ProvDS: Uncertain Provenance Management over Incomplete Linked Data Streams

Qian Liu

Open Distributed Systems, TU Berlin—Germany

1 Problem statement

Data processing in distributed environments is often across heterogeneous systems, bearing the need to exchange provenance information, such as, how and when data was generated, combined, recombined, and processed. Distributed systems involve multiple participants and data sources which can produce unreliable, erroneous data. Besides, there maybe exists oceans amount of data to deal with, e.g., in fields such as Internet of Things (IoT) and Smart Cities. Therefore, dynamic stream-based data processing mechanisms are more reasonable in these environments.

Hence, we propose provenance and recovery-aware data management techniques that take dynamic, incomplete streams as inputs, and simultaneously recover the missing data and compute the provenance over the reconstructed streams. Unlike traditional provenance management techniques, which are applied on complete and static data, our research focuses on dynamic and incomplete heterogeneous data.

2 Relevancy

Provenance provides the knowledge of how a piece of data or query results were produced. Provenance is especially important in open environments such as the IoT where data can be discovered, modified or mashed-up by arbitrary parties. In fact, the IoT involves many uncoordinated data producers, which generate vast amounts of data with high velocity (e.g., from sensors, mobile devices). This variety of data streams yields very heterogeneous and incomplete data. In this context, it is crucial to establish the trust and the transparency of the data to facilitate reliable insight to the end user. Linked Data [1] provides means to integrate static heterogeneous data while Linked Stream Data [2] extends these paradigms to data streams. More specifically, Linked Stream Data allows us to integrate static knowledge with dynamic data from distributed sources.

The outcomes of the research proposed in this paper will be applicable over a range of domains and will benefit users handling different data analysis tasks (e.g., prediction, trend analysis), real-time events streams processing, as well as stream reasoning over Linked Data. Such environments are vulnerable to the error propagation problem since these errors cannot be traced without any knowledge of provenance. In these fields, our approach will allow end-users to mitigate the influence of incomplete or erroneous data on their applications.

Our solution will allow users to better grasp how the data and query results were produced, which is a key element in establishing transparency and data governance.

3 Related work

Various types of provenance information have been formalized semantically in the Open Provenance Model [3]. In the same context, the W3C PROV model [4] has been introduced to standardize a recommendation for the exchange of provenance over the Web. Such provenance models provide a way to describe where data originated and how it is processed and propagated.

On the systems side, Glavic et al. introduced Perm [5], which was a provenance-aware DBMS able to compute, store, and query relational provenance data. Their work assumed a strict schema, whereas Linked Data Streams are by definition schema free.

Provenance of Linked Data is often attached to a dataset descriptor that is typically embedded in a Vocabulary of Interlinked Dataset (VoID) file [6]. The HCLS Community Profile[1] provides detailed descriptions what metadata should be provided by linked datasets. The support for provenance is one reason for the introduction of named graphs in the latest version of RDF [7]. Other approaches, such as nanopublications[2], extensively use named graphs to enable subsets of linked data to be referrable and to describe pieces of data. Provenance can also be attached to data through annotation-based techniques. These techniques assign annotations to each of the triples in a dataset and then track these annotations as they propagate through either the reasoning or query processing pipeline. Formally, these annotated data are represented by algebraic structures such as communitative semirings [8]. Theoharis et al. [9] proposed a comprehensive theoretical foundation for tracing provenance in RDF queries. Provenance in Linked Data has also been used to determine and propagate trust values [10].

Zimmermann et al. [11] proposed annotating triples with temporal data and provenance values referring to the source of the triple. The authors used a standard triple-oriented data model and included temporal and provenance annotations. A triple takes the form of a statement (Subject, Predicate, Object, Annotation), i.e., an N-Quad. Such statements could be stored in any triplestore supporting N-Quads. A similar approach was described by Udrean et al. [12] where the authors extended RDF for temporal, uncertain, and provenance annotations. The main focus of this work was to develop a theoretical model to manage such metadata information. The authors proposed also a query language which allowed users to query such metadata. Unlike Zimmermann's solution, Udrean's solution annotated the predicates with provenance information. In the same context, Nguyen et al. [13] suggested to use a singleton property instead of RDF reification or named graphs to describe provenance.

The implementations described above of annotated RDF provenance approaches often do not address "provenance-tracing", i.e., how a query result was

[1] http://www.w3.org/TR/hcls-dataset/
[2] http://nanopub.org/wordpress/

constructed. Moreover, these implementations are only applied to small (around 10 million triples), static, and complete datasets which focus on inferred triples but do not aim at reporting provenance evolution (e.g., with provenance polynomials, which are algebraic structures representing how data was combined) for SPARQL query results.

Wylot et al. in their system TripleProv [14, 15] introduced techniques to store, trace, and query provenance information in Linked Data. TripleProv returns a description of the way the results of an RDF query were derived; specifically, it gives an explanation how particular pieces of data were combined to deliver the results. The system allows the user also to scope the query execution with provenance information, as the user can input a provenance specification of the data he wants to use to derive the results.

None of the approaches described above specifically target incomplete Linked Data Streams. These approaches were designed for static data, therefor they do not take into account dynamics of input streams and they do not allow users to execute continuous queries over such streams. Moreover, due to the employed storage models (multiple indicies and provenance annotations) their performance deteriorates for dynamic data. Besides, all these techniques assume that data is complete; to the best of our knowledge there are no methods to deal with incomplete or re-constructed dynamic Linked Data.

4 Research questions

In this research, we will investigate two questions in provenance management:

- **(Q1) How can we enable dynamic provenance tracing in the context of incomplete Linked Data Streams?** This question will investigate means to deliver the user a *continuous provenance*, i.e., a dynamic provenance trace of the continuous query that executes over a long period of time on dynamic data. The returned provenance trace represents how particular pieces of data were produced and combined to deliver the results. We will also introduce methods to track provenance of the recovery process, i.e., to provide recovery-aware provenance for continuous queries over multiple incomplete Linked Data Streams.
- **(Q2) How can we derive probabilistic provenance graphs on recovered data?** Continuing with tracking provenance of the recovery process for different sources of incomplete Linked Data Streams in Q1, this question will investigate methodology to derive probabilistic provenance graph based on Linked Data Streams before and after recovery. To achieve this goal, we will use a moving correlation on different sources of Linked Data Streams and will compute their fractional contributions to the recovered data, i.e., the fractional provenance of the recovered pieces of data. Afterwards, we will trace back the provenance graphs of the source elements (elements used in the recovery from different sources) and use them to reconstruct a provenance graph of the recovered pieces of data to assess the probability of the reconstructed provenance graph (see Section 6 for details).

5 Hypotheses

An accurate provenance of query results over incomplete and recovered data in dynamic distributed environments can be computed at low costs in terms of memory consumption and query execution time (bound complexity). State-of-the-art research [16, 17, 14] for static data shows necessary performance overhead of 20%-30%. Our hypothesis is that the same efficiency is possible for the case of dynamic data.

6 Approach

Our main goal is to efficiently handle incomplete and dynamic data. Such data has to be recovered online and the lineage, i.e., full history of what happened to data as it went through diverse processes, of the recovered data has to be provided. We will also include the information of the recovery process in the provenance description of query results, such that users will be able to know how recovered pieces of data have influenced the results of their query.

To achieve this goal, we propose the following research contributions to answer the two research questions (Q):

- **Continuous Provenance Polynomial for describing provenance in a dynamic setting (Q1)**
- **Online provenance-aware recovery of incomplete Data for recovering incomplete data and tracing provenance of the recovery process (Q1)**
- **Fractional provenance of recovered incomplete Linked Data streams for discovering provenance of data recovered with external recovery techniques (Q2)**

Continuous provenance polynomial over Linked Data streams. The goal of this task is to provide a dynamic provenance trace of continuous queries, i.e., a continuous provenance polynomial. This task builds on TripleProv [14] generalizing provenance polynomials to incomplete Linked Data Streams. We will target continuous queries executed over such data. In addition, in these provenance polynomials, we will include information on the recovery process.

The main challenge of this task relates to dealing with the velocity of the input data. The provenance polynomial has to satisfy two requirements: i) It has to be computed efficiently in a continuous fashion along with the execution of the query, and ii) it has to show how the query execution process evolves over time.

Figure 1 illustrates a system that consists of a knowledge base with data on drivers and companies they work for (*KB=(driver1, worksFor, company1),(driver2, worksFor, company1),(driver3, worksFor, company2)*). Each driver is equipped with a device streaming his position (triples $v_n = (driver_n, position, location_i)$). $S|in|_n$ denotes an input stream for the driver n. A triple in the stream has its provenance: device id, physical state of the device, configuration parameters, etc. The provenance of the triple is grouped under a provenance annotation g_j.

We want to notify two truck drivers working for the same company when they are within a given distance, e.g., 1km. We also want to be able to trace back how the notification was generated. To detect spatial locations we use a query $T_n = (?driver_n; detectedAt; ?location_i)$. Let Γ_n be the result of such query for the n^{th} driver. $\bowtie$ denotes a natural join, to obtain information on the proximity of the two drivers.

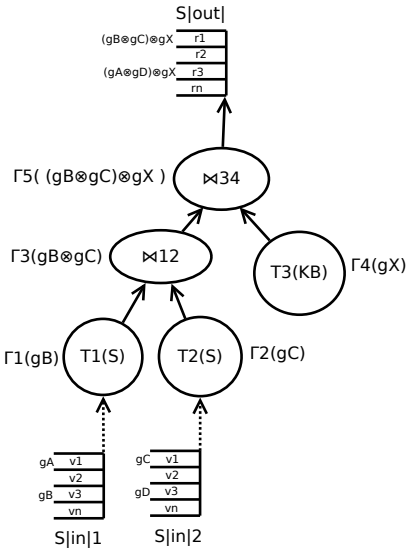

Fig. 1. A provenance polynomial is computed dynamically along with the query execution. At each stage of the data propagation we have information on how the system derived the current state.

Online provenance-aware recovery of incomplete Linked Data Streams. The goal of this task is to develop techniques that will allow us to trace detailed information on the recovery process over incomplete Linked Data streams. The core challenges related to this task are: i) how to detect the incompleteness of the input Linked Data streams, ii) how to handle the dynamics of the input Linked Data streams, and iii) how to devise highly efficient serialization strategies for continuous provenance information. Furthermore, in a heterogeneous integrated environment, the information on the recovery process has to be exchanged between multiple participants to support the accurate derivation of the final provenance.

Fractional provenance of incomplete Linked Data streams. The goal of this task is to determine the fractional provenance of recovered pieces of data, i.e., computing probabilistic information about pieces of data which have contributed to the recovery process and their impact on the recovered piece of data.

The main challenges of this task are dealing with the velocity of the data, as well as, the need to exchange information on fractional provenance between different participants in a Linked Data streams environment. The computation of fractional provenance has to be accurate and efficient. In addition, the information on fractional provenance has to be exchanged in a compact and efficient way.

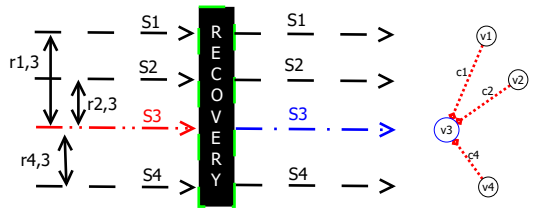

Fig. 2. The fractional provenance is derived from a computed moving streams correlation. The computations are optimized with cumulative moving average to mitigate the velocity of the streams.

Figure 2 shows an example of the process deriving the fractional provenance of the recovered element $v3$. The stream S_3 contains the recovered element, we compute its correlation with the streams S_1, S_2, and S_4 respectively $r_{1,3}, r_{3,3}$, and $r_{4,3}$. Investigating the recovery process is out of the scope of this work. After the recovery, we compute the fractional provenance of the recovered piece of data v_3, i.e., a ratio that defines how much each of the streams S_1, S_2, and S_4 contributed to the recovered element. The exact triples used in the recovery are v_1, v_2, and v_4 with the respective fractional contribution of c_1, c_2, and c_4.

7 Evaluation plan

We will evaluate our approaches in three aspects, 1) extra costs for provenance tracking and tracing, 2) adaptability to changing sources[3] of Linked Data Streams, 3) The accuracy of our query and recovery provenance tracing mechanisms. We will use open Linked Data collections and open collections of time series that contain heterogeneous unstructured data. We will integrate these data collections to our system as static knowledge and stream data. We will also design a set of workload queries using static and dynamic data. The collections of integrated Linked Data and time series data we will employ in our experiments are:

[3] Data sources in distributed systems are very often unstable, which embodies in two aspects: 1) some of the sources can feed data for a period of time, afterwards, disappear completely (e.g., because of some unexpected abnormalities), however, maybe reconnect after a resetting. 2) different sources may have different data generation speed.

- DBpedia[4] is a crowdsourced community effort to extract structured information from Wikipedia and make this information available on the Web. The dataset consists of 6.9 billion RDF triples extracted from Wikipedia. These datasets will be used to evaluate our provenance computation over incomplete Linked Data Streams techniques.
- The Web Data Commons project[5] extracts structured data from the Common Crawl, the largest web corpus available to the public. The dataset contains more than 20 billion RDF triples extracted from the Web. These datasets will be used to evaluate the scalability of our techniques with big data collections of different structure.
- LinkedSensorData[6] is an RDF dataset containing expressive descriptions of 20,000 weather stations in the United States. The dataset contains nearly 2 billion RDF triples describing 160 million observations. These datasets will be used to evaluate the scalability of our methods using real-world sensor data.
- The Swiss Federal Office for the Environment (FOEN)[7] offers access to streams of time series that describe weather phenomena (temperature, air pressure, humidity, precipitation, etc.). The time series contain from 200'000 up to few millions of values each. These datasets will be useful for the evaluation of the compression scalability with the length of time series.

8 Reflections

Large distributed systems with billions of connected devices generate enormous amounts of values every minute. These values are produced in multiple ways for a specific scenario, under different circumstances, with different reputation, etc. This heterogeneous environment requires assessments of quality, reliability, and trustworthiness. With the Linked Data approaches dynamic data can be combined with static knowledge to enable complex data analysis. This combination, however, introduces incomplete and possibly erroneous data to a knowledge base. The number of elements in the physical infrastructure, failures and uncertainty cannot be avoided, such as sensor failures, power outages, sensor to central server transmission problems, etc. In such scenarios we need additional provenance information describing the data involved in the recovery of the missing elements and its fractional contribution. This process increases the overall requirement of transparency from simple physical device information to algorithmic methods, pieces of data involved in producing a value, the probabilistic contribution of correlated elements to the recovery, i.e., the fractional provenance.

To the best of our knowledge, our research is one of the pioneers in the area of handling provenance over incomplete Linked Data Streams.

[4] http://wiki.dbpedia.org
[5] http://webdatacommons.org
[6] http://wiki.knoesis.org/index.php/LinkedSensorData
[7] https://www.bafu.admin.ch/bafu/en/home.html

Acknowledgements

I would like to express my deep gratitude to my supervisors Prof. Dr. Manfred Hauswirth and Dr. Marcin Wylot for their patient guidance, constructive suggestions of this research proposal.

References

1. Heath, T., Bizer, C.: Linked Data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology, vol. 1, no. 1, pp. 1–136 (2011)
2. Le-Phuoc, D., Parreira, J.X., Hauswirth, M.: Linked stream data processing (2012)
3. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., et al.: The Open Provenance Model core specification (v1. 1). Tech. Rep. 6 (2011)
4. Moreau, L., Missier, P.: PROV-DM: The prov data model, W3C recommendation (30 April 2013), http://www.w3.org/TR/prov-dm/
5. Glavic, B.: Perm: Efficient Provenance Support for Relational Databases. Ph.D. thesis, University of Zurich (2010)
6. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing Linked Datasets with the VoID Vocabulary (03 March 2011), http://www.w3.org/TR/void/
7. Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax (25 February 2014), http://www.w3.org/TR/rdf11-concepts/
8. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings (2007)
9. Theoharis, Y., Fundulaki, I., Karvounarakis, G., Christophides, V.: On provenance of queries on semantic web data. IEEE Internet Computing 15(1), 31–39 (Jan 2011)
10. Hartig, O.: Querying trust in rdf data with tsparql. The Semantic Web: Research and Applications pp. 5–20 (2009)
11. Zimmermann, A., Lopes, N., Polleres, A., Straccia, U.: A general framework for representing, reasoning and querying with annotated semantic web data. Web Semant. 11, 72–95 (Mar 2012)
12. Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated rdf. ACM Trans. Comput. Logic 11(2), 10:1–10:41 (Jan 2010), http://doi.acm.org/10.1145/1656242.1656245
13. Nguyen, V., Bodenreider, O., Sheth, A.: Don't like rdf reification?: making statements about statements using singleton property. In: Proceedings of the 23rd international conference on World wide web. pp. 759–770. ACM (2014)
14. Wylot, M., Cudre-Mauroux, P., Groth, P.: TripleProv: Efficient processing of lineage queries in a native RDF store. In: Proceedings of the 23rd international conference on World wide web. pp. 455–466. ACM (2014)
15. Wylot, M., Cudre-Mauroux, P., Groth, P.: Executing provenance-enabled queries over web data. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1275–1285. ACM (2015)
16. Glavic, B., Alonso, G.: The Perm Provenance Management System in Action. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. pp. 1055–1058. ACM (2009)
17. Arab, B., Gawlick, D., Radhakrishnan, V., Guo, H., Glavic, B.: A generic provenance middleware for queries, updates. In: 6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014). USENIX Association, Cologne (Jun 2014)