POSTER: On Trust Evaluation with Missing Information in Reputation Systems

Xi Gong
North Carolina State University
xgong2@ncsu.edu

Ting Yu
North Carolina State University
tyu@csc.ncsu.edu

Adam J. Lee
University of Pittsburgh
adamlee@cs.pitt.edu

ABSTRACT

Reputation plays a critical role in managing trust in decentralized systems. Quite a few reputation-based trust functions have been proposed in the literature for many different application domains. However, one cannot always obtain all information required by the trust evaluation process. For example, access control restrictions or high collect costs might limit the ability gather all required records. Thus, one key question is how to analytically quantify the quality of scores computed using incomplete information. In this paper, we start a first effort to answer the above question by studying the following problem: given the existence of certain missing information, what are the worst and best trust scores (i.e., the bounds of trust) a target entity can be assigned? We formulate this problem based on a general model of reputation systems, and examine the monotonicity property of representative trust functions in the literature. We show that most existing trust functions are monotonic in terms of direct missing information about the target of a trust evaluation.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; K.4.4 [**Computer and Society**]: Electronic Commerce—*Security*

General Terms

Security

Keywords

Reputation Systems, Trust Functions, Missing Information

1. INTRODUCTION

Large-scale decentralized systems, such as P2P networks and online auction communities, allow entities from different security domains to interact and conduct business with each other. As the priori trust relationships do not typically exist between entities, establishing trust in other users is a key problem in the design of decentralized systems. Reputation mechanisms are a prominent technique for trust management in these types of systems. In a reputation system, once a transaction is finished, involved entities issue feedbacks that evaluate each other's service or behavior during the transaction. Before a new transaction starts, one may first assess an entity's trustworthiness based on the feedbacks on its previous transactions. This process can be viewed as the application of a so-called *trust function* that takes as input feedback from an entity's past transactions (and possibly those of other related parties), and outputs a trust value to indicate its trustworthiness.

However, in large-scale decentralized systems all information required by the trust function may not be available for a variety of reasons. For example, due to privacy concerns, some entities may impose access control restriction regarding how feedbacks that they issue can be accessed by others. In other situations, especially in P2P networks, feedbacks are stored in a distributed manner among multiple entities. Some entities may not be online at the time of trust computations, and thus the feedbacks stored by these entities may be unavailable. Sometimes, even if all necessary feedbacks are retrievable, it may be too costly to collected them all. As a result, the computation of trust ratings based upon incomplete information is not uncommon.

A natural question is thus how to quantify the trust computed using incomplete information. For instance, is it wise to conduct a transaction with an entity based upon incomplete trust information? Have we sampled sufficient feedbacks so that the computed trustworthiness is close enough to the true one?

There are many ways to formulate the above question. For example, if we are aware of the distribution of the missing information (e.g., the distribution of the rating of a missing feedback), we may be interested in the distribution of the trustworthiness of an entity. In this paper, we study one possible version of this problem: given the range of possible values taken by missing information, what are the possible trustworthiness ratings for an entity? In particular, we are interested in establishing the worst- and best-possible trustworthiness of the entity (bounds of one's trustworthiness). Answering this problem can help us make many decisions in decentralized systems. For instance, suppose our policy is to conduct a transaction with an entity if its trustworthiness is over 0.8. If the bounds of its trustworthiness can be established as $[0.8, 0.9]$, then it is safe to do business with that entity, even if some information is missing. For sampling based approaches to dealing with missing information, this bound can guide us whether it is appropriate to stop sampling. For example, a wide bound, e.g., $[0.2-1]$, suggests the sampling is insufficient, resulting in big uncertainty about one's trustworthiness. On the other hand, with a narrow

CCS'11, October 17–21, 2011, Chicago, Illinois, USA.
ACM 978-1-4503-0948-6/11/10.

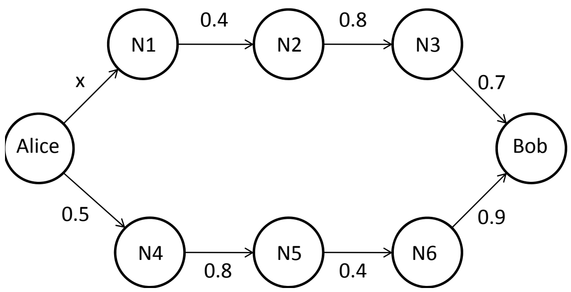

Figure 1: Trust Graph

bound, e.g., $[0.85-0.9]$, it is probably fine to stop sampling, as further sampling will not increase much the accuracy of the estimation of one's trustworthiness.

This paper represents our first effort towards answering the above trust bounding problem. We present a formal description of the problem based on a general model of reputation systems, and then study the *monotonicity* of a large set of existing trust functions in the literature. Intuitively, if a trust function is monotonic regarding a piece of missing information, then the bounds of an entity's trustworthiness can be easily computed by only considering the min/max value of the missing information. Our initial investigation shows that most existing trust functions in the literature are monotonic regarding missing information directly related to the target of trust evaluation. However, for indirect missing information, the problem is more complicated, which requires further research.

2. REPUTATION SYSTEMS

Entities in decentralized systems interact through transactions. Without loss of generality, we assume a transaction is uni-directional, i.e., there is a clear distinction between a service provider and a service consumer. Note that a service provider in one transaction may be the consumer of another, and vice versa. A feedback is issued by a consumer about a provider after a transaction. It can be denoted as (c, s, r, t), indicating that consumer c issues a feedback about provider s at time t with a rating r.

Though multiple transactions may happen between a pair (c, s) of consumer and provider, most trust functions in the literature consider an aggregation of these transactions, e.g., the ratio of positive or negative transactions among all the transactions. We call such an aggregation the opinion of c over s, or c's local trust of s. Therefore, a reputation system can be modeled as a weighted directed graph $G(V, E)$, where $v \in V$ is an entity in the system, and an edge $(c, s) \in E$ with weight w represents that c's opinion over s is w. We call this graph a *trust graph*. Figure 1 shows an example trust graph.

A trust function F takes as input a trust graph G, a source entity u and a target entity v, and returns a trust score t. Intuitively, given the feedback information stored in a system, F computes the trustworthiness of v from u's point of view. For most trust functions, for different u and u', $F(G, u, v)$ may not be the same as $F(G, u', v)$ [2]. Such trust functions reflect subjective trust. Some other trust functions compute a unique global trust score for every entity, instead of from another entity's perspective [1]. Nevertheless, such functions can still be captured by the above model.

3. TRUST BOUNDING PROBLEM

Several types of information may be missing from a sampled trust graph. For example, a subgraph of the complete trust graph G may not be accessible. Or we may not know for sure whether there is an edge between a pair of entities. In this paper, we restrict our discussion to the case that the topology of the graph is complete, but the weight of some edges are uncertain. Such uncertainty can be captured by a set of variables $x_{e_1}, \ldots, x_{e_n}$, each corresponding to the weight of an edge e_i. Meanwhile, each x_{e_i} is associated with a domain, which are the possible weights the edge may have, according to the collected available information. For example, if we know that there were ten transactions v provided to u in the past, and we only collected 8 feedbacks, all of which are positive, then the domain for $x_{u,v}$ are $0.8, 0.9, 1.0$, if the weight is the positive transaction ratio. Sometimes the domain of a variable may be a range instead of a finite set of values. For example, if we cannot collect any feedback from u to v, though we know u did issue feedbacks about v, then the domain of $x_{u,v}$ would be $[0-1]$.

Trust Bounding Problem. Given a trust graph G with edge variables $x_{e_1}, \ldots, x_{e_n}$, along with their domains, a pair of entities (u, v), and a trust function F, compute the minimum and the maximum of $F(G, u, v)$.

Given two entities u and v, we say $F(G, u, v)$ is *positively (negatively) monotonic* to a variable x_{e_i} if by fixing all the edge variables but x_{e_i} to any values, we have that $F(G, u, v)$ is non-decreasing (non-increasing) as x_{e_i} increases (decreases).

It is not hard to see that if $F(G, u, v)$ is monotonic to all variables (either positively or negatively), then we can easily compute the minimum of $F(G, u, v)$ by simply taking the minimum of x_{e_i} if $F(G, u, v)$ is positively monotonic to x_{e_i} or the maximum of x_{e_i} if $F(G, u, v)$ is negatively monotonic to x_{e_i}. The maximum of $F(G, u, v)$ can be similarly obtained.

Next we report our investigation on the monotonicity of representative trust functions in the literature. For brevity, we only consider the case with a single variable $x_{i,j}$ that is direct to the target, i.e., j is the target of a trust evaluation. Otherwise, we say $x_{i,j}$ is indirect.

4. PRELIMINARY RESULTS

4.1 Trust Function Classification

Though trust functions proposed in the literature differ in their specific design, most of them fall into the following categories in terms of their design principles.

1. *Recommendation based.* The idea of this type of trust functions can be explained with the help of the following formula.

$$t_{uv} = \sum_{j \in S} w_{uj} \cdot t_{jv}. \tag{1}$$

   When entity u wants to evaluate the trust of entity v but does not have direct interaction with v before, he will ask for opinions from a set S of witnesses j who have directly interacted with v. These opinions are notated as t_{jv}, and are combined using weights w_{uj} which represent u's belief in j's opinion. Example trust functions that fall into this category include [4] and Credence [5]. The key of this type of functions lies in the definition of weights. For instance, in [4] weights are computed based on the reliability of recommendations the witness made in history, while in Credence the weights depend on the statistical correlation between the vote history of the truster and that of the witness. One key observation is that the opinion t_{jv} is not used for computing the weight of w_{uj}. In other words, an edge variable x_e either affects an opinion or a weight, but not both at the same time.

2. *Topology based.* This type of function explicitly exploits the topology of a trust graph. When an entity u wants to evaluate another entity v he did not have interactions before, he will try to find a path starting at u and ending at v in which each pair of neighboring nodes has directly interacted before. Then some concatenation algorithm is used to derive the strength of a path. The final trust is achieved from a weighted aggregation over all the possible paths. Though this type of functions share a similar principle with the above recommendation-based trust functions, the key difference is that the weight on a single edge may contribute to both the strength of a path and the weight of that path when doing aggregation. One example of this type of functions is the trust function proposed in the NICE system [2].

3. *Matrix based* This type of function takes the adjacency matrix of a trust graph, and the final trust scores of entities are defined as some converged value of the matrix, for some specific iterative matrix operations. PeerTrust [6], EigenTrust [1] and Path Algebra [3] are typical examples of this type of functions.

4.2 Monotonicity Analysis

In the following we summarize our observation of the monotonicity of the above types of trust functions.

1. *Recommendation based* It is not hard to prove that this type of function is monotonic with respect to a single edge variable. With the assumption that w_{uj} is nonnegative (which is true for all existing trust functions of this type), positive monotonicity is guaranteed regarding direct missing information. This can be seen from the fact that direct missing information only contributes to the recommendation from a witness, but not the weight of that recommendation. Clearly, with all the weights fixed, the higher a witness's recommendation, the higher the overall trust score.

2. *Topology based* This type of functions is also monotonic regarding direct missing information. In the construction of existing trust functions of this type, a direct edge variable x_e only monotonically affects the strength of a path but not the weight of the path. Therefore, its impact to the final trust score is also monotonic.

3. *Matrix-based* Matrix-based function is more complicated. Due to the usage of specific iteration operations, it is difficult to provide a general proof. However, from a case study of PeerTrust, EigenTrust and Path Algebra, we get some desirable results. Our simulation results indicate that all of these three functions are positively monotonic with respect to single direct edge. Moreover, we have provided complete proof for both of PeerTrust and Path Algebra. Unfortunately, we have not established a theoretical proof for EigenTrust, due to its special normalization process. This proof is one of the focuses of our future work.

5. FUTURE WORK

Our focus next is to investigate the case of indirect missing information. Many trust functions are not monotonic regarding indirect edge variables. For example, in NICE, since an indirect edge may affect both the strength of a path and its weight, the overall impact on the final trust score is more complicated. If the strength of a path is the minimum among the weights of all the edges in the path, and the weight of the path is the weight of the first hop of the path (a typical setting for NICE), we can easily construct a case, where the trust score is not monotonic to an edge variable who is the first hop of a path. It is also the case for EigenTrust due to its normalization process. However, we do observe through simulation that other matrix-based functions like PeerTrust and Path Algebra seem still monotonic to indirect missing information, which we plan to formally prove in our future work.

Another interesting line of work is to study the probabilistic setting of the problem of missing information in reputation systems, i.e., given the distribution of some unknown variables, what is the distribution of a final trust score. The answer to this question will help us quantify the risk when we base our trust decision on incomplete information.

Acknowledgments

This research is supported in part by the National Science Foundation under awards CCF-0916015, CNS-0964292, CNS-1017229 and CCF-0914946.

6. REFERENCES

[1] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. *Proceedings of the twelfth international conference on World Wide Web*, page 640, 2003.

[2] S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in NICE. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1272–1282. IEEE, 2003.

[3] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. *In Proceeding of the Second International Semantic Web Conference*, 2003.

[4] E. S. Staab and S. Staab. The Pudding of Trust Editor's Perspective. *IEEE Intelligent Systems*, 3(4):19(5):74–88, 2004.

[5] K. Walsh and E. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *Proceedings of the 3rd conference on Networked Systems Design & Implementation-Volume 3*, 2006.

[6] L. Xiong and L. Liu. Building trust in decentralized peer-to-peer electronic communities. *Fifth International Conference on Electronic Commerce*, 2002.