Data Quality Assessment From Provenance Graphs*

Trung Dong Huynh[1], Mark Ebden[2], Sarvapali Ramchurn[1], Stephen Roberts[2], and Luc Moreau[1]

[1] Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom
tdh@ecs.soton.ac.uk
[2] Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom

Abstract. Provenance is a domain-independent means to represent what happened in an application, which can help verify data and infer data quality. Provenance patterns can manifest real-world phenomena such as a significant interest in a piece of content, providing an indication of its quality, or even issues such as undesirable interactions within a group of contributors. This paper presents an application-independent methodology for analyzing data based on the network metrics of provenance graphs to learn about such patterns and to relate them to data quality in an automated manner. Validating this method on the provenance records of CollabMap, an online crowdsourcing mapping application, we demonstrated an accuracy level of over 95% for the trust classification of data generated by the crowd therein.

1 Introduction

Provenance, a description of what influenced an artifact, has become an important topic in several research communities, since it offers the means to verify data products, to infer their quality, to analyze the processes that led to them, and, importantly, to decide whether they can be trusted [2]. Since provenance records 'link' an artifact with its influences, they can be represented in a graph whose nodes represent the artifact and the influences and whose edges their relations with one another, called a provenance graph. Studying such graphs, e.g. visualizing them, can help to facilitate understanding of the provenance records they contain. However, as any graphs, a provenance graph can be very large; this makes it difficult to follow and interpret its information in a manual manner (in the graph representation or otherwise). An automated and principled way to analyze and understand this (relatively) new type of information for such applications is much needed.

* This work was funded by the Engineering and Physical Sciences Research Council (EPSRC) as part of project 'Orchid', grant EP/I011587/1.

Against this background, we propose to combine graph and data mining techniques in order to relate information in provenance graphs to real-word characteristics of the data they describe. In particular, we adapt a number of network metrics to suit provenance graphs and define provenance-specific ones to summarize the topological structure of provenance graphs [1]. The *provenance network metrics* of a provenance graph can then be used as inputs to construct predictive models to gain useful knowledge about its data, such as their quality or importance. Due to the generic nature of the data model, our method is application-independent and can be applied on provenance graphs from a wide range of applications. In this paper, we demonstrate the method on real-world data from CollabMap — an online crowdsourced mapping application [3] and show that it achieves a 95% accuracy in predicting the trustworthiness of CollabMap data.

2 Methodology

We adopted the PROV Data Model, developed by the W3C Provenance Working group, as the data model for provenance in our analyses. Our approach applies supervised machine learning techniques over the provenance networks metrics to build predictive models for data quality. In more detail, it aims to build a quality predictor that classifies data entities from a given data set into quality labels based on the network metrics of their provenance graphs. The approach can be summarized into three phases as follows:

1. **Design**: Define the problem and methods
   (a) Define the input provenance graph: As a provenance graph can record provenance of multiple entities, the remit of an entity's provenance graph needs to be specified in a consistent manner over the given data set.
   (b) Define the data quality: A concrete quality metric $Q(a)$ needs to be defined in the context of the application's domain.
   (c) Check applicability: As this method relies on supervised learning techniques, a curated set of labeled training data is required (i.e. $Q(a)$ is known for all a in the set). The size of the curated set needs to be sufficient to train the chosen predictive model in the next phase.
2. **Training**: Construct a predictive model for Q from the training set
   (a) Determine a learning algorithm that suits Q and the given data set.
   (b) Calculate the network metrics for the provenance graphs of the labeled data and transform them into feature vectors suitable as inputs to the chosen learning algorithm.
   (c) Divide the curated labeled data into a training set and a test set.
   (d) Run the chosen algorithm on the training set and evaluate the accuracy of the resulted predictive model on the test set. If the accuracy is high, proceed to the Prediction phase. Otherwise, tune the learning algorithm or select another one and repeat this step.
3. **Prediction**: Use the predictive model from the Training phase to predict $Q(a)$ for unseen entities a from their provenance network metrics.

3 Trust Classification for CollabMap

CollabMap is a crowd-sourcing platform for constructing evacuation maps for urban areas. These maps need to contain evacuation routes connecting building exits to the road network, while avoiding physical obstacles such as walls or fences, which existing maps do not provide. The application crowd-sources the drawing of such evacuation routes from the public by providing them with aerial imagery and ground-level panoramic views. It allows inexperienced users to perform tasks without them needing the expertise to integrate the data into the system. To ensure that individual contributions are correct and complete, the task of identifying routes for a building was broken into different micro-tasks done by different contributors: **building identification** (outline a *building*), **building verification** (vote for the building's validity), **route identification** (draw an evacuation *route*), **route verification** (vote for validity of routes), and **completion verification** (vote for the completion of the current *route set*). This allows individual contributors to rate and correct each other's contributions (see [3] for more details). In this section, we apply our methodology in Section 2 to classify the trustworthiness of crowd-generated data in CollabMap as follows.

3.1 Design phase

A typical provenance graph in CollabMap contains one building, one or more evacuation routes, and one or more route sets. All of these need to be assessed for their trustworthiness. For a data entity a, we define a subgraph of the provenance graph G containing a such that it contains a and all nodes depending on a, called the *dependency graph* of a. Node v_i depends on a when there exists a path from v_i to a in G, denoted as $v_i \rightarrow^\star a$. Hence, the dependency graph of a extracted from G, denoted as $D_{G,a}$, has the vertex set $V_{G,a} = \{v \in V : v \rightarrow^\star a\}$ and edge set $E_{G,a} = \{e \in E : \exists v_s, v_t \in V_{G,a} \cdot e = (v_s, v_t)\}$, where V, E are G's vertex and edge sets. $D_{G,a}$ is used as the input provenance graph.

Buildings, routes, and route sets are verified and voted either positive or negative multiple times by CollabMap participants. We use those votes to define the concrete quality metric $Q(a)$ for each of them:

$$Q(a) = \begin{cases} \text{trusted} & \text{if } \tau(a) \geqslant 0.8 \\ \text{uncertain} & \text{if } \tau(a) < 0.8 \end{cases} \tag{1}$$

where *trusted* and *uncertain* are the trust labels assigned to a according to its trust value $\tau(a)$. $\tau(a)$ is calculated from the numbers of positive (p) and negative (n) votes of a from the beta family of probability density functions: $\tau(a) = \frac{\alpha}{\alpha+\beta}$, where $\alpha = p + 1$ and $\beta = n + 1$.

Over its deployment, CollabMap participants generated 5,175 buildings, 4,911 evacuation routes, and 3,043 route sets. All these have multiple votes each. Therefore, $Q(a)$ is known for every data entity in the three data sets. In addition, the sizes of the three data sets are significant and are apt to proceed to the next phase.

3.2 Training phase

As we need to classify CollabMap data into only two labels (i.e. trusted and uncertain), we chose the simple classification and regression tree algorithm provided by the Scikit-learn library as our learning algorithm. For each data entity a from the three data sets, we extract its dependency graph $D_{G,a}$ and calculate the following network metrics on $D_{G,a}$: number of nodes, number of edges, diameter, and nine maximum finite distances (MFD) [1]. These serve as the feature vector to train a decision tree classifier for each data set. The available data are divided into training sets and test sets as shown in Table 1.

Table 1. Sample sizes of training sets and test sets.

Data Type	Category:	Trusted	Uncertain
Building	Training set	939	939
	Test set	2357	940
Route	Training set	1088	1088
	Test set	1646	1089
Route Set	Training set	648	648
	Test set	649	1098

We train the decision tree classifiers on the training sets and test them on the test sets for buildings, routes, and route sets. The performance of the classifiers is presented in Table 2 and is summarized by three common statistical measures for the performance of a binary classification test: sensitivity, specificity, and accuracy. The results demonstrated that the trained classifiers could predict the trust labels for all the three test sets with a high level of accuracy: more than 95%. Given such a high accuracy level, the classifiers are deemed suitable to be used in the Prediction phase, which we intend to deploy in our future work.

Table 2. Performance of the trust classification

	Sensitivity	Specificity	Accuracy
Building	96.61%	99.17%	97.00%
Route	94.78%	97.32%	95.28%
Route Set	97.23%	97.78%	97.77%

References

1. Ebden, M., Huynh, T.D., Moreau, L., Ramchurn, S., Roberts, S.: Network analysis on provenance graphs from a crowdsourcing application. In: Groth, P., Frew, J. (eds.) Provenance and Annotation of Data and Processes, Lecture Notes in Computer Science, vol. 7525, pp. 168–182. Springer Berlin Heidelberg (2012)
2. Moreau, L.: The foundations for provenance on the web. Foundations and Trends in Web Science 2(2–3), 99–241 (Nov 2010)
3. Ramchurn, S.D., Huynh, T.D., Venanzi, M., Shi, B.: Collabmap: Crowdsourcing maps for emergency planning. In: 5th ACM Web Science Conference (WebSci '13) (2013)