Provenance for Online Decision Making

Amir Sezavar Keshavarz(✉), Trung Dong Huynh, and Luc Moreau

Electronics and Computer Science (ECS),
University of Southampton, Southampton, UK
{ask2g10,tdh,l.moreau}@ecs.soton.ac.uk

Abstract. It is commonly believed that provenance can be utilised to form assessments about the quality, reliability or trustworthiness of data. Once presented with contradictory or questionable information, users can seek further validation by referring to its provenance. While there has been some effort to design principled methods to analyse provenance, the focus has mostly been on offline use of provenance. How to use provenance at runtime, i.e., as the application runs, to help users make decisions, has been barely investigated. In this paper, we propose a generic and application-independent approach to interpret provenance of data to make online decisions. We evaluate the system in CollabMap, an online crowd-sourcing mapping application, to make decisions about the quality of its data and to determine when the crowd's contributions to a task are deemed to be complete.

Keywords: Provenance · Online decision making · Validity measure · Reliability measure

1 Introduction

It is commonly believed that provenance can be utilised to form assessments about the quality, reliability or trustworthiness of data [6]. Provenance is defined as a "record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing" [7]. It is a crucial piece of information that can help a consumer make a judgement as to whether something can be trusted [8].

A provenance-aware system can generate the provenance of its data and make it accessible to other systems that may use it for other purposes. However, the provenance that is recorded can be application-specific. In order to use it, other systems may require it to be recorded differently and the original application to rerun as a result. This may not be possible nor efficient. As such, we need a principled mechanism for application-specific interpretation of provenance. For example, consider a provenance-aware system that requires users to interact with the system and generate some data. Another application might need to make decisions based on the ratings of the data and users that can be computed from the provenance recorded by the first application. Such a system will need to be able to: (1) Interpret application-specific provenance, (2) Compute ratings for

B. Ludäscher and B. Plale (Eds.): IPAW 2014, LNCS 8628, pp. 44–55, 2015.
DOI: 10.1007/978-3-319-16462-5_4

the entities generated, and (3) Use provenance-based ratings to make decisions in a timely manner.

In order to address these requirements, we introduce an online provenance analysis system that is composed of the following. First, we propose a generic Annotation Computation Framework (ACF) that enables applications to attach application-specific annotations to elements of a provenance graph and to compute new annotations from these.

Second, we put forth a statistical Quality Model (QM) that computes three ratings for data entities and users from their provenance: (1) a *validity measure* for data entities with different validity labels ("valid", "invalid", or "uncertain"), (2) a *reliability measure* for each user reflecting how consistently good their performance is, and (3) a *finish measure* with different finish labels ("finished", "unfinished") that expresses if further user contributions are required.

Finally, we devise an Online Annotation Computation System (OACS) that enables a provenance-aware system to make decisions in a timely manner from its provenance. OACS defines a contract according to which a provenance-aware system should model its provenance, submit it to the OACS, and retrieve ratings (in the form of an annotated provenance graph).

We evaluate the ACF, OACS, and QM in a crowd-sourcing application. To support online quality-based decision making, QM rates each task in the crowd-sourcing application as either finished or unfinished by using validity and reliability measures. The OACS, alongside with QM, helps increase the confidence on validity measure by analysing the performance of users (reliability measure) *during* the execution of the crowd-sourcing application.

To the best of our knowledge, online use of provenance for quality-based decision making has not been previously investigated. Our framework with the online mechanism provides a foundation that enables a provenance-aware system to make online decisions based on provenance of data.

Our contributions are fourfold:

1. A generic Annotation Computation Framework to allow application-specific interpretation of provenance;
2. An Online Annotation Computation System to assist a provenance-aware system to make online decisions during execution time;
3. A statistical quality model that uses provenance to compute (1) validity measure for data entities, (2) reliability measure for users, and (3) finish measure for data entities. It also increases the confidence on validity measure by using reliability measure in an online environment;
4. An evaluation of ACF, OACS, and QM in a crowd-sourcing application called CollabMap.

The remainder of this paper is structured as follows. In Sect. 2, a crowd-sourcing application in which provenance is recorded is introduced. Section 3 and 4 present ACF and OACS, respectively. We specialize online decision making to online quality-based decision making by implementing a statistical Quality Model and show how CollabMap utilises computed ratings to make online quality-based decisions (Sects. 5 and 6). Section 7 presents and discusses the

evaluation of ACF, OACS, and the QM. Section 8 provides the related work and Sect. 9 outlines the future work.

2 Scenario: A Crowd-Sourcing Application

CollabMap is a crowd-sourcing application that recruits people to augment existing maps by identifying buildings outline and drawing their evacuation routes from buildings to nearby roads. Participants are required to verify tasks by others by providing positive or negative votes on buildings and evacuation routes, helping CollabMap to determine their validity. The quality of data generated by a crowd with different backgrounds and expertise is inevitably varied. Therefore, two mechanisms to ensure data quality were suggested for CollabMap.

1. Online Majority Voting: The first version of CollabMap (CollabMap-V1) [9] employed a customized adaptation of majority vote. If total sum of positive and negative votes is above +3, then the building is marked as valid. If the score reaches −2, the building is marked as invalid. Provenance was recorded in CollabMap-V1 but was not used to assess the validity of data.
2. Offline Provenance Network Analysis: In the second version of CollabMap (CollabMap-V2), Huynh et al. [5] extracted a set of provenance network metrics from provenance of data to learn about patterns that correlate with quality of data. This approach was not used online either.

User's reliability was not considered for decision making in either of the above versions. Therefore, to improve the quality assessment done by CollabMap in an online environment, we set the following requirements for our system.

Requirement 1. *To compute a validity label ("valid", "invalid", or "uncertain") for each data entity (buildings and evacuation routes). "valid" data entities are to be included in the final result, while "invalid" data entities are to be discarded. In cases where the validity label is "uncertain", further users would be employed to verify the "uncertain" data.*

Requirement 2. *To compute a reliability measure for each user, so that we can use these measures to increase confidence on validity label by analysing users' reliability.*

Requirement 3. *To compute a finish measure for each data entity to decide to continue or terminate a task.*

Provenance would be used to capture all these measures while CollabMap is executing. In this context, use of provenance offers the following benefits: (1) a generic foundation that provenance recorded in CollabMap can be used in a provenance-aware rating application to compute such measures and (2) a data model that captures all the changes and decisions that are made in CollabMap by using above measures in an online environment.

3 Annotation Computation Framework (ACF)

Provenance-based rating can be decomposed in a generic part involving a provenance graph traversal and annotation manipulation, and an application-specific part computing actual ratings for a given purpose in an application. ACF implements the generic part and allows for instantiations to add the application-specific part.

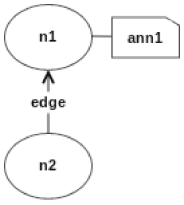
Annotations are utilized as a generic mechanism to enable any information to be attached to elements of a provenance graph. Following provenance record presents the use of an application-specific annotation for a building. The annotation, `validity`, represents validity label of the building.

```
entity(ex:building1,[ex:validity="valid"])
```

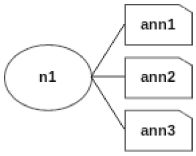
New annotations can be computed for a node from existing annotations for the same graph. In order to compute new annotations, both forward and backward computations are supported. Forward computation is the computation of annotations by following relations between nodes along the direction of time; and vice-versa for backward computation.

Three fixed rules are considered to support computation of annotations. Forward computation rule is defined by (1). In this rule, given there is a directed relation from an influencee ($n2$) to an influencer ($n1$), and influencer ($n1$) has an annotation ($ann1$), a new annotation for the influencee is computed based on $ann1$ and defined by $F_{forward}$. To allow application-specific interpretation, a function F is required to compute the new annotation. Backward computation rule has a similar definition except an influencee node has an annotation and a new annotation for the influencer is computed.

The third computation rule, aggregation rule (See 2), is applied when a node ($n1$) has more than one annotation (e.g. $ann1$, $ann2$, $ann3$, ...). In this case, a new annotation, ($aggAnn$), is computed based on all its existing annotations (defined by F_{Agg}).



IF $\mathbf{G} \vdash$
$node(n1, ann1)$
$node(n2)$
$edge(e; n2, n1)$ (1)
THEN there exists $ann2$ such that
$ann2 = F_{forward}(G, e, n1, n2, ann1)$
$update(G, n2, ann2)$



IF $\mathbf{G} \vdash$
$node(n1, ann1)$
$node(n1, ann2)$
$node(n1, ann3)$
... (2)
Then there exists $aggAnn$ such that
$aggAnn = F_{Agg}(G, n1, [ann1, ann2, ann3, ...])$
$update(G, n1, aggAnn)$

To ensure termination, we introduce a global counter that acts as a bound for termination. After each computation step, the counter is deduced by one. By limiting the number of computations, we guarantee that computation rules will eventually terminate.

4 Online Annotation Computation System (OACS)

OACS introduces a contract according to which a provenance-aware system should model its provenance. This contract makes two assumptions on the provenance with respect to: (1) the structure of provenance data, and (2) the annotations to be incrementally computable.

An important capability of provenance is to express revisions of the same resource. "Plan for revisions" recipe [8, Section 4.1.4] is used for this purpose. Each version of a resource is connected to a single general resource using the prov:specializationOf relation. Each version is related to its previous one using prov:wasRevisionOf relation. Assumptions 1 and 2 are as follow.

Assumption 1. *Provenance is expected to be structured according to the "Plan for revisions" recipe, so data entities are continuously rated by OACS.*

Assumption 2. *OACS expects annotations of any version of a resource to be computable by using the annotations of its previous version without the need for the full provenance of its previous versions.*

The following describes the steps through which a provenance-aware system should submit its provenance data and retrieve newly computed annotations.

Step 1 Whenever there is new provenance data that needs to be annotated, the provenance-aware system is required to bundle up all new assertions (A bundle is a named set of provenance descriptions [7]);
Step 2 In this bundle, the system is required to identify each element of provenance graph to be "annotated" as a distinct element according to the two contractual Assumptions 1 and 2;
Step 3 Submit this bundle to the OACS;
Step 4 When a response from OACS is ready, the system can retrieve a provenance bundle that contains the new annotations from OACS;
Step 5 New annotations can then be used to update the system's local state and to make application-specific decisions;
Step 6 Return to Step 1.

In a decision making situation, as an application is executing and more knowledge is generating, the application is presented with updated information which decisions are based on. In order to validate the updated information and new decisions, the decision makers can consider their provenance. As such we decided to use bundle to allow provenance of provenance to be expressed.

5 Quality Model (QM)

5.1 Validity Measure (VM)

Requirement 1 requires data entities to be annotated with a validity label ("valid", "invalid", or "uncertain"). Table 1 summarises the annotations processed by VM.

Table 1. Annotation Assertion (AA) and Annotation Computation (AC) in Validity Measure for a data entity D (building or evacuation route)

Annotation	Description	Value	Level
Vote	Value of user (U) vote for D	$Vote(D, U)$	AA
Coordinates	Coordinates of a building	$Coord(D)$	AA
Edges	Total number of edges	$Edge(D)$	AA
Positive votes	Number of positive votes for D	$P(D)$	AC
Negative votes	Number of negative votes for D	$N(D)$	AC
Validity label	Validity label of D	$V(D)$	AC

The beta family of probability density functions model the distribution of a random variable representing the unknown probability of a binary event where $T(D)$ is an example of such a variable to model [11]. In (3), $Beta(\alpha, \beta)$ returns the probability of D being valid provided α and β, where α and β are hyper-parameters to define the shape of the density function.

$$Beta(\alpha, \beta) = \frac{\alpha}{\alpha + \beta} \tag{3}$$

Hence, validity measure ($T(D)$) is given by:

$$T(D) = Beta\Big(\big(P(D) + 1\big), \big(N(D) + 1\big)\Big) \tag{4}$$

Now, validity label ($V(D)$) can be defined by two thresholds $t1$ and $t2$:

$$V(D) = \begin{cases} valid & \text{if } T(D) \geq \text{t2} \\ invalid & \text{if } T(D) \leq \text{ t1 or Edge(D)} < 4 \text{ for building} \\ & \quad \text{or Coord(D) has self-intersecting lines for building} \\ uncertain & \text{if } \text{t1} < T(D) < t2 \end{cases} \tag{5}$$

By analysing CollabMap-V1, 0.7 and 0.3 are the threshold we chose for $t1$ and $t2$ respectively, to select valid and invalid data entities.

5.2 Reliability Measure (RM)

Requirement 2 requires each user to be annotated with a reliability measure. A user can have two roles in CollabMap: identifiers (those who generate data) and verifiers (those who verify generated data). For identifiers, we are interested in computing their total number of "valid" and "invalid" identifications. For verifiers, we are interested in computing the total number of "aligned" (positive vote on a valid data or negative vote on an invalid data) and "non-aligned" (positive vote on an invalid data and negative vote on valid data) verifications. Table 2 summarises the annotations processed by RM.

Table 2. Annotation Assertion (AA) and Annotation Computation (AC) in Reliability Measure for a user U - D can be a building or an evacuation route

Annotation	Description	Value	Level
Good identification	Total number of good D identification for U	$M(I, G, D, U)$	AC
Bad identification	Total number of bad D identification for U	$M(I, B, D, U)$	AC
Aligned	Total number of aligned votes for U	$M(V, A, D, U)$	AC
Non-aligned	Total number of non-aligned votes for U	$M(V, N, D, U)$	AC
Identification reliability	Reliability of U in D identification	$R(I, D, U)$	AC
Verification reliability	Reliability of U in D verification	$R(V, D, U)$	AC

User reliability, $R(I, D, U)$ and $R(V, D, U)$, is computed by applying (3):

$$\begin{aligned} R(I, D, U) &= Beta\Big(\big(M(I, G, D, U) + 1\big), \big(M(I, B, D, U) + 1\big)\Big) \\ R(V, D, U) &= Beta\Big(\big(M(V, A, D, U) + 1\big), \big(M(V, N, D, U) + 1\big)\Big) \end{aligned} \tag{6}$$

5.3 Finish Measure (FM)

Requirement 3 requires each data entity to be annotated with a finish measure, which is computed from reliability measure. Table 3 summarises the annotations processed by FM.

Table 3. Annotation Assertion (AA) and Annotation Computation (AC) in Finish Measure for data entity D (building or evacuation route)

Annotation	Description	Value	Level
Cumulative users' reliability	Total cumulative users' reliability	$C(D)$	AC
Finish label	Label showing if a task is terminated	$F(D)$	AC

Finish measure, $C(D)$, is computed by applying (7):

$$C(D) = \sum_{U \in VD} R(V, D, U) \quad \text{where VD is the set of all verifiers of D} \tag{7}$$

Now, the finish label ($F(D)$) can be assigned based on the finish measure:

$$F(D) = \begin{cases} Yes & \text{if } C(D) \geq \text{ t2} \\ & \quad \text{or } C(D) \leq \text{ t1} \\ & \quad \text{or Edge(D) } \leq 4 \text{ for building} \\ & \quad \text{or Coord(D) has self-intersecting lines for building} \\ No & \text{if t1 } < C(D) < \text{ t2} \end{cases} \tag{8}$$

By analysing CollabMap-V1, $+1.5$ and -1.5 are the threshold we chose for $t1$ and $t2$ respectively, to annotate a data entity as finish assuring the crowd's contributions to the data entity is deemed to be complete with high confidence.

6 Decision Making in CollabMap

CollabMap-V3 uses the measures computed in Sect. 5 to make a decision on next course of action:

$$\text{Decision} = \begin{cases} \text{Continue} & \text{if } F(D) = No \\ \text{Terminate} \begin{cases} \text{Not-Accept} & \text{if } \Big(F(D) = Yes \text{ and } V(D) = Invalid\Big) \\ & \text{or } \Big(F(D) = Yes \text{ and } V(D) = Uncertain \text{ and } C(D) \leq -1.5\Big) \\ \text{Accept} & \text{if } \Big(F(D) = Yes \text{ and } V(D) = Valid\Big) \\ & \text{or } \Big(F(D) = Yes \text{ and } V(D) = Uncertain \text{ and } C(D) \geq 1.5\Big) \end{cases} \end{cases} \tag{9}$$

7 Experiments and Results

For a preliminary evaluation, we develop several hypotheses that we validate by applying online quality-based decision making to CollabMap-V3 and examining the results. We designed an experiment where 22 users were recruited to work with CollabMap-V3 which ran over 60 h.

Hypothesis 1. *The validity label reflects the actual validity of data as verified by an expert.*

Method 1. We asked an expert to verify all the identified data entities. Then we compared the expert's opinion with the dataset of data entities that were accepted or not-accepted by CollabMap-V3.

Analysis 1. In total, 237 buildings were identified (235 annotated as finished and 2 as unfinished; CollabMap accepted 75 % of all finished buildings and discarded the rest as they were annotated as invalid). 183 evacuation routes were identified (around 80 % annotated as finished and 20 % as unfinished; CollabMap accepted above 98 % of all finished ones). The dataset of data entities in CollabMap-V3 matched the verified data by expert; thus verifying Hypothesis 1.

Hypothesis 2. *User's reliability measure reflects the actual performance and reliability of a user.*

Method 2. We formed a control group where we asked two users to consistently draw valid buildings, two users to consistently draw invalid buildings, and two users to consistently provide verification votes opposed to what they reckon to be true (to provide negative verification votes for valid data entities and vice versa for invalid data entities).

Analysis 2. Figure 1a represents the reliability measures for two users. The reliability measure for all users are similar at the beginning. As they continue engaging with the system, RM updates users' reliability measure based on their performance. The reliability measure for User 433, who consistently draw invalid buildings, decreased from 50 % to less than 2 % (blue dashed line). Whereas the reliability measure for User 427, who consistently draw valid buildings, increased from 50 % to 98 % (red line). At this point, we can validate Hypothesis 2 as users' reliability measure truly reflects their performance.

Hypothesis 3. *If reliable users verify a data entity, the task can be terminated faster than when unreliable users verify a data entity.*

Method 3. We evaluate if (1) QM can incrementally learn the reliability of users and (2) the reliability measure of users was used to terminate the task.

Analysis 3. Figure 1b represents the proportion of finished and unfinished data entities over time. As expected, at the beginning, the growth ratio of unfinished data entities (white boxes on top) is higher than finished data entities (red pattern filled boxes). However, as RM is gaining knowledge about users, the growth ratio of finished data entities are higher.

Figure 1c represents the total number of data entities that are being annotated by their validity and finish labels over the time. At the beginning, as RM does not have enough knowledge about the users, most data entities are annotated as Unfinished-Uncertain (blue triangle-dotted line). As time progresses,

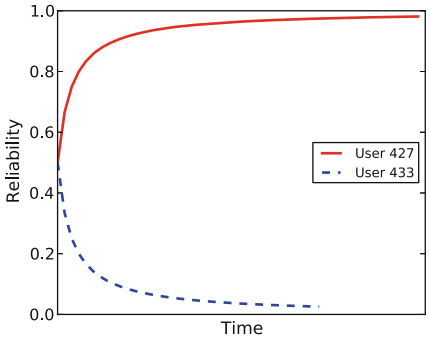
RM gains enough knowledge over users to annotate data entities as Finished-Valid/Invalid (the rapid jump in green diamond-dashed line). We can observe another growing trend and it is those data entities that are annotated as Finished-Uncertain (red square-solid line). This shows that QM reduces the number of votes required when it has gained enough knowledge about the participants. We expect more growth in this line had we let our trial continued. The reason is at the beginning, all users have the similar reliability measure and it takes time to annotate a data entity as finished. As time progresses, reliable users are identified and they will have a higher reliability measure (reliability of some users were measured as above 90 %) which means the data entity is annotated as finished earlier.

Figure 1d represents an average number of votes required to annotate a data entity as finished. The total number of requested votes depends on the finish measures. After one day of execution, there is a decreasing trend that QM requires less votes to annotate a data entity as finished. CollabMap-V1 requires at least 3 votes to terminate a task. As can be seen from Fig. 1d, the average total number of votes requested for a finished building over times, was reduced to 2.5. At this point, it is possible to verify Hypothesis 3. Although at the beginning, QM may require more verification votes, there is a decreasing trend in requesting verification votes toward the end.

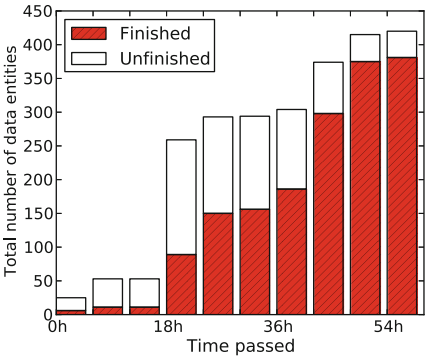
8 Related Work

Provenance can be used to estimate quality of data and data reliability based on the source data [10]. Golbeck reviews trust issues on the World Wide Web [3] and identifies provenance as a key element necessary to derive trust. One trustworthiness [2]. Dai et al. [2] propose a method to compute trust scores for data, depending on the trust of the information used to generate it. In order to assess quality of data and reliability of users, Allen et al. [1] describe a provenance system, PLUS, that uses provenance of data to detect potential malicious behaviour and help users assess trust in information. On the same venue of work, Hartig et al. [4] propose a model for Web data provenance and an assessment method that can be adapted for specific quality criteria. None of these works used provenance to infer trustworthiness of data nor performance of users in an online environment. Our approach motivates the use of provenance in an online environment where quality-based decisions can be made in a timely manner.

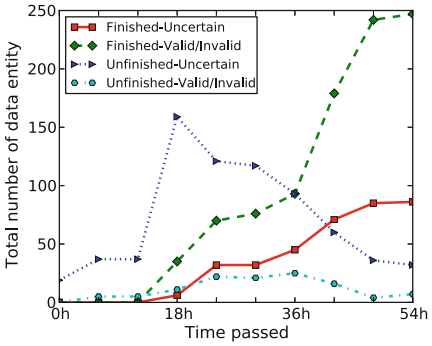
The issue of data quality and user reliability can be observed in crowd-sourcing applications. In a crowd-sourcing application, tasks are broken down into smaller activities and are allocated to the crowd; upon completion, some rewards are issued. There are mainly two issues associated with some crowd-sourcing applications: (1) quality of generated data, and (2) evaluation of user performance. To assure quality, the crowd-sourcing application assigns the same labelling task to multiple users. When multiple labels are provided for the same task, the crowd-sourcing application fuses all labels to estimate the actual label. Whitehill et al. [12] present a probabilistic model to compute the expertise of



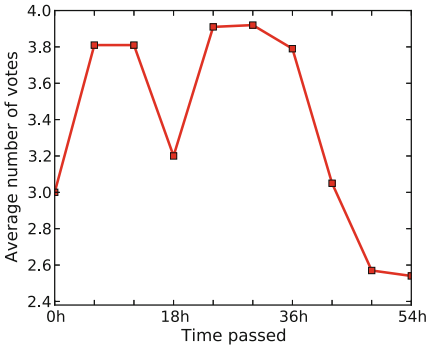
(a) The reliability measures of 2 users in control group



(b) Proportion of finished and unfinished data entities



(c) Finish and Validity measures for data entities



(d) Average number of votes for finished data entities

Fig. 1. Analysis of CollabMap-V3 deployment (Color figure online)

each user, difficulty of each task, and the label of each task. Our proposed QM is similar to this approach in computing a validity and reliability measures, however in contrast, we use reliability measures to decide to continue or terminate a task while the application is executing.

9 Conclusion

In this paper, we have presented a principled approach for online application-specific interpretation of provenance that consists of: (1) a generic part involving a provenance graph traversal and annotation manipulation, (2) an application-specific part computing the annotations for data quality assessment and task termination.

We carried out a preliminary analysis of the approach on CollabMap, a crowdsourcing application for designing evacuation maps. We showed that it is able to classify data with high accuracy by analysing the reliability of contributors while

the application is executing. We also demonstrated that with our framework, CollabMap was able to make online decisions whether to terminate or continue a task.

Going forward, we plan to deploy CollabMap and ACF in a wider community, employing more users for a complete empirical evaluation of our framework. It would allow us to evaluate how accurately ACF can help CollabMap to terminate tasks. Furthermore, we plan to use user's reliability measure to decide whether to accept or reject a user's contribution and to explore dynamic task allocation to users based on their reliability.

Acknowledgements. This work is funded by the EPSRC ORCHID Project (EP/-I011587/1).

References

1. Allen, M.D., et al.: Provenance for collaboration: detecting suspicious behaviors and assessing trust in information. In: 2011 7th International Conference on Collaborative Computing, pp. 342–351. IEEE (2011)
2. Dai, C., Lin, D., Bertino, E., Kantarcioglu, M.: An approach to evaluate data trustworthiness based on data provenance. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, pp. 82–98. Springer, Heidelberg (2008)
3. Golbeck, J.: Trust on the world wide web: a survey. Found. Trends Web Sci. **1**(2), 131–197 (2006)
4. Hartig, O., Zhao, J.: Using web data provenance for quality assessment. In: SWPM, vol. 526 (2009)
5. Huynh, T.D., Ebden, M., Venanzi, M., Ramchurn, S.D., Roberts, S., Moreau, L.: Interpretation of crowdsourced activities using provenance network analysis. In: First AAAI Conference on Human Computation and Crowdsourcing (2013)
6. Moreau, L.: The foundations for provenance on the web. Found. Trends Web Sci. **2**(2–3), 99–241 (2010)
7. Moreau, L., Groth, P.: Prov-dm: The prov data model. Technical report (2013). http://www.w3.org/TR/prov-dm/
8. Moreau, L., Groth, P.: Provenance: an introduction to prov. Synth. Lect. Semant. Web: Theory Technol. **3**(4), 1–129 (2013)
9. Ramchurn, S.D., Huynh, T.D., Venanzi, M., Shi, B.: Collabmap: crowdsourcing maps for emergency planning. In: Proceedings of the 5th Annual ACM Web Science Conference, pp. 326–335. ACM (2013)
10. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM SIGMOD Rec. **34**(3), 31–36 (2005)
11. Teacy, W.L., Patel, J., Jennings, N.R., Luck, M.: Travos: trust and reputation in the context of inaccurate information sources. Auton. Agent. Multi-Agent Syst. **12**(2), 183–198 (2006)
12. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J.R.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: NIPS, vol. 22, pp. 2035–2043 (2009)