such as gender, age or occupation of users), where annotations of “similar” data items are intuitively more amenable to be grouped together. But importantly, it is also geared towards the intended use of provenance (namely explanation and/or provisioning): we define a distance function between provenance expressions that is based on the intended use, and optimizing this distance (while obtaining small expressions) is an important consideration guiding the summarization.

Demonstration

We will demonstrate the system in the context of a movies recommendation website called *Movie-Lens* [1]. We will show that while full provenance is too large to present, PROX allows for a summarized representation of the provenance that provides a concise explanation of the reviews, and further allows for approximate provisioning.

We next provide details on the technical background underlying PROX (Sec. 2), on the system implementation (Sec. 3), and on the demonstration scenario (Sec. 4).

2. TECHNICAL BACKGROUND

We (informally) introduce the main technical notions involved in the development of PROX, through examples. The full details can be found in [3].

Semiring provenance model

We first explain in general the provenance model described in [10, 5, 4]. We start by fixing a finite set X of *provenance annotations*, corresponding to the basic units of data manipulated by the application, and which can be thought of as abstract variables identifying the data. Depending on the application, these annotations may correspond to different tuples in a database, to different users, to different questions, etc. Given our set X of basic provenance annotations, the *provenance semiring* is the semiring of polynomials with natural coefficients, with indeterminates from the set X. It was shown in [10] to capture provenance for positive relational queries. Intuitively, the + operation corresponds to the *alternative use* of data (as in union and projection) and $\cdot$ to the *joint use* of data (as in join); 1 annotates data that is present, and 0 annotates data that is absent. To capture aggregate queries, in [5], relations were further generalized by extending their data domain with *aggregated values*. In this extended framework, relations have provenance also as part of their values, rather than just in the tuple annotations. Such a value is a *formal sum* $\Sigma_i\, t_i \otimes v_i$, where v_i is the value of the aggregated attribute in the i^{th} tuple, while t_i is the provenance of that tuple. We can think of $\otimes$ as an operation that pairs values (from a monoid M) with provenance annotations. Each such pair is called a *tensor*. The formal sum, presented by the $\oplus$ operation is used to capture the aggregation function.

Example 2.1—Consider a crowdsourcing application, similar to IMDb, that allows users to rate different movies and aggregates their ratings. A possible provenance expression for the movie "Pretty Woman", may e.g. be $\mathbf{P_1} = \mathrm{UID}_1 \otimes (5, 1)$ where UID_1 is a user id, and as aggregation we use a monoid of pairs to capture the aggregated rating (MAX rating with value 5 here) and how many users contributed to this aggregated value (1 here). Multiple reviews (say, for "Free Willy") can be combined using the formal sum operation, e.g. $\mathbf{P_2} = \mathrm{UID}_2 \otimes (1, 1) \oplus \mathrm{UID}_3 \otimes (3, 1) \oplus \mathrm{UID}_4 \otimes (5, 1)$ The $\oplus$ operation is given a "concrete semantics" depending on the aggregation function used to aggregate the ratings (e.g. SUM, MAX or AVG [1]).

Valuations and provisioning

An important use of semiring provenance is for provisioning, i.e. examining changes to the application's execution that are the result of some hypothetical modifications to the data. Examples include "how would the movie ratings change if we ignore some reviews (suspected as spam)?" Provenance expressions enable this using *truth valuations* applied to annotations. Intuitively, specifying that UID_1 is a spammer corresponds to mapping it to *false* (and that UID_1 is reliable to mapping it to *true*), and recomputing the derived value w.r.t this valuation. Such valuation can again be extended in the standard way to a valuation $V : \mathbb{N}[X] \mapsto \{true, false\}$.

[1]These correspond formally to a choice of operation for the aggregation monoid

Summarization through mappings

Full description of the provenance may be extremely long and convoluted, and so instead we would like to summarize the provenance expression. We formalize such summarization through the notion of mappings. Let X be a domain of annotations (for the $\mathbb{N}[X]$ semiring) and X' be a domain of annotation "summaries". Typically, we expect that $|X'| \ll |X|$. We then define a mapping $h : X \mapsto X'$ which maps each annotation to a corresponding "summary". Abusing notation, this extends naturally to a homomorphism $h : \mathbb{N}[X] \mapsto \mathbb{N}[X]'$ (i.e. define $h(x + y) = h(x) + h(y)$, $h(x \cdot y) = h(x) \cdot h(y)$) and further extends to $\mathbb{N}[X]' \otimes M$ by the standard construction $h(k \otimes m) = h(k) \otimes m$. Essentially, to apply h to a provenance expression p (we denote the result by $h(p)$), each occurrence of $x \in X$ in p is replaced by $h(x)$. The mapped expression is a "summary" of the real provenance, in the sense that we lose track of some exact annotations and summarize the provenance using the "abstract" annotations in X'.

Example 2.2—We may map user annotations to annotation summaries that intuitively reflect values of attributes of the corresponding users. Then if we map UID_3 and UID_4 to an "annotation summary" called "Female" [2], we obtain (by applying congruences in the tensor structure and the use of *max* as aggregate function), an expression describing a maximum female score of 5 collected from two users):

$$\boldsymbol{P_2'} = UID_2 \otimes (1,1) \oplus Female \otimes (5,2)$$

Another possible summary may e.g. be the result of mapping annotations UID_2 and UID_3 to the annotation "Student":

$$\boldsymbol{P_2''} = Student \otimes (3,2) \oplus UID_4 \otimes (5,1)$$

Both of these mappings do not concern the provenance expression P_1 which stays intact.

In the example, we used two possible mappings h that combine reviews based on gender or occupation. In general there may be many possible mappings and the challenge is, given a provenance expression p, to (a) define what a good mapping h is (correspondingly, what is a good summary $h(p)$), and (b) find such good h.

Quantifying Summary Quality

Several, possibly competing, considerations need to be combined.

Provenance size—Since the goal of summarization is to reduce the provenance size, it is natural to use the size of the summary, the number of annotations it consists of after simplifications, as a measure of its quality.

[2]We later describe which mappings are possible and which are preferable to ours.

Semantic Constraints—The obtained summary may be of little use if it is constructed by identifying multiple unrelated annotations; it is thus natural to impose constraints on which annotations may be grouped together. One simple example of such a constraint is to allow two annotations $x, x' \in X$ to be mapped to the same annotation in X' (or to 0 or 1) only if they annotate tuples in the *same input table*, meaning that they belong to the same domain. We further allow user-defined constraints based on equality of values of these annotated tuples in user-selected attributes, such as occupation or gender in the above examples.

Distance—Depending on the *intended use* of the provenance expression, we may *quantify the distance* between the original and summarized expression. As an example, consider a distance function designed to use provenance for *provisioning in presence of spammers*. For that we use again the notion of valuations, and consider as input to the problem a subset $\mathcal{V}_X$ of all possible valuations w.r.t. the original provenance. Intuitively $\mathcal{V}_X$ reflects possible scenarios that are of interest to the user. A central issue is how we transform a valuation in $\mathcal{V}_X$, on the original annotations to one in $\mathcal{V}_{X'}$, on the annotation summaries. We propose that this will be given by a combiner function ϕ, that sets a boolean value to $x' \in X'$ based on the truth values assigned to x annotations that were mapped to it. E.g. ϕ may be a disjunction of these values, then intuitively an annotation summary is cancelled only if all of the annotations it summarizes are cancelled.

We next define the distance between a provenance expression p and its summary $h(p)$ as an average over all truth valuations, of some property of p, $h(p)$, and the valuation. This property is based on yet another function we call VAL-FUNC, whose choice depends on the intended provenance use. For provisioning, we may e.g. use the absolute difference between the two expressions values under the valuation or, alternatively, a function whose value is 0 if the two expressions agree under the valuation, and 1 otherwise (so the overall distance is the fraction of disagreeing valuations). Similarly, when dealing with multiple expressions (such as one for each movie) we need a function to combine the VAL-FUNC values; here a natural choice is Euclidean distance.

Example 2.3—To simplify the example we assume that the scenarios include at most a single spammer. So the class of valuations consists of those assigning 0 to some single user annotation, and 1 to all others. Observe that P_2'' is at distance 0 from P_2 with respect to this class of valuations: all these valuations yield the same value with respect to the two provenance expressions (if UID_4 is mapped to true then the aggregated MAX value is 5 regardless of other truth values, and otherwise both UID_2 and UID_3 are mapped to true and so is Student). In contrast, P_2' differs from P_2 for the valuation that maps UID_4 to false and the rest to true.

Computing Summarizations

We can show that computing an optimal summarization is #*P -hard*, since even computing the distance (even under highly limiting restrictions) is already #*P* -hard. On the other hand, we have implemented an *absolute approximation* algorithm for computing the distance between two such provenance expressions, based on sampling the possible valuations. This

leads to a simple greedy algorithm. The details of the algorithm are omitted for lack of space and can be found in [3].

Related Work

Provenance models have been extensively studied in multiple lines of research such as provenance for database transformations (see [6]), for workflows (see [7]), for the web [2], for data mining applications [9], and many others, but typically full and exact provenance is presented. Provenance views have been proposed in context of workflows (see e.g. [7]), but the summarization obtained through these views is based on a notion of granularity levels, and is lossless rather than approximate. A notion of approximate provenance was proposed in [11], and somewhat resembles ours, but is limited to UCQs (and in particular allows no aggregates), geared towards probabilistic computation, and does not account for semantic constraints. Our notion of mapping to summarized annotations is also reminiscent of clustering, however the function that we optimize is one that depends on the provenance expression itself and its intended uses, which leads to different design choices and results.

3. SYSTEM IMPLEMENTATION

PROX Architecture

PROX server-side is implemented in Java and its client-side is implemented in Angular JS. This web application is deployed to Apache Tomcat server on a Windows 7 machine. The system architecture is depicted in Figure 1. The server is comprised of three major services: a selection service that allows simple restriction of the provenance according to user-defined selection criteria, the summarization service that summarizes the selected provenance; and a provisioning service that allows to use the summarized provenance for exploration of hypothetical scenarios.

PROX Web UI

We developed a web UI which contains three views. The selection view allows the user to choose movies, whose provenance she would like to observe, according to title or genre and year (as shown in Figures 2a and 2b respectively). The summarization view presented in Figure 2c shows the selected provenance and allows the user to configure parameters for the summarization algorithm. The third view presents the summary in two views shown in Figures 2d and 2e: the expression view that shows the summary in its polynomial form, as exemplified throughout this paper and the groups view that shows the groups of users that the algorithm chose to map together. For instance, for the Female group in the figure, we can see the group size (9), its aggregated (MAX) rating (AGG:4), its users, the movies they rated and their aggregated ratings. Also, on hover on group user or movie their meta data is displayed. Using this last view, the user can choose a valuation to evaluate on the current provenance by selecting annotations or attributes to cancel (assign to *false*), as shown in Figure 2f. Using the left and right arrows, the user can also view and provision the algorithm's intermediate results.

4. DEMONSTRATION SCENARIO

We will demonstrate the usefulness of PROX in the context of a movies review system. We will use a real-life movies data set taken from [1]. The first example will demonstrate how PROX can be used for provisioning. We will first select provenance by title, e.g. the movie “Free Willy”. Before we compute the summary, we will show the different parameters for the summarization algorithm. We will use the default values e.g. MAX for aggregation and will limit the number of steps to 1 for this example. Using the groups view, we will show the user the two annotations that the algorithm chose to map to an annotation summary. For the same reasons discussed in Example 2.3, we expect that the algorithm would prefer to first map the annotations of users that did not give the movie the MAX rating and we will show that this is indeed the case. To end this example, we will provision the result, by choosing a valuation that cancels the two annotations. We expect that the result would be the same as if we applied the valuation on the original expression. To prove this, using the left arrow for navigating back, we will evaluate the valuation on the original expression as well. By summarizing the original provenance, we are able to provision the result by evaluating valuations on the summary, which is more efficient.

The second example will demonstrate another important provenance use which is presentation. For this example, we will choose a large provenance expression, e.g. provenance of “Comedy” movies released in the year 2000. We will summarize it using Average aggregation and a large number of steps. We will next show the groups of annotations along with their meta data and then switch to the expression view and compare its size to the original expression size which is much greater. Finally, we will let a volunteer user select her own provenance, summarize and provision the result.

Acknowledgments

This work has been partially funded by the European Research Council under the FP7, ERC grant MoDaS, agreement 291071, by the Broadcom Foundation and Tel Aviv University Authentication Initiative, by the Israeli Science Foundation (Grant No. 1636/13), by the National Science Foundation Institute (NSF), Information and Intelligent Systems (IIS) Division (Grant No. 1302212) and by the French National Research Agency (ANR), Aggreg project.

References

1. Movielens site. https://movielens.org/
2. Provenance working group. http://www.w3.org/2011/prov/
3. Ainy E, Bourhis P, Davidson SB, Deutch D, Milo T. Approximated summarization of data provenance. CIKM. 2015
4. Amsterdamer Y, Davidson SB, Deutch D, Milo T, Stoyanovich J, Tannen V. Putting Lipstick on Pig: Enabling Database-style Workflow Provenance. PVLDB. 2012
5. Amsterdamer Y, Deutch D, Tannen V. Provenance for aggregate queries. PODS. 2011
6. Cheney J, Chiticariu L, Tan WC. Provenance in databases: Why, how, and where. Foundations and Trends in Databases. 2009; 1(4):379–474.
7. Davidson SB, Freire J. Provenance and scientific workflows: challenges and opportunities. SIGMOD. 2008:1345–1350.
8. Deutch D, Moskovitch Y, Tannen V. A provenance framework for data-dependent process analysis. PVLDB. 2014; 7(6):457–468.

9. Glavic B, Siddique J, Andritsos P, Miller RJ. Provenance for Data Mining. Theory and Practice of Provenance (TAPP). 2013
10. Green TJ, Karvounarakis G, Tannen V. Provenance semirings. PODS. 2007:31–40.
11. Ré C, Suciu D. Approximate lineage for probabilistic databases. PVLDB. 2008; 1(1):797–808.

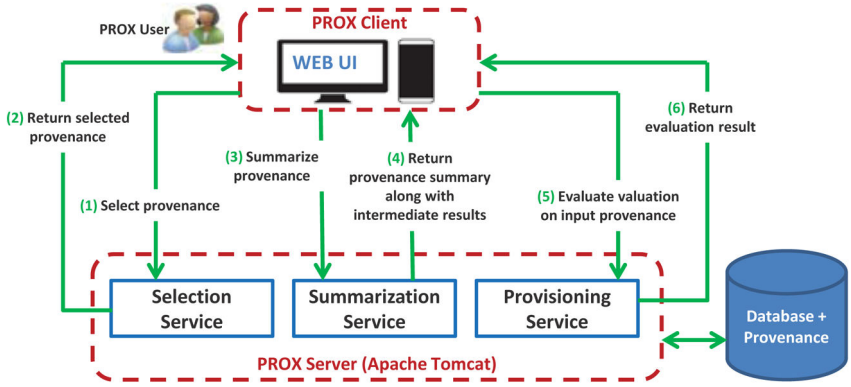
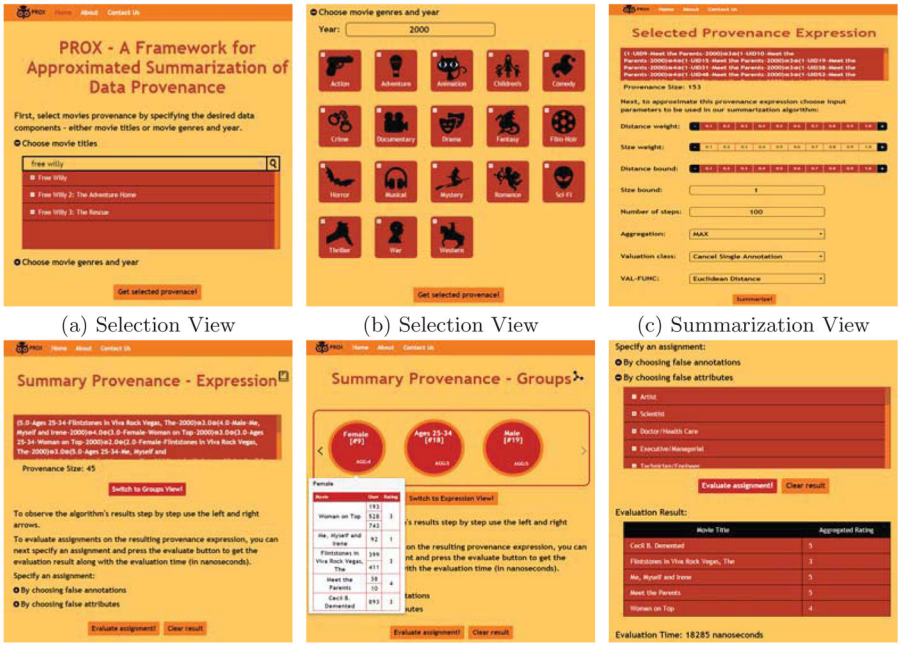


Figure 1.
System Architecture



(a) Selection View (b) Selection View (c) Summarization View

(d) Expression View (e) Groups View (f) Provisioning View

Figure 2.
PROX Web UI