Automatic Metadata Annotation through Reconstructing Provenance

Paul Groth[1], Yolanda Gil[2], and Sara Magliacane[1]

[1] VU University Amsterdam
[2] Information Sciences Institute, University of Southern California

Abstract. Annotating datasets with metadata is an important part of organizing and curating data. However, it is a time consuming process and often not done in a rigorous fashion. In this paper, we propose a new approach to annotating datasets through the use of reconstructed provenance. A detailed survey of the related work in this area is given. Additionally, we provide an overview of our approach for both reconstructing provenance and using that provenance to automatically annotate datasets with metadata. This approach leverages existing work in AI planning and change detection algorithms.

Keywords: provenance, reconstruction, metadata annotation

1 Introduction

A major impediment to data aggregation and exploitation is the need to describe datasets and their contents with appropriate metadata, so they can be appropriately organized and prepared for analysis. Typically, simple metadata about location and time of collection are available, but important metadata about data properties and provenance require effort and are not typically captured. Moreover, scientists tend to rely on spreadsheets and other data preparation software that does not create metadata for the resulting data. Despite major investments in infrastructure for metadata annotation, the collection of metadata remains a challenging area for science because of the effort it requires.

We are investigating a new approach that automatically derives metadata rather than requiring scientists to provide it. The key idea is that rather than manually annotating the metadata of many datasets, we manually annotate the (much fewer) models that use the data. Scientists will be able to upload datasets they have collected together with informal descriptions, but with no structured metadata associated with them. Other scientists will download these datasets and prepare them to be analyzed by models implemented in software. Our system will have access to the original datasets and the prepared datasets that are input to models.

Our approach is to reconstruct the provenance of the prepared dataset, that is, to infer what sequence of transformations could have been done to the original dataset to obtain the final dataset. The final datasets can be assigned metadata because of the way they are used in a model, and once the provenance is reconstructed then the metadata can be propagated to the initial dataset. We assume a messy environment where data is provided as is (e.g., a normal desktop file system).

Being able to reconstruct provenance is of interest because it places less of a burden on scientists to either adapt to an underlying provenance system or document provenance themselves. In this paper, we provide a review of related work in possible approaches to addressing the problem of reconstructing provenance. From this review, we outline a new approach to solving the problem of reconstructing provenance tailored towards automatic metadata annotation.

2 Approaches to reconstructing provenance

Provenance has been studied from a variety of perspectives. There have been several good surveys of the provenance literature [20, 9, 13]. Here, we focus on the specific literature related to reconstructing provenance. We begin by looking at work directly from the provenance community. We then recast the problem of reconstructing provenance as one of either change detection or planning and review the related work in those two areas.

2.1 Approaches from the provenance literature

We classify the related work in provenance into three broad areas: mining provenance from data, using network structures to infer provenance, and leveraging the execution environment to reconstruct provenance.

Mining provenance The problem of reconstructing chains of historical evolution for a corpus of text documents is discussed in [11]. The approach consists in clustering the documents based on their cosine similarity as vectors of terms and ordering the documents in each cluster based on their creation time. Due to the similarity metrics involved, this method can only reconstruct the dependencies between the documents, while ignoring the transformations that lead from one document to another.

In [24], provenance is interpreted simply as the type of process that created the data. In the considered application domain, i.e. reservoir engineering, the same process often generates instances of semantically related concepts. Assuming access to historical data with complete provenance information, it becomes possible to compute confidence values for semantic associations between concepts that have the same generating process. Given an instance of a concept, its missing provenance can be predicted using the semantic association with the highest confidence value and assigning the provenance of associated items. This work overlaps with workflow mining [1], where workflows are mined from log files.

In the computational workflow environment, [5] describes an approach for inferring service substitutions using examples found in provenance traces, essentially, mining a high-level provenance description.

Leveraging network structures Other work (e.g., [17, 3]) has proposed to reconstruct the provenance of information based on the topology of the underlying network. In this case provenance is intended as a provenance path, i.e. the set of nodes and edges through which the information is communicated. Specifically, in [17], some simple techniques for the reconstruction of incomplete provenance in an information sharing network are given. Provenance is represented as a list of signed metadata from the nodes that have received a specific information item. These metadata include the node identifier, the location and time at which the node processed the item. In case of partial metadata for one node, the missing parts can be approximated based on the metadata from the neighboring nodes. On the other hand, if the provenance chain is incomplete, the path of the information can be reconstructed by first listing all possible paths (either constructing a reachability set or by previously profiling the system) and then matching a path that is most compatible to the known provenance, both by total length and order of common subsequences.

The problem of tracking the information provenance path in a social media setting is defined in [3]. The paper describes how to leverage the structure of a social network present to estimate the most likely provenance paths for a given piece of information. The notion of provenance in this setting is limited to a list of transmitting nodes, without differentiating between the operations that could be performed on these nodes.

Leveraging the execution environment The following approaches rely on knowledge about the execution environment to infer or rebuild provenance information. Work in the database community, has defined the notion of a registry of weak inverse functions that allow transformation allow the inverse of functions applied within a database to be approximated in order to track back provenance [23]. These approximation functions must be registered by users of the system.

In the context of stream data processing, complete provenance information can be very large. In order to reduce the required storage, [19] proposes to store only coarse-grained provenance. Through coarse-grained information about the transformations performed on data and a temporal data model they introduce an algorithm to reconstruct the processing window data and compute fine-grained provenance (tuple-level).

[16] discusses the need for provenance systems that are able to detect and correct errors in provenance records. The authors consider several examples in which the provenance information is incomplete, missing or erroneous, either because of rogue users or failing processes, and conclude that provenance systems should include redundancy (e.g. having several copies of the same record in different nodes) and tamperproof mechanisms to minimize these issues.

Several systems have gathered provenance information about provenance transparently by monitoring application at the operating system level [18, 14]. Based on knowledge about how processes run and reads and writes to the file system occur, these system can reconstruct provenance information.

2.2 Reconstructing provenance as change detection

There exists extensive research on reconstructing sequences of operations based on input and output data, in particular in change detection and edit distance algorithms. These approaches can be seen as analogous to the problem of reconstructing provenance. We give a brief overview of this work here.

Edit distance is a common similarity measure between two entities that consists of the number of transformations required to transform one entity into another. Algorithms for computing the edit distance can also output the related sequence of operations, called edit script. This edit script can be seen as corresponding to some approximate form of provenance.

In the literature, there are several well-known algorithms for computing the edit distance for different types of entities, for example strings, ordered and unordered trees [6] and graphs [15]. In these cases, usually the set of considered transformation consists in elementary operations, e.g. insert/delete node, substitute node, etc. In general, computing the minimal edit distance for unordered trees and graphs has been proved to be an NP-hard problem, although polynomial algorithms have been devised for some special cases of restricted graph structures. Other possible approaches consider using heuristics or approximating the minimal edit distance.

Leveraging domain knowledge, it becomes possible to define more efficient heuristic solutions, e.g. for hierarchically structured data with node insert, node delete, node update as well as subtree move and copy operations [8, 7]. Other edit distance algorithms have been tailored for specific types of data with the corresponding operations. [10] introduce an algorithm for edit distance in ordered XML documents. [12] proposes three different similarity metrics for Business Process Models: text similarity, structural similarity and behavioral similarity. Bao et al. [2] compare provenance traces of several executions of the same workflow. Provenance traces are series-parallel graphs with well-nested forking and looping and the set of considered edit operations (path insertion, deletion, expansion, contraction) is different from the standard tree edit distance problem, thus it is possible to define efficient polynomial time algorithms. PROMPTDIFF is a tool for differentiating ontologies that allows for the detection of high-level changes, which provide richer semantics than change primitives just discussed [21]. In particular, the tool first reconstructs the basic change operations using a set of heuristic matchers and then applies a set of rules to infer complex change operations.

We note that all of the above-mentioned approaches for change detection refer to entities of the same type and optimizations are possible because of deeper knowledge about the domain.

2.3 Reconstructing provenance as planning

Another related field is the planning of composite operations from several atomic operations based on user requirements about the output operation (i.e., AI planning).

A particular instance of this problem is automated web service composition, i.e. the problem of creating plans composing several web services automatically based on user requirements, possibly taking into account also the availability of the web services and the quality of service at run-time. This work is similar to reconstructing provenance as the data involved (i.e. service descriptions) tend to involve complex representations that need to be connected by a set of complex operations. However, unlike provenance, service descriptions are generally of one format.

The general assumptions in this work is that there exists a repository of web services and that a formal description of each web service is available, as well as the formalization of user requirements. In most approaches the composition is divided in two phases: synthesis, which aims at creating a plan of abstract services, and orchestration, which substitutes the abstract services with one of the possibly many functionally equivalent concrete services. Several surveys (e.g. [22, 4]) describe a number of methods that have been proposed for this problem, they can be categorized into workflow composition and AI planning.

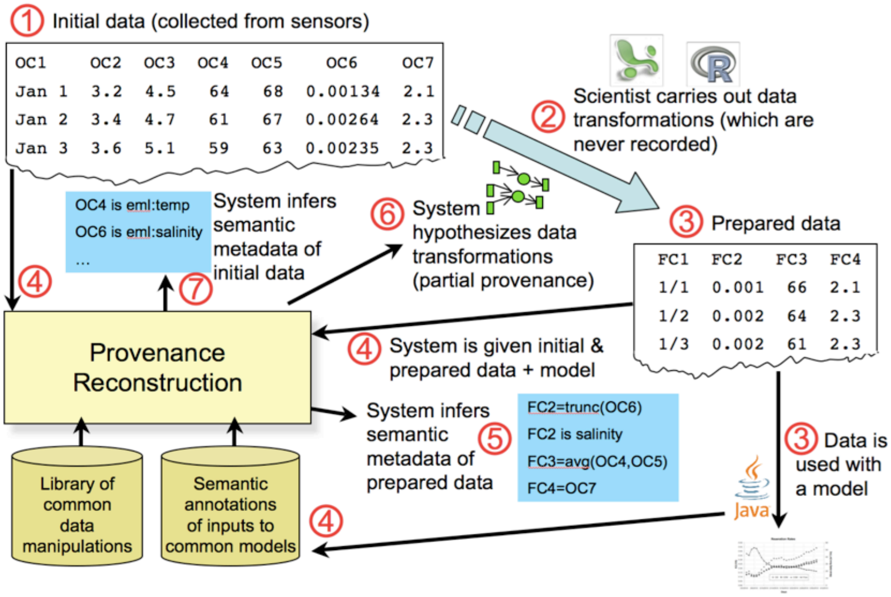
In workflow composition approaches a composite service can be seen as a workflow of atomic services, thus dynamic workflow methods for binding the abstract workflow plan to concrete resources can be reused. However, often these methods require a predefined abstract plan with the set of tasks and in most cases they are limited to serial and parallel composition of tasks. In AI planning approaches formal descriptions of the preconditions and effects of each service are provided. From these descriptions, a plan can be generated automatically by a logical theorem prover or an AI planner.

One of the challenges in these approaches is that they need to handle non-determinism and partially observable states, as well as considering fault-tolerance, quality of service and interactivity with the user during the planning phase. These issues are not present when reconstructing provenance. Furthermore, unlike our domain these approaches require formal descriptions of data.

3 A new approach to reconstructing provenance

Our approach builds upon the above work to develop a new approach to reconstructing provenance that is less dependent on formal descriptions or extensive domain knowledge. Figure 1 illustrates our approach with an example.

The user prepares the data, typically with Excel or a programming tool such as R, but those steps are not recorded. The prepared data is used as input to a model (for example, the Owens-Gibbs model for estimating reaeration rates), which the system knows takes as input date, salinity, average temperature, and CO2 levels in that order. From that, the system infers the metadata for the prepared dataset that was used as input, so FC1 is date, FC2 is salinity, FC3 is temperature, and so on. Now the system searches for transformations that could have been used to transform the initial data into the prepared data, and hypothesizes that column FC2 was derived from truncating the values in OC6, column FC3 from averaging each entry in OC4 and OC5, and column FC4 from OC7.



OC1	OC2	OC3	OC4	OC5	OC6	OC7
Jan 1	3.2	4.5	64	68	0.00134	2.1
Jan 2	3.4	4.7	61	67	0.00264	2.3
Jan 3	3.6	5.1	59	63	0.00235	2.3

FC1	FC2	FC3	FC4
1/1	0.001	66	2.1
1/2	0.002	64	2.3
1/3	0.002	61	2.3

1 Initial data (collected from sensors)
2 Scientist carries out data transformations (which are never recorded)
3 Prepared data
OC4 is eml:temp
OC6 is eml:salinity
...
System infers semantic metadata of initial data
7
4
6 System hypothesizes data transformations (partial provenance)
Provenance Reconstruction
4 System is given initial & prepared data + model
System infers semantic metadata of prepared data
5
FC2=trunc(OC6)
FC2 is salinity
FC3=avg(OC4,OC5)
FC4=OC7
3 Data is used with a model
Java
Library of common data manipulations
Semantic annotations of inputs to common models
4

Fig. 1. An example illustrating the reconstructed provenance of a dataset, and the semantics that were assigned to it because of the model that was used to analyze it.

The system may not be able to figure out that FC1 was derived from OC1. These hypothesized transformations constitute the (possibly partially) reconstructed provenance, which the system then uses to infer semantic metadata for the initial dataset by propagating the metadata of the prepared dataset through the reconstructed provenance.

A major technical challenge for this to work is that there is a very large search space of possible data transformations. Another challenge is that the transformations that users can make to a dataset are not enumerable in principle so the search space is unbound. To address these challenges, there are three key features of our approach.

First, we use anytime algorithms for our search. This means that at any time after they start running, the system will be able to output a partial understanding of how the data was transformed. For example, it may have figured out in a few minutes what six of the nine columns are, it may figure out two more columns in an hour, but it may never be able to figure out what two other columns are because the transformations were not defined in the systems library.

Second, we use systematic search to explore the space of transformations in a principled manner. This means that the system detects when the same partial set of transformations were reached in two different areas of the search space, and only spend time once to further explore them. We use heuristics to guide the search to explore the most promising partial transformation at any given time.

Third, we are developing a library of basic transformations that are common across scientific domains. These include basic mathematical functions, spatial and temporal data transformations, and string transformations (truncation, prefix additions, etc.)

We have developed a prototype of this system to demonstrate the approach. It uses the A* search algorithm combined with a heuristic function based on edit distance to infer the provenance as a sequence of transformations on the original dataset. This search algorithm is heuristic and expands the most promising partial sequence of transformations at each search iteration. Essentially, it combines the approach of AI planning with similarity measures to try to come up with a reasonable approximation of the provenance of a given dataset. The current prototype supports only a small number of structural transformations on tabular data (e.g. CSVs) but we are currently incorporating more.

A key element of the prototype is that we are able to identify particular cells within the output data that can be traced back to cells in the input data. We are currently implementing an approach to back propagate metadata about the output data to the input data using this reconstructed provenance trace.

4 Conclusion

In this paper, we have provided an overview of related work in reconstructing provenance. Based on this overview, we have outlined a new approach to reconstructing provenance that combines AI planning and change detection techniques. While our current work is research in progress, it offers the community a novel frame to think about the problem of reconstructing provenance.

Importantly, reconstructing provenance provides a new solution to the problem of metadata annotation in science. The approach requires no effort from the scientist and would provide a number of benefits: 1) provenance would be automatically reconstructed for tools that do not track it and are ubiquitous in sciences such as Excel; 2) metadata would be automatically annotated including the original data used in an analysis; 3) the reconstructed provenance could used to automatically prepare new data from the same initial sources (e.g. sensors).

Acknowledgements This publication was supported by the Dutch national program COMMIT.

References

1. van der Aalst, W., van Dongen, B.F.and Herbst, J., Maruster, L., Schimm, G., Weijters, A.: Workflow mining: A survey of issues and approaches. Data & Knowledge Engineering Vol. 47, No. 2. pp. 237–267 (2003)
2. Bao, Z., Davidson, S.B., Cohen-Boulakia, S., Eyal, A., Khanna, S.: Differencing Provenance in Scientific Workflows. In: Proceedings of ICDE 2009. pp. 808–819 (2009)

3. Barbier, G., Liu, H.: Information provenance in social media. In: SBP 2011. pp. 276–283. Springer (2011)
4. Baryannis, G., Plexousakis, D.: Automated Web Service Composition: State of the Art and Research Challenges. Tech. rep., 409, ICS-FORTH (2010)
5. Belhajjame, K., Goble, C., Soiland-Reyes, S., De Roure, D.: Fostering Scientific Workflow Preservation Through Discovery of Substitute Services. In: eScience 2011
6. Bille, P.: A survey on tree edit distance and related problems. Theoretical Computer Science 337(1-3), 217–239 (2005)
7. Chawathe, S., Garcia-Molina, H.: Meaningful change detection in structured data. In: ACM SIGMOD Record. pp. 26–37 (1997)
8. Chawathe, S., Rajaraman, A., Garcia-Molina, H.: Change detection in hierarchically structured information. ACM SIGMOD (1996)
9. Cheney, J., Chiticariu, L., Tan, W.C.: Provenance in databases: Why, how, and where. Found. Trends databases 1, 379–474 (April 2009)
10. Cobena, G., Abiteboul, S., Marian, A.: Detecting changes in XML documents. In: Proceedings of ICDE. pp. 41–52 (2002)
11. Deolalikar, V., Laffitte, H.: Provenance as data mining: combining file system metadata with content analysis. In: First workshop on Theory and practice of provenance. p. 10. USENIX Association (2009)
12. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (Apr 2011)
13. Freire, J., Koop, D., Santos, E., Silva, C.T.: Provenance for computational tasks: A survey. Computing in Science and Engg. 10, 11–21 (May 2008)
14. Frew, J., Metzger, D., Slaughter, P.: Automatic capture and reconstruction of computational provenance. Concurrency and Computation: Practice and Experience 20(5), 485–496 (2008)
15. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Analysis and Applications 13(1), 113–129 (Jan 2009)
16. Gates, C., Bishop, M.: One of These Records Is Not Like the Others. Proceedings of workshop on Theory and practice of provenance (2011)
17. Govindan, K., Wang, X., Khan, M., Dogan, G., Zeng, K., Davis, C.: PRONET : Network Trust Assessment Based on Incomplete Provenance. IEEE The Premier International Conference for Military Communications (2011)
18. Holland, D.A., Seltzer, M.I., Braun, U., Muniswamy-Reddy, K.K.: Passing the provenance challenge. Concurrency and Computation: Practice and Experience 20(5), 531–540 (2008)
19. Huq, M., Wombacher, A.: Inferring fine-grained data provenance in stream data processing: reduced storage cost, high accuracy. Database and Expert Systems pp. 118–127 (2011)
20. Moreau, L.: The foundations for provenance on the web. Found. Trends Web Sci. 2, 99–241 (February 2010)
21. Noy, N., Kunnatur, S., Klein, M., Musen, M.: Tracking changes during ontology evolution. In: ISWC. pp. 259–273. Springer (2004)
22. Rao, J., Su, X.: A survey of automated web service composition methods. Semantic Web Services and Web Process Composition pp. 43–54 (2005)
23. Woodruff, A., Stonebraker, M.: Supporting fine-grained data lineage in a database visualization environment. In: Proceedings of ICDE. pp. 91–102 (1997)
24. Zhao, J., Gomadam, K., Prasanna, V.: Predicting Missing Provenance using Semantic Associations in Reservoir Engineering. In: Fifth IEEE International Conference on Semantic Computing (ICSC). pp. 141–148. IEEE (2011)