A Classification Scheme for Trust Functions in Reputation-Based Trust Management

Qing Zhang[1], Ting Yu[1], and Keith Irwin[1]

Department of Computer Science
North Carolina State University
{qzhang4, tyu, kirwin}@ncsu.edu

Abstract. Reputation is an important means to establish trust in decentralized environments such as the Semantic Web. In reputation-based trust management, an entity's reputation is usually built on feedback from those who have direct interactions with the entity. A trust function is used to infer one's trustworthiness based on such feedback. Many trust functions have been proposed in the literature. They are typically designed for specific application domains, thus differ in a variety of aspects, including trust inference methodologies, complexity and accuracy. In this paper, we propose a classification scheme for trust functions, which will help the systematic analysis and selection of trust functions for particular applications.

1 Introduction

The Semantic Web is visioned as the next Web. By instilling semantics into ”flat” documents, the Semantic Web enables users to retrieve information that satisfies queries much more complex and expressive than today's keyword-based queries.

Due to the decentralized nature of the Semantic Web, an entity can publish whatever information without going through any scrutiny. Though seemingly relevant information can be retrieved, there is no assurance of its usefulness and correctness. Cryptographic techniques may be used to protect information confidentiality and integrity, but not its quality. To mitigate the problem, the trustworthiness of information sources needs to be considered.

Reputation is an important means to establish trust in decentralized environments. An entity's reputation is usually built on feedback from those who have direct interactions with the entity. Given a set of feedback, one's trustworthiness can be inferred through the use of *trust functions*. Trust functions have a direct impact on one's decision regarding information collection and other critical tasks, and is a key component to any reputation-based trust model. Many trust functions have been proposed, targeting at different application domains. They differ in a variety of aspects, including trust inference methodologies, complexity and accuracy. When a user or a group of users wants to join a decentralized system, a natural question is what trust function should be used for trust-related decisions. Instead of always designing their own trust functions from scratch, it will be beneficial if existing trust functions can be reused. To make a rational choice, it is necessary to have a classification scheme for trust functions so that their advantages and disadvantages for the problem at hand can be systematically analyzed.

Such a classification scheme is the focus of this paper. In section 2, we present a general framework for reputation-based trust. The framework not only includes relevant information for trust decisions, but also models the social interaction structure of a decentralized system. The framework thus can accommodate many of the existing trust models in the literature. Based on this framework, we propose a classification scheme for trust functions in section 3. The scheme has four dimensions, namely the nature of trust (subjective trust vs. objective trust), information collection (complete information vs. localized information), trust decisions (rank-based vs. threshold-based) and inputs of trust functions (opinion-based vs. transaction-based). We discuss the impact of each dimension on the properties of trust functions, and review some representative trust functions in the literature, based on the proposed classification scheme in section 4.

This work is not the first trying to classify trust functions. For example, Ziegler et al. [15] also provides a classification scheme of trust metrics. Some of their classification dimensions overlap with ours, while some others concern about other aspects of a trust model, such as where trust computation takes place. We will compare their scheme with ours when appropriate in this paper.

2 A Framework for Reputation-Based Trust

We assume that entities in a decentralized environment interact with each other through transactions. Transactions are not limited to monetary interactions. They also include activities such as retrieving information from a website, downloading files from a ftp server, etc. We further assume a transaction is uni-directional, i.e., given a transaction, there is a clear distinction between a service provider (server) and a service consumer (client). We introduce the following notion to facilitate our discussion.

Trustworthiness An entity's trustworthiness is an indicator of the quality of the entity's services. It is often used to predicate the future behavior of the entity. Intuitively, if an entity is trustworthy, it is likely that the entity will provide good services in future transactions. In most trust models [4, 6, 11], the domain of trustworthiness is assumed to be $[0, 1]$.

Feedback A piece of feedback is a statement issued by the client about the quality of a service provided by a server in a single transaction. In general, feedback may be multi-dimensional, reflecting the client's evaluation on a variety of aspects of a service, e.g., price, product quality and timeliness of delivery. For simplicity, we assume in this paper that feedback is one-dimensional, and is also from the domain $[0, 1]$.

Opinion An opinion is a user's general impression about a server. It is derived from its feedback on all the transactions that are conducted with the server. We also assume an opinion is one-dimensional and is from the domain $[0, 1]$.

In some trust models [3, 7], if an entity A has direct interactions with another entity B, then A's opinion of B is treated as B's trustworthiness from A's point of view. In some other models [4, 8, 11], A also needs to consider the information of B from other entities to infer the overall trustworthiness of B. Thus, in this paper we distinguish opinions from trustworthiness.

Source and Destination of Trust Evaluation If an entity A is interested in knowing the trustworthiness of another entity B, then we say A and B are the source and destination of a trust evaluation respectively.

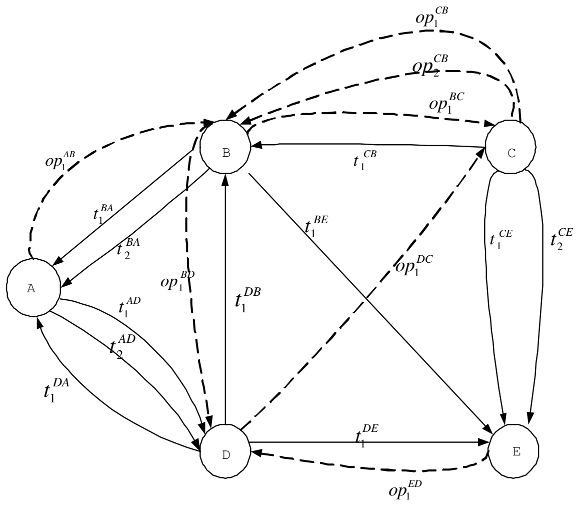

Fig. 1. An example trust graph

We model the relevant information for trust decisions as a directed multigraph $G(V, E)$, where V is a set of vertices and E is a set of labeled edges. V represents all the entities in an open system. They may provide service to others or request services from others, or both. We also call G a *trust graph* over V.

There are two types of edges: transaction edges and opinion edges. A transaction edge e from vertices A to B represent a transaction where B provides a service to A. The label of e contains the basic information of the transaction, e.g., the time of the transaction, the transaction type (e.g., file downloading and movie reviews), and the quantity of the services (e.g., total size of downloaded files and the number of ordered products). In particular, the label contains A's feedback on the transaction. There may be multiple transaction edges from A to B as more than one transaction may be conducted between them.

Similarly, an opinion edge from A to B represents A's opinion on B's service. There may also be multiple opinion edges from A to B, each of which represents A's opinion on a certain type of services provided by B. For example, A may have a good experience with B's reviews on movies but not those on small appliances. Figure 1 shows an example trust graph, where solid edges are transaction edges and dashed edges are opinion edges. Note that a trust graph may not be weakly connected since some users may not have any interactions with others, e.g., those just joining the system.

Let V be the set of entities in an open environment and $\mathcal{G}$ be the set of all the possible trust graphs over V. A *trust function* is a mapping $F : \mathcal{G} \times V \times V \rightarrow [0, 1]$. Intuitively, let A and B be the source and destination of a trust evaluation. Then $F(G, A, B)$ reports B's trustworthiness from A's point of view.

Many trust functions $[7, 11, 3]$ assume that trust is transitive, i.e., if A trusts B and B trusts C, then A may also trust C. Thus transactions or opinions of the same type are used in trust functions. Some others [14] adopt a referral model, which makes a distinction between service recommenders and service providers. In other words, a good service provider may not offer useful information in recommending other service providers (e.g., due to competitions). Our framework accommodates both approaches since referrals can be modeled as a special type of transactions.

3 Trust Function Classification Scheme

Based on the above trust framework, we propose a classification scheme for trust functions. The scheme is composed of the following four dimensions.

3.1 Subjective Trust vs. Objective Trust

An entity's trustworthiness is often related to the quality of services it provides to others. If the quality of a service can be objectively measured, then an entity's trustworthiness for that service is called *objective trust*. For example, suppose a website provides specification information of automobiles. The quality (or accuracy) of such information can be indisputably checked against the official data released by manufacturers.

For some other services, their quality cannot be objectively measured. For example, given a movie review from a website, different people may have different opinions about its quality. It largely depends on each individual's taste and other subjective factors. In this situation, it is only meaningful to discuss the trustworthiness of the entity from a specific source's point of view. We call such trust *subjective trust*.

Intuitively, if the quality of a service can be objectively measured, then an entity's trustworthiness for that service reflects some intrinsic property of that entity, which should be independent of the source of the trust evaluation. For example, the accuracy of automobile specification information provided by a website should be the same to everybody. Formally, given a trust function F, if $F(G, A, C) = F(G, B, C)$ for any trust graph G, and any entities A, B and C, then we say F is suitable for objective trust evaluation, or F is an *objective trust function*.

An entity's subjective trust, however, may vary greatly when different sources of trust evaluation are considered. Thus, given a trust function F and an entity C, if there exist a trust graph G and entities A and B, such that $F(G, A, C) \neq F(G, B, C)$, then we say F is suitable for subjective trust evaluation, or F is a *subjective trust function*.

In general, subjective trust functions and objective trust functions are not comparable. They are suitable for different types of trust applications.

This classification dimension is similar to the distinction between global trust and local trust proposed in [15].

3.2 Transaction-Based vs. Opinion-Based

Some trust models rely on the information of individual transactions to infer an entity's trustworthiness, while others only request opinion information. To reflect the difference, we have the following definition. Given a trust graph $G(V, E)$, let $G_T = (V, E_T)$ where $E_T = \{e \mid e \in E \; and \; e \; is \; a \; transaction \; edge\}$. We call G_T the *transaction trust graph* of G, denoted. The *opinion trust graph* of G, denoted G_O, is similarly defined. Let F be a trust function. If $F(G, A, B) = F(G_T, A, B)$ for all G, A and B, then we say F is a *transaction-based* trust function. Similarly, if $F(G, A, B) = F(G_O, A, B)$ for all G, A and B, then F is *opinion-based*.

Note that a transaction-based trust function does not always require detailed information of every transaction. Instead, some may only rely on some statistic information, e.g., total number of positive/negative transactions and the number of transactions during a certain period of time. But in general, it requires more information than opinion-based trust functions, inflicting a higher communication cost.

Since opinion-trust functions give each entity the autonomy to form their own opinions and conceal detailed transaction information, they are more privacy-friendly. Due to the same reason, however, opinion-based trust functions may be more easily influenced by malicious users. For example, Alice may have had a lot of transactions with Cathy, and forms an opinion on Cathy reasonably by using the percentage of positive transactions among all her transactions with Cathy. Another entity Bob, in the extreme case, may have an opinion on Cathy of value 0, even though he has never interacted with Cathy. By simply looking at these two opinions, it is hard to tell which opinion is more valuable for one's trust decisions[1].

3.3 Complete Information vs. Localized Information

Trust functions can also be classified according to the way information is collected. Some trust functions [12, 8] assume that every entity has the same access to all the transaction or opinion information. In other words, to apply a trust function, a complete transaction or opinion graph is a must. We call such trust functions *global trust functions*.

Another approach is to adopt a localized search process. Typically, it is assumed that an entity has several neighbors, who may or may not have interactions with the entity before. If Alice wants to evaluate Bob's trustworthiness, she will broadcast to her neighbors the requests for Bob's transaction/opinion information. This process continues until her neighbors have returned sufficient information for Alice to make a trust decision. To achieve better performance, information collection is usually a controlled "flooding" process. Therefore, the trust function is applied on a subgraph of the complete trust graph. Since each entity chooses their neighbors freely, different trust evaluation sources may construct different subgraphs. Thus we call trust functions of this kind *localized trust functions*. Intuitively, for a localized trust function, each entity typically has access to different information. A localized trust function is thus also subjective [15].

In general, localized trust functions scale better and are more suitable for decentralized environments. They also avoid the privacy concerns which may arise with the use of global trust functions. However, global trust functions tend to produce better results due to its access to a complete trust graph.

[1] Of course, a malicious users may issue biased feedback. But feedback has to be associated with a transaction trail, which can be signed by both the server and the client [9], or created by a trusted third party, e.g., in ebay. Thus, it is harder to influence a transaction-based trust function than an opinion-based trust function.

3.4 Rank-Based vs. Threshold-Based

Once a trust function returns the trustworthiness of a server, should we request services from that server? In other words, how should we utilize one's trustworthiness to make a trust decision? This question, in fact, relies on the nature of the output of a trust function.

For most trust functions, its returned trustworthiness can be interpreted as an approximation of some of the properties of a server. For example, if the trustworthiness of an automobile website is 0.8, we may think that approximately 80% of the information provided by the website is accurate. For such trust functions, it is appropriate to pre-define a threshold of trustworthiness to make trust decisions. For example, if a website's trustworthiness is over 0.9, then we trust information from that website. Thus, we call such functions *threshold-based*.

In some other trust functions, the calculated trustworthiness of a single entity alone does not convey much information. It becomes meaningful only when it is compared with the trustworthiness of other entities. In some sense, such trust functions return the relative ranking of an entity. We call such functions *rank-based*.

These two kinds of trust functions are suitable for different application requirements. If we would like to have certain quality assurance, then threshold-based trust functions are ideal. For rank-based trust functions, even if a server has a high ranking, it does not necessary mean that its service is of high quality. On the other hand, if we would like to know whether the quality of a server is among the top 10% of all service providers, then rank-based trust functions is more appropriate. Of course, we can also obtain the ranking information if we use a threshold-based trust function to infer every entity's trustworthiness. But it would be very expensive for large-scale decentralized systems like the Semantic Web.

The above four dimensions provide for easy matching between problems and trust functions. For each of these categories, it should be clear from the situation being addressed what sort of a trust function is needed. Questions about whether the services being measured are objective or subjective, whether every transaction or just overall opinion are important, whether complete information is available or local information should be used, and whether we care about an absolute threshold or a relative rank should be fairly easy to answer in most situations. We can, therefore, use these categories to conveniently identify which trust functions would be applicable for a given situation, significantly narrowing the process of choosing a trust function.

4 Classification of Existing Trust Functions

With the trust function classification scheme at hand, we now classify some existing works. A summary of the classification is given in table 1.

4.1 NICE

Lee et al. [7] proposed a trust inference scheme for NICE, a platform for Internet cooperative applications. In their scheme, after each transaction, the client A signs a cookie stating

functions	subj./obj.	trans./opinion	complete/localized	rank/thresh.
NICE	subj.	trans.	localized	thresh.
Evidence-based model	subj.	opinion	localized	thresh.
PeerTrust	obj.	trans.	complete	thresh.
EigenTrust	obj.	trans.	complete	rank
Reputation Inference	subj.	opinion	localized	thresh.
Trust for the Semantic Web	subj.	opinion	localized	thresh.
Heuristics Complaint Checking	obj.	trans.	complete	rank

Table 1. Classification of trust functions

the quality of the transaction to the server B, which later can be used by B to prove its trustworthiness to others, e.g., C. If B does not have cookies issued by C directly, it may consult its neighbors to collect chains of cookies from C to B. Such a set of chains of cookies indeed forms a subgraph G'_T of the transaction trust graph G_T. The final trustworthiness of B from C's point of view is obtained by calculating either the strongest path or the weighted sum of strongest disjoint paths in G'_T between C and B.

NICE is designed to help entities form cooperative groups. For the same service provider, entities from different groups may have different evaluations of its service. So the proposed scheme is a subjective trust function. Clearly, the scheme is transaction-based as cookies are collected before a subgraph of G_T can be formed.

4.2 Evidence-Based Model

Yu and Singh [14] proposed a distributed reputation management model which views feedback as evidence of the trustworthiness of a service provider. Trust evaluation is thus modeled as an evidence collection process.

In [14], a feedback can be either positive, negative or neutral. Given a set of feedback from user A to B, A's opinion on B is modeled as a triple $(m(T), m(T, -T), m(-T))$, where $m(T)$, $m(T, -T)$ and $m(-T)$ are the percentages of positive, neutral and negative transactions respectively. If A does not have direct interactions with B, A has to search for B's evidence through the help of her neighbors. This is where a subgraph of the trust graph G is formed. Then A will collect opinions from those who have direct interaction with B and form her own judgement of B's trustworthiness.

Similar to NICE, the evidence-based model is suitable for subjective trust evaluation. But it is opinion-based. No detailed transaction information is exchanged between entities.

4.3 PeerTrust

Xiong and Liu [12, 13] proposed PeerTrust, a reputation trust model for peer-to-peer systems. Using the P-Grid [1] data storage structure, PeerTrust assumes that every user is able to retrieve all the transaction information in a system. User i's trustworthiness is the normalized number of satisfactory services over the total number of transactions in which the user has taken part. It is further weighted by the trustworthiness of the feedback issuers and the quantity of each transaction.

Since all the users can get the complete information about any other user's transactions, they will end up with a common view of the trustworthiness of any user. PeerTrust is ideal for objective trust evaluation.

4.4 EigenRep

EigenRep [4] is a rank-based trust function. In EigenRep, system-wide complete transaction information is obtained via the CAN [10] data structure. The number of satisfactory and unsatisfactory transactions between each pair of entities is collected and used to construct a matrix. One special property of the matrix is that the entries in each row will add up to 1. The matrix will be repetitively multiplied with an initial vector, until it converges. The initial vector is a pre-defined system parameter which contains the default trustworthiness of each user. Each entry of the converged trust vector represents a user's final trustworthiness. Every user will get the same trust vector, since the matrix and the computation process is the same for all users. So this model would be a good candidate for objective trust evaluation.

Kamvar et al. [4] showed that the final trustworthiness of all users will also adds up to 1. Thus, only given the trustworthiness t_i of a particular user, it does not have any indication of the quality of that user's service. The output trust value just shows the comparative relations between users,i.e., given the global trust vector, we only can tell whether user i is more trustworthy than user j.

4.5 The Reputation Inference Algorithm

Compared with other trust functions, Golbeck and Hendler [3] propose a totally localized approach. In their model, each entity has several trusted neighbors. To infer the trustworthiness of a user B, A only polls her neighbors about their trust on B. This process continues recursively until it reaches entities who have interactions with B so that they can directly evaluate B's trustworthiness. The trustworthiness returned by each entity is either 1 (trusted) or 0 (untrusted). Once A receives all her neighbor's evaluation of B's trust worthiness, it simply takes a vote and decide whether it should trust B.

The trust graph is implicitly explored through recursive trust evaluation, which offers a simple protocol. On the other hand, since an entity does not have a relatively global view of the system and no transaction information is ever collected, it is critical to choose trusted neighbors. If one or more neighbors become malicious, an entity's trust decision can be greatly influenced.

4.6 The Trust Interpretation Model for the Semantics Web

Richardson et al. [11] discussed reputation-based trust management for the Semantic Web. The problem they considered is as follows. Suppose every user has local opinions on the rest of users in a system, which can form an opinion vector. How to get a global trust matrix such that each entry t_{ij} specifies user j's trustworthiness from user i's point of view, when considering other users' opinion. Two approaches are proposed based on different modeling of the problem.

The first approach assumes that t_{ij} in the global trust matrix only depends on the paths from i to j in the opinion trust graph G_O. Two auxiliary functions are defined to derive the global trust matrix from users' local opinions. Given a path from i to j, a concatenation function calculates i's indirect opinion on j along the path. An aggregation function calculates i's aggregated opinion on j over a set of paths from i to j. The global trust matrix can be derived through a sequence of matrix multiplication from users' local opinion vectors, by using the above two functions.

In the second approach, the same global trust matrix is built as in the first approach. But the problem is modeled as a random walk on a Markov chain, similar to the Page Ranking model in search engines [5]. t_{ij} in the matrix is interpreted as the probability that user i's surfer arrives at user j. Further, the second approach introduce a parameter λ to weight users' local opinion and the global trust matrix. Thus it allows users to maintain their own opinions which can be factored into their final trust decisions.

Though the trust function itself relies on complete information of the opinion trust graph, Richardson et al. designed an algorithm that only requires iterative information exchange between adjacent users in the trust graph. Further, this approach is opinion-based, which further reduces information exchange. Therefore, the global trust matrix can be derived efficiently.

4.7 Heuristics Complaint Checking

Aberer and Despotovic [2] considered the trust inference problem when users only issue negative feedback. Similar to PeerTrust, their approach also utilizes the P-Grid storage structure so that complaints issued by any entity can be retrieved. The trustworthiness of a user i can be calculated directly, based on the user's transaction history and compliants it receives. Aberer and Despotovic proposed a trust decision inequality to determine whether an entity should be trusted or not. The inequality does not provide a parameter explicitly for user customization. As long as two entities are both trusted, it cannot tell which one is more trustworthy. It can be viewed as a very course-grained rank-based trust function. Also, since every entity has access to the same information and uses the same trust function, they will always reach the same trust decision on a given user.

5 Discussion

Credentials are another important means to establish trust in decentralized systems. Credential-based trust and reputation-based trust are complementary to each other. On one hand, credential-based trust policies help avoid a large class of unreliable or irrelevant entities for a particular application, and make reputation collection more accurate and efficient. This is especially desirable for large-scale open systems. On the other hand, reputation-based trust is particularly useful when some of the authorities along a credential chain cannot be unconditionally trusted and continual history-based trust evaluation is required. It is desirable to design a comprehensive trust model which combines the strength of both. One key issue is to design a policy model that seamlessly integrates constraints on both credentials and reputations into trust policies.

6 Conclusions and Future Work

Trust functions are a central component to reputation-based trust management. An appropriate classification scheme of trust functions will help us systematically analyze their properties and choose the right one for a particular application. In this paper, we propose such a classification scheme based on a generic trust framework. We further review some representative trust functions in the literature according to the classification scheme.

As a part of our future work, we would like to investigate the application of reputation-based trust models in wireless network routings and distributed system load balancing.

References

1. K. Aberer. P-Grid: A self-organizing access structure for p2p information systems. In *Cooperative Information Systems, 9th International Conference (CoopIS)*, 2001.
2. K. Aberer and Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM)*, 2001.
3. J. Golbeck and J. Hendler. Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks. In *International Conference on Knowledge Engineering and knowledge Management (EKAW)*, Northamptonshire, UK, Oct. 2004.
4. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. EigenRep: Reputation Management in P2P Networks. In *World-Wide Web Conference*, 2003.
5. R. Lawrence, B. Sergey, M. Rajeev, and W. Terry. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Department of Computer Science, Stanford University, 1998.
6. S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative Peer Groups in NICE. In *INFOCOM*, 2003.
7. S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative Peer Groups in NICE. In *IEEE Infocom*, San Francisco, CA, Apr. 2003.
8. L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model of Trust and Reputation. In *35th Hawaii International Conference on System Science*, 2002.
9. J. Peha. Making Electornic Transactions Auditable and Private. In *Internet Society (ISOC) INET*, San Jose, CA, June 1999.
10. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, 2001.
11. M. Richardson, R. Agrawal, and P. Domingos. Trust Management for the Semantic Web. In *Proceedings of the Second International Semantic Web Conference*, 2003.
12. L. Xiong and L. Liu. Building Trust in Decentralized Peer-to-Peer Electronic Communities. In *The 5th International Conference on Electronic Commerce Research. (ICECR)*, 2002.
13. L. Xiong and L. Liu. A reputation based trust model for peer-to-peer ecommerce communities. In *IEEE International Conference on E-Commerce (CEC)*, 2003.
14. B. Yu and M. P. Singh. An Evidential Model of Distributed Reputation Management. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2002.
15. C.-N. Ziegler and G. Lausen. Spreading Activation Models for Trust Propagation. In *IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE '04)*, 2004.