Towards Web-Scale How-Provenance

Daniel Deutch, Amir Gilad, Yuval Moskovitch

Computer Science Department, Tel Aviv University
{danielde,amirgilad,moskovitch1}@post.tau.ac.il

***Abstract*—The annotation of data with meta-data, and its propagation through data-intensive computation in a way that follows the transformations that the data undergoes ("how-provenance"), has many applications, including explanation of the computation results, assessing their trustworthiness and proving their correctness, evaluation in presence of incomplete or probabilistic information, view maintenance, etc. As data gets bigger, its transformations become more complex, and both are being relegated to the cloud, the role of provenance in these applications is even more crucial. But at the same time, the overhead incurred due to provenance computation, in terms of time, space and communication, may limit the scalability of how-provenance management systems. We envision an approach for addressing this complex problem, through allowing *selective* tracking of how-provenance, where the selection criteria are partly based on the meta-data itself. We illustrate use-cases in the web context, and highlight some challenges in this respect.**

I. Introduction

Recording provenance information for data transformation has quite a few applications. Provenance may be used to explain a transformation result or asses the level of *trust* in it (see e.g. [14]); it may be used for *view maintenance* [15], or for *hypothetical reasoning* [9] where the goal is to propagate some (hypothetical) modifications made to the input, to the computation output; for computation under *access control* constraints [16], or in presence of *incomplete* or probabilistic information [20]; and in many other settings.

A particular line of research studying the tracking of *how-provenance* (also termed semiring provenance) aims at recording the computational process induced by a given query while accounting for *relevant meta-data*. The main high-level idea is to leverage the distinction between *joint* and *alternative* derivation of data, towards a corresponding algebraic structure over the domain of meta-data. Joint use of data corresponds to multiplication in this algebraic structure, while alternative use corresponds to addition. It was shown in [16] that for provenance tracking to be "aligned" with the axioms of the positive relational algebra (i.e. for equivalent queries yield equivalent provenance), the structure should be accompanied with equivalence axioms that correspond exactly to those of *commutative semirings*. Then, through an appropriate choice of semiring, i.e. appropriate semantics to addition and multiplication along with corresponding equivalence axioms, many applications such as those mentioned above are enabled. The construction proposed in [16] is generic, in the sense that it allows to track the exact computational process for positive relational algebra – up to equivalence axioms – and then "specialize" the tracked provenance in a domain of interest. Constructions for how-provenance, in a similar vein, have been proposed for various languages including queries over XML data [11], positive relational algebra enriched with aggregates [3], Datalog [10], and others.

With the emergence of Big Data and with data and computation being moved to the cloud, provenance becomes even more crucial. In particular, trust assessment is a major challenge in this setting, and provenance can serve as a "certificate" for the computation results; the need for reasoning with access control constraints naturally arises where different applications distributed among different peers manipulate data; etc. Tracking how-provenance, due to the recent development of constructions for increasingly expressive formalisms, is a promising approach in this context. But at the same time, how-provenance tracking typically incurs a significant *size and computation time overhead*. This is particularly a problem in web settings due to the large scale of data and due to the *communication overhead* that would further be incurred by accompanying data with provenance in these settings.

A distinct approach to provenance tracking has focused on the generation of execution traces as an explanation to query/program results. In a particular line of research (e.g. [6], [5], [2]), the authors have developed a model of *trace and program slicing*, so that only relevant (according to some pattern) parts of the trace is shown/tracked. In the context of workflow provenance (e.g. [4], [18]), queries over traces are used to instrument the workflow so that only relevant parts of the traces are generated. This allows to significantly reduce the size of provenance, but in contrast to the semiring approach, these approaches do not account for *meta-data*.

Our main observation is that meta-data may in fact "guide" the selection of the parts of provenance that are relevant. For instance, if facts are associated with trust scores (say, in a setting where they are extracted from different web sources / using extraction rules of varying credibility), then provenance parts that include mostly trusted facts is preferred to ones using mistrusted facts. Similarly, facts may be associated with (possibly automatically inferred) "importance" levels, which reflect on the usefulness of explaining them. We highlight in the next Section a use of confidence scores that are associated with Datalog rules, to rank derivations. Access control constraints may also affect the choice of provenance presentation (e.g. only unclassified facts should appear in the explanation), etc. So the selection of parts of the provenance to be tracked and presented is, in many cases, entangled with the computation of meta-data, enabled by provenance tracking itself.

We thus envision a scalable, general solution for provenance tracking that combines the approach of how-provenance, and

in particular the tracking of meta-data of various kinds, with ideas from program slicing that allow the selective tracking of provenance (but one that itself relies on meta-data). To be applicable to web data, the solution should (in addition to being highly scalable) be as general as possible, in three ways: (1) it should support provenance for rich data transformation models (e.g. Datalog [1], that has recently regained popularity in the context of Web data), (2) it should support meta-data from a general domain (e.g. one that is representable through semirings) and (3) it should allow for robust specification of which parts of the provenance are of interest. Such solution will potentially allow to produce "goal-oriented" concise provenance (in the sense that it is selective and is geared towards a particular meta-domain), for web-scale data.

We note that a possible approach to the problem involves the generation of full how-provenance, and then its analysis through queries. In particular, in [17] the authors have proposed a compact graph representation for semiring-based provenance (for the positive relational algebra), and a provenance query language (ProQL). However, for large-scale data combined with complex transformations (specified e.g. in Datalog), generating the full provenance as an intermediate step may be costly or even infeasible. For example, we highlight in the next Section a use case for provenance in the context of Information Extraction (the AMIE system [13]), where complex rules are applied to an already large KB in order to enrich it. Generating full how-provenance for the execution of these rules turns out to be infeasible even if compact representations are employed. Similar limitations on presenting full provenance apply in different settings, as illustrated in the use-cases that we present.

Our envisioned approach is thus as follows. We start with a program that manipulates data, specified e.g. in Datalog. In addition, we are given a specification of relevant provenance, in a language that is *meta-data-aware*. Such specification may e.g. be "all derivations of non-confidential facts", "all / top derivations for which overall trust is above a given threshold", or "all derivations that *use* only non-confidential facts", etc. These may be either manually specified, or perhaps *learned* based on feedback from users on provided explanations. Then, we *statically instrument* the program with provenance tracking directives so that whenever it is run (with a concrete database as input), it generates only relevant provenance with respect to the specification.

There are many challenges in the development of such a solution. In particular, as mentioned above, we aim at a generic and robust solution in the sense of supporting expressive transformation languages, rich meta-data domains, and expressive formalisms for specification of relevant provenance that possibly rely on meta-data. Supporting this generality while allowing for efficiency and scalability is a challenging task. Additionally, large-scale computation is often *distributed*, and thus the (selective) generation of provenance needs to be correspondingly distributed among the peers.

We next outline some use-cases of the envisioned solution.

II. Use Cases

We envision a general-purpose system for the selective tracking of provenance, based on two principles: (1) tracking how-provenance, namely the ways in which facts are derived, and (2) restricting the tracked provenance, (partially) based on meta-data. We next highlight several use cases of such system, and then some challenges with respect to them. Beyond these, we see as a grand challenge the development of a unified approach for *selective* how-provenance that can accommodate these as well as other applications, as specific cases.

A. Ranking Explanations in Information Extraction

AMIE [13] is a system for mining logical rules from Knowledge Bases (KBs), which can then be used to address incompleteness of KBs, gradually deriving additional new facts and introducing them to the KB. Since the rules are automatically mined, they are quite complex and furthermore there is an inherent uncertainty with respect to their validity – which calls for *explanation* of results obtained using them, so that users can understand how they were derived and consequently whether to trust them. Such explanations may also provide insights into the behavior of the complex system. However, the complexity of rules and the KB vast size also lead to the infeasibility of presenting – or even computing – full provenance. Just to illustrate, the number of different rules that may be used to *directly* derive a single fact in AMIE exceeds 20. Many of these rules are *recursive* (in the Datalog sense), and are very complex. Naturally, derivations based on these rules use additional facts, which again have many possible rules deriving them, etc.

Thus, we have proposed in [8] to leverage an important meta-data that is available in this context, namely *confidence scores* associated with different rules, towards the selective tracking of provenance. By means of maintaining how-provenance we can propagate and aggregate these confidence scores to compute a confidence score for *explanations*, which in this case correspond to *derivation trees* of facts extracted using the rules. Then, selective tracking of provenance is based on two facets: (1) a pattern that is specified manually and restricts interest to a particular set of derivations, and (2) ranking of derivation trees based on their aggregated confidence. So selective tracking of provenance amounts in this case to finding the top-k derivation trees, based on numeric meta-data associated with them, which is their aggregated confidence score. Our preliminary results indicate that this may be performed efficiently, alongside with standard (semi-naive) evaluation of the rules.

An intriguing avenue of research to explore in this context is the automatic inference of "provenance patterns" that are of interest; for instance, such inference may be based on user feedback to explanations that were already presented. Additionally, the ranking of provenance could be based on further meta-data beyond confidence levels in rules (where available), and may itself be automatically inferred.

B. Monitoring and Analyzing Declarative Networking

Declarative Networking, i.e. the use of declarative languages such as Datalog to specify network protocol and services, has recently gained popularity. Provenance tracking is very valuable in such settings, since it may be used to better understand the behavior of the system and of participating peers, and ultimately to optimize the network and its use. How-provenance for programs that are typically used in this setting may e.g. track the full sequence of routers that a message can follow (where the routes have been computed by the declarative program), along with the internal logic of applications deciding the routing. However, with large networks and high communication load, tracking full how-provenance may become infeasible. Furthermore, not all provenance is "equally interesting" for the network owners. For instance, the owners may be particularly interested in traffic going through routers at particular location, or ones that are already congested, etc. As another example, for routed messages, the owners may not be interested in viewing as provenance all possible routes it may follow but instead only those with minimal *latency*. This may be captured by associating appropriate meta-data (expected latency, connectivity constraints etc.) with modeled routers (serving as data to the declarative networking program), and then tracking relevant provenance based on this meta-data (such as paths of minimal latency). In addition, a declarative language for specifying selection criteria on provenance tracking, of the form that we envision, could be used to specify the owners' interest in provenance tracking for particular messages, routers or routes, based on meta-data.

In addition to challenges similar to those mentioned for the previous use-case, here there is additional challenges stemming from *distribution*. Since manipulation of data is performed in a distributed manner, the tracking of provenance should be appropriately distributed as well. How to do so efficiently, while accounting for selection criteria and ranking of provenance, is an intriguing problem to be studied.

C. Explaining Web Application Executions Under Privacy Constraints

Consider a declaratively defined "collaborative" web application that is distributed across peers which exchange and combine Data in intricate ways (see e.g. [12], [7], [19] for examples). Due to the typical complexity of such applications, explaining their executions through provenance is useful for understanding the application behavior. Different peers in the network may be interested in explanations for different parts of the computation, that are relevant for them. The envisioned system could allow each peer to specify a "pattern" of interest, which will lead to presenting (and/or tracking) possibly different parts of the provenance for different peers.

Moreover, computation may involve private data of some peers, who are willing to expose their data to the application, but not to (some) other peers (this is e.g. typically the case in social networks). This means that computation and presentation of provenance need to avoid compromising privacy constraints, which in turn are often considered as meta-data, and computed for derived facts through means of how-provenance. So here the selective tracking/presentation of provenance based on meta-data is required not only as an optimization for the overhead incurred by provenance, but also due to privacy constraints. Precisely defining privacy-aware selective provenance, and then computing it in the context of complex Web applications, is an intriguing task that could fit the general proposed paradigm.

III. Conclusion

We have illustrated in this short paper an approach for provenance tracking that employs detailed, meta-data-aware how-provenance, while significantly reducing provenance size, based on criteria that may depend on the meta-data itself. We have highlighted some use cases for different applications and different meta-data domains that could benefit from the approach, as well as challenges in realizing it.

Acknowledgments: This research was partially supported by the Israeli Ministry of Science, by the Isareli Science Foundation (ISF), by the Broadcom Foundation and Tel Aviv University Authentication Initiative, and by the Advanced ERC grant Modas (grant 291071).

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] Umut A. Acar, Amal Ahmed, James Cheney, and Roly Perera. A core calculus for provenance. *Journal of Computer Security*, 21(6):919–969, 2013.

[3] Y. Amsterdamer, D. Deutch, and V. Tannen. Provenance for aggregate queries. In *Proc. of PODS*, 2011.

[4] O. Biton, S. Cohen Boulakia, and S. B. Davidson. Zoom*userviews: Querying relevant provenance in workflow systems. In *VLDB*, 2007.

[5] James Cheney, Umut A. Acar, and Roly Perera. Toward a theory of self-explaining computation. In *In Search of Elegance in the Theory and Practice of Computation*, pages 193–216, 2013.

[6] James Cheney, Amal Ahmed, and Umut A. Acar. Database queries that explain their work. *CoRR*, abs/1408.1675, 2014.

[7] E. Damaggio, A. Deutsch, R. Hull, and V. Vianu. Automatic verification of data-centric business processes. In *BPM*, 2011.

[8] D. Deutch, A. Gilad, and Y. Moskovitch. selp:selective tracking and presentation of data provenance (demo). In *ICDE*, 2015. To appear.

[9] D. Deutch, Z. G. Ives, T. Milo, and V. Tannen. Caravan: Provisioning for what-if analysis. In *CIDR*, 2013.

[10] D. Deutch, T. Milo, S. Roy, and V. Tannen. Circuits for datalog provenance. In *ICDT*, 2014.

[11] J. N. Foster, T. J. Green, and V. Tannen. Annotated xml: queries and provenance. In *PODS*, 2008.

[12] Xiang Fu, Tevfik Bultan, and Jianwen Su. Analysis of interacting BPEL web services. In *WWW*, pages 621–630, 2004.

[13] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422, 2013.

[14] Floris Geerts, Grigoris Karvounarakis, Vassilis Christophides, and Irini Fundulaki. Algebraic structures for capturing the provenance of SPARQL queries. In *ICDT*, pages 153–164, 2013.

[15] T. J. Green, G. Karvounarakis, Z. Ives, and V. Tannen. Update exchange with mappings and provenance. In *VLDB*, 2007.

[16] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, 2007.

[17] Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. Querying data provenance. In *SIGMOD Conference*, pages 951–962, 2010.

[18] Z. Liu, S. B. Davidson, and Y. Chen. Generating sound workflow views for correct provenance analysis. *ACM Trans. Database Syst.*, 36(1), 2011.

[19] Royi Ronen and Oded Shmueli. Soql: A language for querying and creating data in social networks. In *ICDE*, pages 1595–1602, 2009.

[20] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.