Interpretation of Crowdsourced Activities Using Provenance Network Analysis

T. D. Huynh,[†][*] M. Ebden,[‡] M. Venanzi,[†] S. Ramchurn,[†] S. Roberts,[‡] L. Moreau[†]

[†] Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, United Kingdom

[‡] Department of Engineering Science, University of Oxford
Oxford, OX1 3PJ, United Kingdom

[*] Email: *tdh@ecs.soton.ac.uk*

Abstract

Understanding the dynamics of a crowdsourcing application and controlling the quality of the data it generates is challenging, partly due to the lack of tools to do so. Provenance is a domain-independent means to represent what happened in an application, which can help verify data and infer their quality. It can also reveal the processes that led to a data item and the interactions of contributors with it. 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 provenance graphs, constructed from provenance records, to learn about such patterns and to use them for assessing some key properties of crowdsourced data, such as their 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

Crowdsourcing is an increasingly popular approach for tasks for which algorithmic or computational solutions do not exist readily; the method distributes tasks among human contributors, often across the Web. For instance, citizen-science projects at Zooniverse[1] have managed to enlist hundreds of thousands of volunteer "citizen scientists" to classify distant galaxies, transcribe historical naval logs, and more. Some other crowdsourcing projects proved to be less successful. The effort by a team at the University of California, San Diego to solve the DARPA Shredder Challenge[2] (piecing together roughly 10,000 pieces of documents that have been shredded), for example, was marred by sabotage from a few participants and failed to capitalize from its initial progress (Aron 2011). Moreover, even in the absence of malicious behavior, the quality of crowdsourced data can vary greatly depending on the contributors' background (e.g., country, language) and expertise (e.g., drawing or mathematical skills). Usually cross-verification among participants helps to discard inaccurate results (Bernstein et al. 2010), yet challenges remain in anticipating how different human contributors behave and in designing a robust crowdsourcing application. Crucially, to date no principled approach to understand behaviors in such applications has been proposed. In particular, we lack a generic method to assess some key properties of crowd-generated data (e.g. accuracy, reliability, trustworthiness) that may decide the success of a crowdsourcing project.

In this context, *provenance* is a standard way (Moreau and Missier 2013) to record what happened in an application. It is domain-independent and provides a powerful abstraction of the activities and the data they generated. It offers the means to verify data and to infer their quality, to analyze the processes that led to a thing, and to determine whether it satisfies a set of policies of use or can be trusted (Moreau 2010). For these very purposes, provenance of some form has been used to record how knowledge is created in collaborative environments such as Wikipedia, OpenStreetMap, and CollabMap.[3] In such environments, information and data are continuously and organically generated, revised, extended, and removed by participating contributors. Provenance therein, hence, plays the important role of keeping track of the evolution of a piece of knowledge. As such, it can provide insights into how people interacted with a data item, reflecting their interests and, in some cases, the interactions among them. A heavily edited section in a Wikipedia article, for instance, could suggest that the content is controversial, that it had been reviewed by many people, or sometimes the animosity between two groups of contributors if they kept removing each other's edits.

Against this background, in this paper we present a novel application-independent methodology to assess properties of crowd-generated data from analyzing provenance records. Specifically, it studies provenance represented in the form of a graph by looking at a number of common network metrics from the topology of a provenance graph — number of nodes, number of edges, graph diameter — in addition to a number of provenance-specific network metrics (see Section 2.1). Machine learning techniques are then employed to explore potential correlations between the above topological metrics and properties of crowdsourced data, allowing us to

[1] www.zooniverse.org

[2] http://archive.darpa.mil/shredderchallenge

[3] See www.wikipedia.org, www.openstreetmap.org, and www.collabmap.org, respectively.

build a predictive model for the property of interest. In order to validate the approach, we applied it to classifying the trustworthiness of data from CollabMap — an online crowdsourced mapping application (Ramchurn et al. 2013). Moreover, we empirically show that the network metrics from its provenance graphs can reliably predict the trust classification of its crowd-generated data (as verified by the application's contributors).

By so doing, this paper advances the state-of-the-art in the following ways. We propose the first method for assessing properties of crowdsourced data based on topological analyses of provenance graphs. We show how provenance graphs can be used to capture the behavior of a crowdsourcing system and its actors; and, more importantly, how they can be analyzed to predict the performance of a crowdsourcing system. The method is application independent since it does not rely on domain-specific information. Finally, we demonstrate the effectiveness of the approach on CollabMap and show that it achieves a 95% accuracy in the prediction of trustworthiness of its data.

The remainder of the paper is structured as follows. We introduce our method of analyzing provenance graphs in Section 2. The CollabMap application is described in Section 3 and the results of applying our method on CollabMap provenance are provided in Section 4. The related work is referred in Section 5 and, finally, Section 6 concludes the paper and outlines the future work.

2 Provenance Graph Analytics

We adopted the PROV Data Model (Moreau and Missier 2013), developed by the W3C Provenance Working group to be a generic data model for the interchange of provenance information between a wide range of heterogenous applications, as the data model for provenance in our analyses. PROV defines provenance as a "record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing". The core PROV concepts[4] are illustrated in Figure 1. In brief, provenance records describe the production and use of *entities* by some *activities*, which may be influenced in some ways by *agents*. They can be represented in the form of a directed graph, in which entities, activities and agents are represented as nodes, and the relations between them (e.g. used, wasGeneratedBy, wasDerivedFrom, etc.) as directed edges (see Figure 2 for an example). The rest of this section introduces the network metrics we employ to quantify the topological characteristics of a provenance graph (Section 2.1) and the analysis of these to assess the quality of crowdsourced data (Section 2.2).

2.1 Provenance Network Metrics

A provenance graph is a directed graph $G = (V, E)$, with vertex set V and edge set E. In order to describe the structural characteristics of a provenance graph, we use the following common network metrics: number of nodes $|V|$, number of edges $|E|$, and graph diameter, which is the longest *distance* in a graph, where the distance between two vertices is the length of the shortest path between them. Since nodes in provenance graphs are separated by directed edges, thereby preventing some nodes from forming a path to certain others, strictly speaking the diameter of each graph is infinite. However, by temporarily assuming the edges are undirected, we are able to calculate the diameter of a provenance graph.

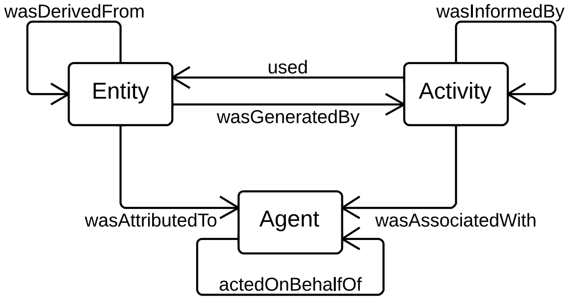

Figure 1: The UML class diagram for PROV core concepts.

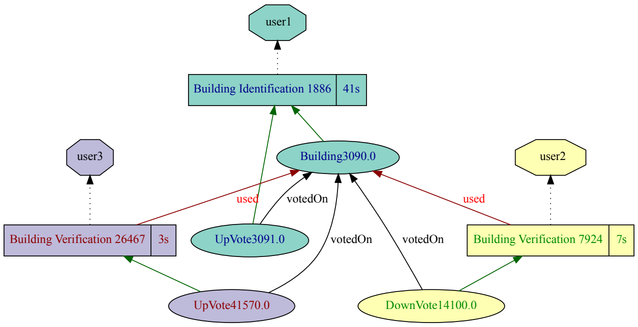

Figure 2: An example provenance graph recorded by CollabMap representing a building was drawn and voted on by three different users (in three different background colors).

In addition to the above standard metrics, we are also interested in a provenance-specific network metric called the *maximum finite distance* (MFD) (Ebden et al. 2012). MFD is a variation of the graph diameter and is defined as the greatest minimum finite distance from one node type to another on a directed graph G (ignoring infinite distances). Since there are three different node types in a provenance graph, there are nine different MFD metrics, one for each pair of node types.[5]

2.2 Analyzing Provenance Graphs

As mentioned earlier, we are interested in assessing the quality of crowdsourced data, which are represented as entities in provenance graphs. As a generic record, a provenance graph

[4] Due to limited space, the complete descriptions of those concepts could not be included here, but the reader is encouraged to refer to (Moreau and Missier 2013) for their formal definitions.

[5] There are several other provenance-specific network metrics introduced in Ebden et al., 2012. However, they characterize the growth of provenance graphs over time and are not directly applicable in this work since we analyse only final provenance graphs, which are static snapshots rather than evolving graphs.

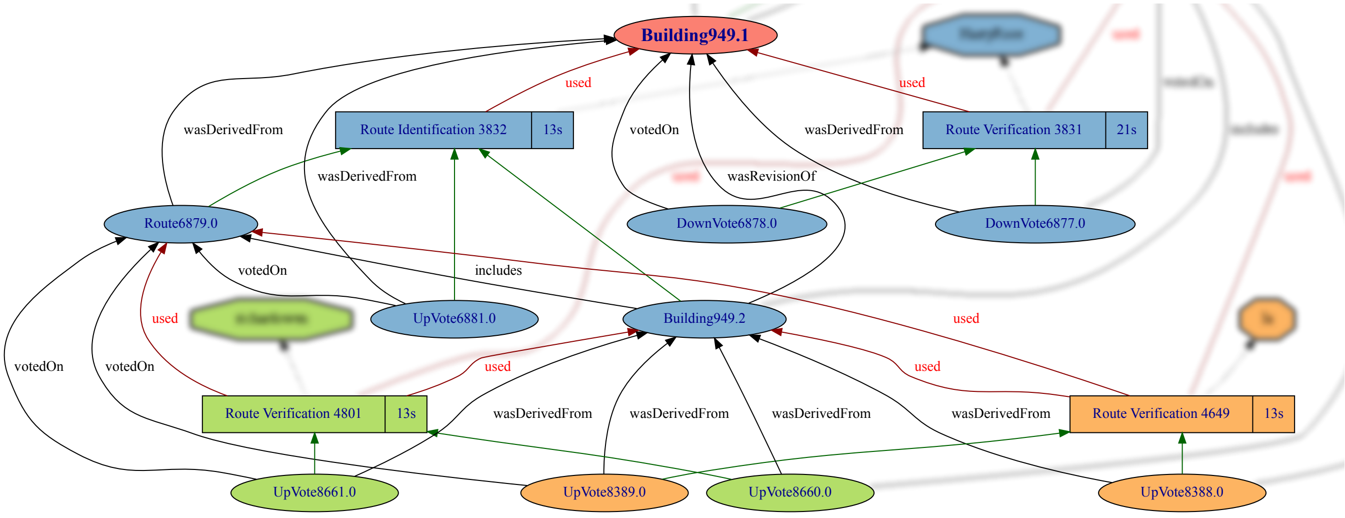

Figure 3: The dependency graph of the node `Building949.1` (top of the graph). The blurred-out nodes and edges belong to the full provenance graph of a task, but are not included in the building's dependency graph.

may or may not carry domain-specific information that is useful for determining the quality of an entity, but it typically describes the relations between the entity and its other nodes. Studying these relations can provide an indication of the entity's value in the graph. A highly cited academic paper, for example, is generally considered of high value thanks to its citations, or in other words, the relations it has with other papers. Such relations show how many times the paper was used in the generation of, or had influence on, the others. The influence of an entity is, thus, the focus of our quality assessment analyses.

Since an edge in a PROV graph represents some form of influence between its source and its target (Moreau and Missier 2013), if there exists a path in a PROV graph from node v_i to node v_0, denoted as $v_i \rightarrow^\star v_0$, then v_i was, in some way, potentially influenced by v_0. In other words, we consider here the transitive (potential) influence of v_0. It is possible to extract from graph G a sub-graph $D_{G,a}$ containing only the nodes that were directly or indirectly influenced by a particular node a, as follows:

$$
\begin{aligned}
D_{G,a} &= (V_{G,a}, E_{G,a}) && (1)\\
V_{G,a} &= \{v \in V : v \rightarrow^\star a\} && (2)\\
E_{G,a} &= \{e \in E : (\exists v_s, v_t \in V_{G,a})\,(e = (v_s, v_t))\} && (3)
\end{aligned}
$$

We call $D_{G,a}$ the *dependency graph* of a extracted from the provenance graph G, or the transitive closure of a's potential influence in G; $V_{G,a}$ and $E_{G,a}$ are its vertex set and edge set, respectively. Hence, it is now possible to analyze the influence of a in G by examining the dependency graph $D_{G,a}$. Figure 3 presents an example of such a dependency graph (shown as extracted from a bigger provenance graph). Our hypothesis is that by studying the dependency graph of a, in many cases, we can gain insights on how a was used, which might reveal properties of a such as its value or quality. Although this approach is domain independent, its usefulness will depend on the amount of activities and information modeled and recorded in a's provenance graph. Therefore, in the context of a specific application, it is needed to determine whether there is indeed a correlation between a's quality and $D_{G,a}$. Our method for this is as follows.

Assuming a quality metric $Q(a)$ for an entity a is defined,[6] a predictive model for $Q(a)$ can be built based on the network metrics of $D_{G,a}$ by applying a suitable (supervised) machine learning technique on a curated set of data for which $Q(a)$ is known (e.g., from expert assessments). If the predictive model is empirically shown to be able to predict $Q(a)$ with high accuracy, this implies that the correlation between $Q(a)$ and the network metrics of $D_{G,a}$ exists. The model then can serve as a quality predictor for other entities from the same application domain.

In order to demonstrate the method, we show how it can be applied on the data generated in the CollabMap application (Section 4), which crowdsourced the identification of evacuation routes for residential buildings. Before so doing, however, we first need to introduce CollabMap.

3 CollabMap

In planning the responses to city-wide disaster scenarios, simulating large-scale evacuation is a major challenge, owing in part to the lack of detailed evacuation maps for residential 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. CollabMap was developed to crowdsource the drawing of evacuation routes from the public by providing them with two freely available sources of information from Google Maps: aerial imagery and ground-

[6] This must be done in a specific application domain since "quality" is a generic term and the quality of data can only be defined within their application's context.

level panoramic views. It allows inexperienced users to perform tasks without them needing the expertise to integrate the data into the system and does not rely on having experts verifying the tasks in order to generate meaningful results.

To ensure that individual contributions are correct and complete, CollabMap extended the Find-Fix-Verify pattern by Bernstein et al. (2010) into an adaptive workflow that includes consensus-based trust metrics. 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 (i.e. buildings, routes, and route sets). They were, however, not allowed to vote for their own contributions to avoid biases (see Ramchurn et al., 2013 for more details).

During all the tasks, the provenance of crowd activities were recorded; i.e., the data entities that were shown to the user in a micro-task and the new entities it generated, along with their inter-relationships (see Figure 2 for an example). After our trials finished (which are summarized in the next section), we extracted the dependency graph for each building, routes, and route sets. The network metrics of those graphs were then used to predict the quality of the corresponding data. The following section provides a brief summary of the two deployments of CollabMap and Section 3.2 explains how the trustworthiness of its data was assessed.

3.1 Deployments

In collaboration with the Emergency Planning Unit at Hampshire County Council, CollabMap was deployed to help map the area around the Fawley Oil refinery (situated near the city of Southampton in the UK). As a benchmark, we also ran the system using Amazon's Mechanical Turk[7](AMT), a popular crowdsourcing platform, to hire its workers from around the world to participate in the application.

Local deployment In the first case, CollabMap was run over the course of three months and generated more than 38,000 micro-tasks from the crowd. The contributors to CollabMap were mainly recruited from the local community around the refinery and staffs and students from our university. They were incentivized by various lottery prizes, one lottery ticket was given for every ten micro-tasks completed. During the trial, the contributors were informed of how many of their buildings or routes had been voted as invalid. It was also made clear that those would not been counted towards their allocations of lottery tickets. Thanks to this mechanism, the proportions of invalid buildings and routes were low, 1.5% of buildings and 0.3% of routes.

AMT deployment In the second deployment, CollabMap micro-tasks were posted to the AMT platform and its workers were paid a set price of $0.02 per every completed micro-task. In this trial, the tasks were still on the same area around the Fawley Oil refinery, but the system was deployed afresh without the data from the previous trial. It ran for just over three hours with 8,000 micro-tasks completed. Due to technical limitations, the payments for completed tasks could not be deferred until the validity of the work was confirmed as in the previous case (with respect to lottery ticket allocations). Some AMT workers appeared to exploit this lack of verification by consistently submitting input that seemingly ignored the content of the images provided. Such data were normally detected and voted down by other participants in the first deployment, but some AMT users also voted up invalid work by others. Given the number of tasks and the relatively short time in which they were completed, we were not able to detect this anomaly immediately. As a result, when we stopped the trial once this was discovered, the proportion of invalid buildings was 21.5%, which resulted in a rather noisy data set. At the same time, this new behavior of the participants manifested in a topological change of the buildings' provenance graphs, as revealed by our analytics method in Section 4.3.

3.2 Estimating the Quality of Data

With the large number of buildings and routes drawn, it was impractical to have them checked by experts, and, hence, CollabMap relied on its contributors to verify each other's work. The validity of buildings, routes and the completion of route sets was ascertained by giving those entities either positive or negative votes. From the votes recorded, following the TRAVOS trust model (Teacy et al. 2006), we defined the trustworthiness of a voted entity based on the beta family of probability density functions as follows:

$$\tau(e) = \frac{\alpha}{\alpha+\beta} \tag{4}$$

$$\alpha = p+1 \tag{5}$$

$$\beta = n+1 \tag{6}$$

where $\tau(e)$ is the trust value for the data entity e being evaluated (which is the mean of the beta distribution defined by the hyper-parameters α and β), p and n are the numbers of positive and negative votes of e, respectively. This trust metric will serve as the basis for estimating the quality of an entity in CollabMap.

4 Analyzing CollabMap Data

In this section, we demonstrate the application of the provenance analysis approach described in Section 2.2 on the crowdsourced data from CollabMap. First, a quality metric is needed to be defined in the application's context. Although we did not have expert assessments, CollabMap buildings, evacuation routes, and route sets were cross-checked and voted by the participants multiple times. Using the trust value specified by Equation 4, $Q(a)$ can be defined for any data entity a in CollabMap as follows:

$$Q(a) = \begin{cases} \text{trusted} & \tau(a) \geqslant 0.8 \\ \text{uncertain} & \tau(a) < 0.8 \end{cases} \tag{7}$$

[7] www.mturk.com

where *trusted* and *uncertain* are the trust labels assigned to the data according to their trust value, 0.8 is the threshold we chose to select highly trusted data in CollabMap. Having defined $Q(a)$, we formulate the correlation hypothesis:

Hypothesis 1 $Q(a)$ correlates with the network metrics of $D_{G,a}$, where G is the provenance graph that contains a.

In order to validate the above hypothesis, we built a model to predict the trust label of entity a from the network metrics of $D_{G,a}$: number of nodes, number of edges, diameter, and nine MFDs (see Section 2.1). These metrics served as the classifying *features* of entity a that the predictive model took as inputs. For this purpose, we used a decision tree classifier by the Scikit-learn Python library (Pedregosa et al. 2011), which is an implementation of C4.5 (Quinlan 1993), an algorithm for classification and regression trees. In the next section, we present the results of checking Hypothesis 1 and testing the predictive model on buildings, routes, and route sets from the first CollabMap deployment. Section 4.2 then verifies the performance of the predictive model on data from the second deployment on the AMT platform and Section 4.3 looks into the AMT data sets on their own.

4.1 Local Deployment

In the first deployment of CollabMap, its contributors generated 5,175 buildings, 4,911 evacuation routes, and 3,043 route sets. The trust values of these were calculated and trust labels assigned accordingly. Each data set (one for each of the three data types above) was divided randomly into two sets: the training set and the test set (see Table 1). The trusted and uncertain data in the training set were selected so that their numbers were equal, avoiding unbalanced training data that might bias the classifier. The features of the entities in the training set and their respective trust labels served as training inputs for the decision tree classifier. The trained classifier was then used to predict the trust labels for the entities in the test set from their feature data. Comparing the predicted labels with the actual trust labels derived from votes allows us to gauge the predictive power of the classifier, which indirectly reflects the correlation between the network metrics and the perceived data quality by the contributors (but only in the case of high predictive power).

Following the method outlined above, we trained the decision tree classifier for and tested it on the buildings, routes, and route sets generated in the first CollabMap deployment. The performance of the classifier 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 buildings, route and route sets in the test sets with a high level of accuracy: more than 95%.[8] Given such a high predictive power of the classifiers, we conclude that there is indeed a strong correlation between the network metrics (of the dependency graphs) and the trust categories of the data generated in CollabMap, validating Hypothesis 1 in all the three cases of buildings, routes, and route sets. This finding is important because it shows that analyzing the network metrics of provenance graphs can be suitable for making sense of the activities and data they describe, such as classifying generated data into trust categories as in this case.

Table 1: Classification data from the local deployment

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

Table 2: Performance of the trust classification of CollabMap data from the local deployment

	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%

One benefit of using a decision tree classifier is that, after having been trained, it is able to explain its decision model for making predictions in the form of a decision tree. An example of such a tree is provided in Figure 4, which shows the resulting tree from training the classifier for trust labels of routes from the route training set above. Its depth, however, was limited to three during the training phase in order to generate a tree that can fit the limited space here. The full decision tree for routes, whose classification performance is shown in Table 2, had 268 nodes in total. It is worth noting that even the simple decision tree with nine nodes shown in Figure 4 can still achieve the relatively high level of classification accuracy of 91%. Although the votes from contributors are subjective and there is no hard-and-fast rule for select trusted/uncertain entities, the decision tree classifier found that most routes with small dependency graphs (i.e. less than 11 nodes) were voted as uncertain. This reflects the fact that those routes were not been 'used' (or depended on) as much as those with bigger dependency graphs, suggesting less trust from the contributors. The classifier then split the data according to the MFD (entity→entity), the length of the longest dependency chain between entities, of their dependency graphs. This suggests that how far newer entities are away from the root of a dependency graph reflects its trustworthiness. Any further interpretation would require a close examination of the actual dependency graph and its data before any conclusion could be made. Nevertheless, this approach provided us with useful hints for further investigations to understand the relationship between the crowd's activities and the data they generated in CollabMap.

In addition to the decision tree, the classifier also revealed which features were the most relevant with respect to its prediction task from the training data, which is presented in Table 3. The most noteworthy result from this table is the

[8] We also found that the accuracy of the classifiers was largely insensitive to the choice of trust threshold.

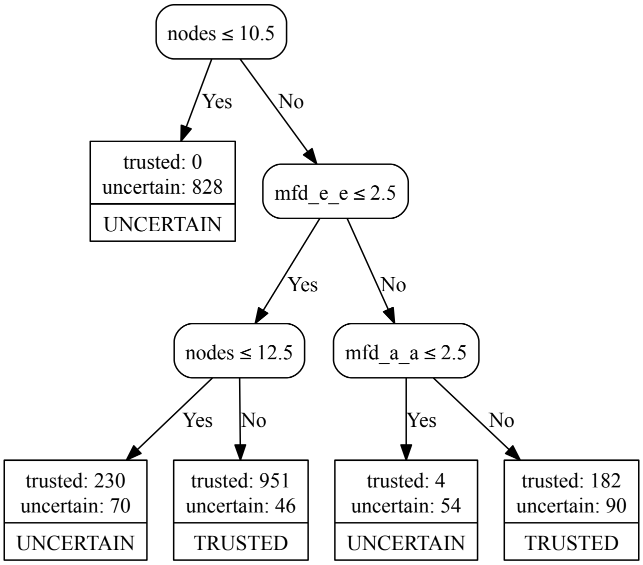

Figure 4: The 3-depth decision tree for predicting trust labels of routes.

contrasting differences between the set of relevant network metrics with respect to the trust classification of buildings, routes, and route sets. The number of edges in dependency graphs appeared to be overwhelmingly important in determining the trust categories of buildings, while the number of nodes was significantly more relevant in the case of routes; and both the metrics are important in classifying route sets alongside the graph diameter. It is also interesting to learn that the MFD metrics[9] did not seem to help the classifier in the case of buildings and route sets, while it was marginally relevant in the case of routes (more so in the decision tree in Figure 4 whose depth was limited to three). Such differences were anticipated given the different natures of how buildings, routes, and route sets were used in CollabMap (and it should be noted that the results here only apply to data from CollabMap). Although the decision tree and the relevance values above do not explicitly account for the connections between the features (i.e. the network metrics) and the prediction categories (i.e. the trust labels), they give useful indications as to what to focus on in further data analysis to help identify such connections.

4.2 Cross-check Validation on AMT Data Sets

Having shown that the classifiers can reliably predict the trust categories of data from the first CollabMap deployment, we used the same trained classifiers above to predict the trust labels of data obtained from the AMT CollabMap deployment, participated by a completely different set of contributors. The test sets in this case contained 808 buildings, 994 routes, and 546 route sets.

Table 3: The relevance of each network metric in predicting the trust labels of buildings, routes, and route sets from the training sets in Table 1 (irrelevant metrics are omitted)

Network Metric	Building	Route	Route Set
number of nodes	0.087	0.704	0.502
number of edges	0.900	0.193	0.190
graph diameter	0.012	0.025	0.308
MFD (entity→entity)	0.001	0.067	-
MFD (entity→activity)	-	0.006	-
MFD (activity→activity)	-	0.005	-

Table 4: Cross-check validation results of trust classification on the AMT data sets

	Sensitivity	Specificity	Accuracy
Building	72.43%	50.19%	77.23%
Route	99.78%	93.08%	96.48%
Route Set	100%	90.53%	95.05%

The results of the second experiment are presented in Table 4, which show that the trained classifiers for routes and route sets still performed well on the AMT test data, with high levels of accuracy: over 95% in both cases. The trained classifier for buildings, however, performed less reliably with the AMT building data: it could predict the correct trust labels for 77.23% of the buildings, compared to 97% of the buildings in the previous deployment. This decrease in classification accuracy most likely stemmed from the anomaly of the AMT deployment, which generated a sizeable proportion of nonsensical building outlines (as detailed in Section 3.1). Questionable buildings were normally voted down in the local deployment and their dependency graphs are typically shallow. In the AMT deployment, a significant number of such buildings were continued to be worked on as some AMT users just wanted to complete as many tasks as possible. The dependency graphs of those buildings, therefore, continued to grow resulting in unexpectedly deeper dependency graphs (as compared with those of invalid buildings from the local deployment). As a results, the classifier trained with data from the local deployment did not work as well as in the first experiment. Therefore, we re-trained the classifiers, this time with AMT data, and test them again in the next section.

4.3 Classifying AMT Data Sets

Given the irregularity in the case of buildings in the second experiment, we repeated what we performed in the first experiment on the data generated in the AMT deployment, in order to check the method with the AMT data sets. Again, the AMT data were split into training and test sets as shown in Table 5. The decision tree classifiers were re-trained and re-tested, whose performance is presented in Table 6.

The classifier performed better with the buildings from the AMT deployment this time, having been re-trained with the data generated by the same contributor population. Its

[9] Since provenance graphs in CollabMap do not have edges starting from agent nodes, MFD (agent→entity), MFD (agent→activity), and MFD (agent→agent) are not define. As a result, they played no role in the classification. This, however, needs not to be the case with other provenance graphs in general.

Table 5: Classification data from the AMT deployment

Data Type	Category:	Trusted	Uncertain
Building	Training set	92	92
	Test set	93	531
Route	Training set	229	229
	Test set	229	307
Route Set	Training set	129	129
	Test set	129	159

Table 6: Performance of trust classification of data from the AMT deployment

	Sensitivity	Specificity	Accuracy
Building	93.55%	53.37%	86.86%
Route	99.56%	94.61%	97.39%
Route Set	100%	96.27%	98.26%

performance with the test routes and routes sets was highly accurate, similarly to the performance in the first experiment. It is worth noting, however, that the specificity of the predictive model for building is significantly low in both Tables 4 and 6. This suggests that the noisy (hence inconsistent) building data from the AMT deployment somewhat confuses the classifier and diminishes the correlation between the network metrics and the trust labels of buildings from this deployment.

In addition, studying the relevance of the network metrics given by the newly trained classifier (shown in Table 7) revealed significant differences with that from the first experiment (Table 3). The most notable difference is the relevance values given for the building classification. The number of edges was the overwhelmingly significant in the first experiment, but in this experiment, the relevance of the number of nodes and the number of edges is almost equal. Although our analytics method was still able to establish the connections between the network metrics and the trust categories of data from the AMT deployment (as demonstrated by the very high levels of classification accuracy), whatever the connections, they were not quite the same as those in the local deployment. This deviation, perhaps, reflects a change in the behavior of the AMT participants compared to that of those from the local deployment (as belatedly discovered by us and noted in Section 3.1). As a result, that change manifested in a different relationship between the quality of buildings and the activities carried out by the AMT participants, as made apparent in the different relevance values.

This experiment has showed that our analytics method is not only useful for revealing the correlations between provenance graphs and the reality they describe, but it can also serve as an automated tool for monitoring phenomenal changes in the way dynamic tasks are executed. Such tasks of course include those whose utility or performance is difficult to be qualified, such as the tasks in CollabMap. Retrospectively, had we made use of such a tool in the AMT deployment, for example, we might have noticed the anomaly in the AMT deployment (as compared with the local deployment) at a much earlier stage (than after the 8,000 microtasks had been completed).

Table 7: The relevance of each network metric given by the classifier trained with the AMT training sets (irrelevant metrics are omitted)

Network Metric	Building	Route	Route Set
number of nodes	0.474	0.893	0.230
number of edges	0.505	0.020	0.770
graph diameter	0.021	0.046	-
MFD (entity→entity)	-	0.006	-
MFD (entity→process)	-	0.035	-

5 Related Work

Our work is conducted within the context of the descriptive analysis of network graph characteristics (Kolaczyk 2009). It has been shown that when studying a complex system such as a long-term crowdsourcing application or any program giving rise to a large amount of data (provenance or otherwise), various questions of interest can be rephrased usefully as questions regarding some aspect of the structure or characteristics of the corresponding network graph (Brandes and Erlebach 2005). For example, particular notions of the importance of individual system elements may be captured by measurements related to the corresponding vertices in the network. The field is continually evolving, and graphs can be viewed in a growing number of ways; provenance data itself can be interpreted as collaboration networks (Altintas et al. 2010) or otherwise. Recently, Margo and Smogor (Margo and Smogor 2010) examined a provenance graph based on components of a file-store, to show that provenance and other metadata can successfully predict semantic attributes: in particular, they predicted file extensions in a file-history graph. ('Predict' refers not to the temporal sense of the word, but to the re-inferring of removed data.) Although their particular choice of attribute to predict "has few applications", the study functioned as a useful proof of concept. The authors employed the C4.5 decision tree algorithm on their provenance graph, with the network structure and artifact attributes as input; the levels of accuracy achieved were comparable to our own, even though in the present work we examine provenance graphs of a different topology and size. The authors recognized that "further exploration" of the feature space over provenance graphs was called for; among other things, our methodology extends the types of features used in such analyses.

Our method provides a broader type of analysis than certain previous work on hyperlink network analysis (Park 2003) in which the links between web pages were studied to estimate the value of websites (e.g. its credibility) or to identify the social networks between the pages. In the former case, the previous work only counted the number of links and did not investigate the network connections further than one link away (in contrast with the size of dependency graphs in our analyses). In the latter, the focus was on clustering similar nodes or detecting outliers, e.g. isolated nodes or those with few links, not on the properties of these nodes as in this work.

In summary, network analysis is a large research area which spans various applications and types of data. However, no previous work has studied the network metrics of provenance graphs and employed those in data quality assessment, especially in a crowdsourcing context.

6 Conclusions

Assessing the quality of crowd-generated data, often by volunteers of varied expertise, has always been challenging. It is usually a manual process that requires retrospection by experts who understand well the concerned application domain; in some other cases, it instead uses the consensus opinions of the participants (e.g. via a voting-like mechanism). In this work, we have presented an application-independent and principled method for analyzing crowd-sourced data and applications based on their provenance graphs. Using this method, it is now possible to explore and learn about some properties of crowd-generated data in an automated manner. We have demonstrated the applicability of this method within the context of CollabMap, showing that it can accurately classify the trust labels for buildings, routes, and route sets drawn by the application's contributors. While so doing, we also outline how this method could be useful for discovering the relationship between the data by the crowd and their activities and for detecting behavioral differences between two crowds. Since the method employs common network analyses and machine learning techniques on generic provenance graphs, it can serve as a generic analytics tool in a wide range of applications.

Going forward, we plan to refine our method and validate it in new application domains. The analyses here can be extended to study the provenance network metrics that characterize the evolution of provenance graphs (like those introduced by Ebden et al.), which reflects the development of the tasks they represent. Such an extension could potentially help us to understand developing dynamic behaviors in a crowdsourced task, and to make appropriate interventions on-the-fly (to stop an undesirable behavior from progressing, for instance). In addition, our topological approach could also be extended to include generic information about node attributes; for example, including the knowledge of whether votes were up or down might improve accuracy for a subclass of applications that use Find-Fix-Verify.

Provenance graphs do not only describe the origin of data, but they also reveal the interactions of agents in connected activities and how the activities themselves unfolded at the same time. The provenance analytics method presented in this work, therefore, could find new and useful applications in other areas in addition to quality assessment. Analyzing the influence of agents in the provenance graph of a collaborative task could identify the most valuable team member. Studying the distances between the agents in the graph could reveal close collaboration or team breakdown. In addition, focusing on the activities in the graph could help detect bottlenecks, important data, and activities that were crucial to the outcome of the task. Given the generic nature of network analysis techniques, the possibilities are promising and vast.

7 Acknowledgements

We gratefully acknowledge funding from the UK Engineering and Physical Sciences Research Council for project 'Orchid', grant EP/I011587/1.

References

Altintas, I.; Anand, M.; Crawl, D.; Bowers, S.; Belloum, A.; Missier, P.; et al. 2010. Understanding collaborative studies through interoperable workflow provenance. In *Third International Provenance and Annotation Workshop*, 42–58.

Aron, J. 2011. Inside the race to crack the world's hardest puzzle. *The New Scientist* 212(2841):26–27.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; Panovich, K.; and Arbor, A. 2010. Soylent: A Word Processor with a Crowd Inside. *Artificial Intelligence* 313–322.

Brandes, U., and Erlebach, T. 2005. *Network Analysis: Methodological Foundations*. Springer.

Ebden, M.; Huynh, T. D.; Moreau, L.; Ramchurn, S.; and Roberts, S. 2012. Network analysis on provenance graphs from a crowdsourcing application. In Groth, P., and Frew, J., eds., *Provenance and Annotation of Data and Processes*, volume 7525 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 168–182.

Kolaczyk, E. 2009. *Statistical Analysis of Network Data*. Springer.

Margo, D., and Smogor, R. 2010. Using provenance to extract semantic file attributes. In *Proceedings of the 2nd conference on Theory and practice of provenance, Berkeley, USA*, 7–7. TAPP'10, USENIX Association.

Moreau, L., and Missier, P. 2013. PROV-DM: The PROV Data Model. Technical report, World Wide Web Consortium. W3C Recommendation.

Moreau, L. 2010. The foundations for provenance on the web. *Foundations and Trends in Web Science* 2(2–3):99–241.

Park, H. 2003. Hyperlink network analysis: A new method for the study of social structure on the web. *Connections* 25(1):49–61.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.

Ramchurn, S. D.; Huynh, T. D.; Venanzi, M.; and Shi, B. 2013. Collabmap: Crowdsourcing maps for emergency planning. In *5th ACM Web Science Conference (WebSci '13)*.

Teacy, W. T. L.; Patel, J.; Jennings, N. R.; and Luck, M. 2006. TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources. *Autonomous Agents and MultiAgent Systems* 12(2):183–198.