Trust Representation and Aggregation in a Distributed Agent System*

Yonghong Wang and **Munindar P. Singh**
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{ywang4, singh}@ncsu.edu

Abstract

This paper considers a distributed system of software agents who cooperate in helping their users to find services, provided by different agents. The agents need to ensure that the service providers they select are trustworthy. Because the agents are autonomous and there is no central trusted authority, the agents help each other determine the trustworthiness of the service providers they are interested in. This help is rendered via a series of referrals to other agents, culminating in zero or more trustworthy service providers being identified.

A trust network is a multiagent system where each agent potentially rates the trustworthiness of another agent. This paper develops a formal treatment of trust networks. At the base is a recently proposed representation of trust via a *probability certainty* distribution. The main contribution of this paper is the definition of two operators, *concatenation* and *aggregation*, using which trust ratings can be combined in a trust network. This paper motivates and establishes some important properties regarding these operators, thereby ensuring that trust can be combined correctly. Further, it shows that effects of malicious agents, who give incorrect information, are limited.

Introduction

In electronic markets, due to the high uncertainty about quality and reliability of the products and services offered by others, it is crucial for agents to compute the trustworthiness of the other agents before initiating any service request. Similar considerations apply in Web-based information systems, in general: agents must be able to compute how much trust to place in others with whom they might have had no prior experience.

The mechanisms that support finding trust estimations are called reputation systems. There are two kinds of reputation systems: centralized and distributed. Centralized reputation systems include collaborative filtering (social information filtering) systems (Breese, Heckerman, & Kadie 1998; Resnick *et al.* 1994; Dellarocas 2004), and online reputation systems (Dellarocas 2004). Distributed reputation systems include peer-to-peer (P2P) systems (Aberer & Despotovic 2001; Xiong & Liu 2004) and referral systems (Yu & Singh 2002; 2003).

*This research was partially supported by the National Science Foundation under grant ITR-0081742.

Collaborative filtering approaches select resources based on the relationships between agents as modeled by the similarity or dissimilarity between their subjective judgments. For example, GroupLens (Resnick *et al.* 1994) helps users find netnews articles based on how their past ratings correlate with other users'. Online reputation systems (Dellarocas 2004) build the reputation for an object by aggregating beliefs from agents about a specified agent. One of the most successful examples is eBay. After each transaction, buyers and sellers can rate each other. A participant's reputation is the sum of these ratings over the last six months. The centralized approaches do not scale well and they all have a single point of failure, so they are not appropriate in a distributed system.

Peer-to-peer systems and referral systems are built for distributed environments. They enable peers to share their experiences with each other. Trust is captured by the neighborhood relation. Referral systems maintain and use trust for recommendation. Various mathematical approaches have been used to combine trust ratings in these systems. Traditional methods in combining the trust from the referrals are not efficient in dealing with malicious agents who provide biased information.

Challenges. A reputation system for an open environment should support the following features. First, agents may join or leave an open environment arbitrarily. Second, the agents need not be cooperative. An agent may give biased information on its own experience. It may also provide many witnesses whose ratings of the intended target are biased.

Contributions. This paper uses a probabilistic theory of evidence to represent the trust between agents in terms of referrals and the quality of service obtained. To support an agent computing trust in a service provider with which it has no direct interactions, this paper defines operators to determine trust by concatenating and aggregating paths from an agent to the given service provider. The underlying architecture of a distributed system is common to several approaches, and not novel to this paper. This paper formalizes some key properties of the concatenation and aggregation operators. Lastly, this paper shows how our operators mitigate the effects of deception by some agents.

Technical Framework

This section introduces the necessary mathematical tools for our approach: (1) probability certainty distributions. and (2) the construction of a path algebra.

Trust and Certainty

This discussion is summarized from (Wang & Singh 2006). Let $\langle r, s\rangle$ be a binary event with r positive outcomes and s negative outcomes.

Definition 1 *The posterior probability of a binary event after $\langle r, s\rangle$ is given by (Casella & Berger 1990, p. 298):*

$$f_{r,s}(x) = \frac{x^r(1-x)^s}{\int_0^1 x^r(1-x)^s dx}$$

The following is a motivating example for the definition of certainty adopted here. Consider randomly picking a ball from a bin that contains N balls colored white or black. Suppose p is the probability that the ball randomly picked is white. If we have no knowledge about how many white balls there are in the bin, we can't estimate p with any confidence. Thus, our certainty (notated c) is 0. If we know that exactly m balls are white, then we have perfect knowledge about the distribution. We can estimate $p = \frac{m}{N}$ with $c = 1$. However, if all we know is that at least m balls are white and at least n balls are black, then we have partial knowledge. Here $c = \frac{m+n}{N}$. The following definition reflects the above intuitions (Wang & Singh 2006).

Definition 2 *Let $c(r, s)$ be the certainty based on the binary event $\langle r, s\rangle$, then*

$$c(r, s) = \frac{1}{2}\int_0^1 |f_{r,s}(x) - 1|dx$$

where $f_{r,s}(x)$ is as in Definition 1

Instead of modeling the binary events by a pair (p, q), the probability of positive and negative outcomes, we model the binary event $\langle r, s\rangle$ by a belief (b, d, u), where b, d and u represent the probability of positive outcomes, negative outcomes, and uncertainty, respectively. Here, $b = c\frac{r+1}{r+s+2}$, $d = c\frac{s+1}{r+s+2}$, and $u = 1 - c$ where c is the certainty based on the binary event. The certainty lies in $[0, 1]$, with $c = 1$ and $c = 0$ indicating perfect knowledge and ignorance, respectively.

Definition 3 *Define the evidence space*

$$E = \{(r, s)|r > 0, s > 0\}$$

and the belief space

$$B = \{(b, d, u)|b > 0, d > 0, u > 0, b + d + u = 1\}$$

Let $Z = (B, D, U)$ be a transformation from E to B such that $Z(r, s) = (B(r, s), D(r, s), U(r, s))$

$$\begin{array}{rcl} B(r, s) & = & c\frac{r+1}{r+s+2} \\ D(r, s) & = & c\frac{s+1}{r+s+2} \\ U(r, s) & = & 1 - c \end{array} \quad (1)$$

where c is defined in Definition 2.

Definition 2 also ensures that the transformation Z is a bijection between E and B (Wang & Singh 2006). Let $Z^{-1} : B \mapsto E$ be the inverse transformation, define

$$Z^{-1} = (R, S) \quad (2)$$

where $R : B \mapsto \Re^+$ and $S : B \mapsto \Re^+$. For any $(r, s) \in E$, we have $Z^{-1}(Z(r, s)) = (r, s)$

Path Algebra

We use the idea of *path algebra* from the generalized transitive closure literature (Richardson, Agrawal, & Domingos 2003) for our computation of the merged trust. Path algebra provides a means to talk about trust propagation and aggregation. Below we formalize it in our terms.

Definition 4 *Let $G = (V, E)$ be a graph. For each edge $e(i, j) \in E$, where i is the source and j is the destination, define the label associated with the edge as $L(e(i, j)) \in L$. Here L is the range of the labels, for example, R for real numbers, R^2 for vectors. The edge $e(i, j)$ together with the label $L(e(i, j))$ is called an* edge tuple.

Definition 5 *A path from node i to node j is denoted by $P(i, j)$, which is the concatenation of an ordered set of labeled edges $e_k(v_k, v_{k+1})$, for $k = 1, \cdots, n$, where $i = v_1$ and $j = v_{n+1}$. Define the label associated with the path as $L(P(i, j)) \in L$. The label associated with the path is computed as a function of the labels associated with the edges in the path through the concatenation operator $\otimes$. The path together with its label is called a* path tuple.

Definition 6 *A* path-set *from node i to node j, denoted by $\psi(i, j)$, is the set of all paths from i to j in the given graph $G = (V, E)$. $\psi(i, j) = \{P_k(i, j)\}$, for $k = 1, \cdots, m$. Define the label for the* path-set *as $L(\psi(i, j)) \in L$, which should be computed as a function of all the labels of the paths in $\psi(i, j)$ through an aggregation operator $\oplus$. A* path-set *together with its label is called a* path-set *tuple.*

Definition 7 *Concatenation operator $\otimes : L \times L \mapsto L$. This accommodates the propagation of trust along a path.*

Definition 8 *Aggregation operator $\oplus : L \times L \mapsto L$. This accommodates combining trust from different paths.*

Propagating Trust

Say an agent A_r does not have adequate evidence to determine whether to trust an agent A_g. A_r could obtain evidence from other agents, i.e., *witnesses*, who have experience with A_g. The mechanism of finding witnesses is irrelevant here: a possible approach is based on referrals (Yu & Singh 2002). A_r can evaluate the trustworthiness of A_g by combining the trust in A_g placed by those witnesses. In general, A_r would not have direct experience with the witnesses, and would need witnesses for them, and so on: so this would be a case of propagating trust.

We use the term "reference" to generalize over a referral or another means for an agent to indicates its level of trust in another agent. A reference could correspond to a URI on a web page or even to a reference in a specialized application or vocabulary such as Friend of a Friend (FOAF).

Definition 9 *A reference r to A_j by A_i is represented by $\langle A_i, A_j, M_{ij}\rangle$, where M_{ij} represents the trust in A_j placed by A_i.*

Definition 10 *A trust network TN(A_r, A_g, A, R, W) is an acyclic directed graph rooted at A_r, where A is a finite set of agents $\{A_1, A_2, \ldots, A_N\}$, R is a finite set of references $\{r_1, r_2, \ldots, r_n\}$, and W is a set of witnesses for A_g.*

A referral network, which is a kind of a trust network, can be constructed as described in (Yu & Singh 2002).

Merge and Combine Trust Ratings

Given a trust network, we adapt the two operators used in the path algebra to merge the trust.

Assume agent A has a trust M_1 in agent B's references and B has a trust M_2 in agent C. Suppose $M_1 = (b_1, d_1, u_1)$ and $M_2 = (b_2, d_2, u_2)$. As proposed by Jøsang (Jøsang 1998), A disbelieves B's references means that A thinks B is uncertain about agent C's trustworthiness. We define A's trust in C due to the reference from B to be $M = M_1 \otimes M_2$. Here $\otimes$ is the *concatenation operator*.

Definition 11 *Concatenation operator $\otimes$ (Jøsang 1998). Suppose $M_1 = (b_1, d_1, u_1)$ and $M_2 = (b_2, d_2, u_2)$ are two belief functions, we define $M = M_1 \otimes M_2 = (b, d, u)$ as:*

$$\begin{array}{lcl} b & = & b_1 b_2 \\ d & = & b_1 d_2 \\ u & = & 1 - b_1 b_2 - b_1 d_2 \end{array} \quad (3)$$

Assume agents A and B have trust M_1 and M_2, respectively, in A_g. The combined trust of A and B in A_g is captured via the *aggregation operator* $\oplus$, as in $M_1 \oplus M_2$. Suppose A has r_1 positive experiences and s_1 negative experiences with A_g, and B has r_2 positive experiences and s_2 negative experiences with A_g. Then the combined evidence will be $r_1 + r_2$ positive experiences and $s_1 + s_2$ negative experiences with A_g.

In order to combine the beliefs M_1 and M_2, we first transform the two beliefs to two bodies of evidence in the evidence space, combine them, then map the combined evidence back to belief space.

Definition 12 *Aggregation operator $\oplus$. Let $Z = (B, D, U)$ be the transformation from evidence space to belief space as defined in Definition 3 and $Z^{-1} = (R, S)$ is the inverse of Z. Suppose $M_1 = (b_1, d_1, u_1)$ and $M_2 = (b_2, d_2, u_2)$. Then $M_1 \oplus M_2 = M = (b, d, u)$ where*

$$\begin{array}{lcl} b & = & B(r_1 + r_2, s_1 + s_2) \\ d & = & D(r_1 + r_2, s_1 + s_2) \\ u & = & U(r_1 + r_2, s_1 + s_2) \\ r_1 & = & R(b_1, d_1, u_1), r_2 = R(b_2, d_2, u_2) \\ s_1 & = & S(b_1, d_1, u_1), s_2 = S(b_2, d_2, u_2) \end{array} \quad (4)$$

For a given trust network TN(A_r, A_g, A, R, W), we combine beliefs as following. For any agent $A_i \in A$, suppose $\{B_1, B_2, \ldots, B_m\}$ are the neighbors of A_i. Suppose the trust ratings that A_i assigns to $B_1, \ldots, B_m$ are $M_1, M_2, \ldots, M_m$. Suppose that all the neighbors have already obtained their trust ratings in A_g, let these be $M'_1, M'_2, \ldots, M'_m$. Then we obtain the trust of A_i in A_g, M, by

$$M = (M_1 \otimes M'_1) \oplus (M_2 \otimes M'_2) \oplus \cdots \oplus (M_m \otimes M'_m)$$

If the neighbor has not obtained the trust in A_g, we can run the algorithm recursively to obtain the trust from merging and combining the trust from the neighbor's neighbors, since all the leaves in the trust network are the witnesses who have their trust values in A_g computed from their direct interactions with A_g. So the trust ratings can be merged in a bottom up fashion, from the leaves of the trust network up to its root A_r.

Properties of the Operators

In order to combine the trust ratings in a meaningful way and to deal with deception caused by malicious agents, we desire certain properties of our approach. We motivate and prove that the concatenation operator $\otimes$ and aggregation operator $\oplus$ satisfy the following properties. The proofs of the theorems are given in the appendix.

Theorem 1 *The concatenation operator $\otimes$ is associative.*

This property enables us to merge the trust in a bottom up fashion. For example, consider the path $A_r \to B \to C \to A_g$, (see Figure 1). Suppose it is the only path from A_r to A_g in the trust network. The trust of A_r in B is M_1, the trust of B in C is M_2 and the trust of C in A_g is M_3. If we merge M_1 and M_2, we will get the trust of A_r in C, which is $M_1 \otimes M_2$, when we merge it with M_3, we get the trust of A_r in A_g, which is $(M_1 \otimes M_2) \otimes M_3$. If we do it bottom up, then we merge M_2 and M_3 and get the trust of B in A_g and then merge it with M_1 and get the trust of A_r in A_g which is $M_1 \otimes (M_2 \otimes M_3)$. Certainly we expect these two results are the same, that is $(M_1 \otimes M_2) \otimes M_3 = M_1 \otimes (M_2 \otimes M_3)$. This means $\otimes$ should be associative.

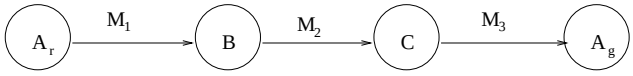

Figure 1: Path in a trust network

Theorem 2 *The aggregation operator $\oplus$ is associative.*

This is required to aggregate trust ratings across independent paths. Suppose A_r has three neighbors B, C and D who are the only witnesses of A_g, (see Figure 2). The trust ratings of B, C and D in A_g are M'_1, M'_2 and M'_3, respectively. Suppose the trust ratings of A_r in B, C and D are M_1, M_2 and M_3, respectively. Then according to the algorithm to find the trust of A_r to A_g. We first concatenate the trust ratings, $M_1 \otimes M'_1$, $M_2 \otimes M'_2$ and $M_3 \otimes M'_3$. Then we aggregate them. There are different ways to aggregate them. We can aggregate $M_1 \otimes M'_1$ with $M_2 \otimes M'_2$ and then with $M_3 \otimes M'_3$, or we aggregate $M_2 \otimes M'_2$ with $M_3 \otimes M'_3$, and then with $M_1 \otimes M'_1$. These should be the same. So we require $((M_1 \otimes M'_1) \oplus (M_2 \otimes M'_2)) \oplus (M_3 \otimes M'_3) = (M_1 \otimes M'_1) \oplus ((M_2 \otimes M'_2) \oplus (M_3 \otimes M'_3))$. That is, $\oplus$ should be associative.

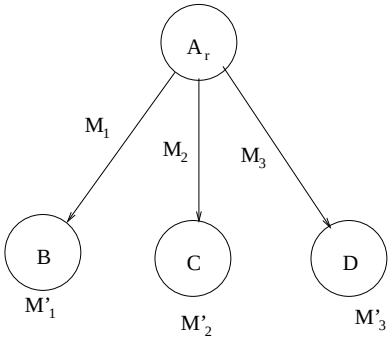

Figure 2: Associativity of $\oplus$

Theorem 3 *The aggregation operator $\oplus$ is commutative.*

This is required to aggregate trust ratings across paths that are not ordered. For example, suppose A_r has two neighbors B and C, who are the only witnesses. The trust ratings of B and C in A_g are M_1' and M_2'. (see Figure 3). Suppose the trust ratings of A_r in B, and C are M_1, and M_2, respectively. After concatenating the trust ratings, we obtain $M_1 \otimes M_1'$, and $M_2 \otimes M_2'$. We have two ways to aggregate them: $(M_1 \otimes M_1') \oplus (M_2 \otimes M_2')$, or $(M_1 \otimes M_1') \oplus (M_2 \otimes M_2')$, these should be the same. That is, $(M_1 \otimes M_1') \oplus (M_2 \otimes M_2') = (M_1 \otimes M_1') \oplus (M_2 \otimes M_2')$. So $\oplus$ should be commutative.

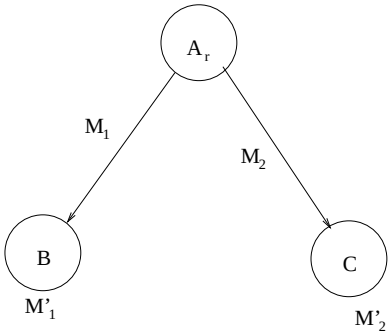

Figure 3: Commutativity of $\oplus$

Concatenation operator $\otimes$ should not distribute over the aggregation operator $\oplus$. If $\otimes$ were to distribute over the aggregation operator $\oplus$, along with the properties mentioned above, then the *path-algebra* defined will be a *well-formed decomposable path problem*. Surprisingly, this property is not desirable. The reason is as follows. Let's consider two cases, as shown in Figure 4.

Case 1. Suppose A_r has a neighbor B, and it is the only neighbor of A_r. B has referred two witnesses C and D, which are the only witnesses to A_g. Suppose the trust of A_r in B is M_1, the trust ratings of B in C, and D are M_4 and M_5, respectively, and the trust ratings of C and D in A_g are M_6 and M_7, respectively. Then according to our algorithm, the merged trust of A_r to A_g is $M_1 \otimes (M_2 \oplus M_3)$, where $M_2 = M_4 \otimes M_6$ and $M_3 = M_5 \otimes M_7$.

Case 2. Suppose A_r has two neighbors B and C. B referred D and C referred E. Let D and E be the only two witnesses of A_g. Suppose the trust ratings of A_r in B and C is the same, namely, M_1. Suppose the trust of B in D is M_4, and the trust of C in E is M_5. The trust ratings of D and E in A_g are M_6 and M_7, respectively. Then according to our algorithm, the merged trust of A_r to A_g is $(M_1 \otimes M_2) \oplus (M_1 \otimes M_3)$, where $M_2 = M_4 \otimes M_6$ and $M_3 = M_5 \otimes M_7$.

In Case 1, the two witnesses are referred by the same agent, but in Case 2 the two witnesses are referred by two different agents. So we expect they are different.

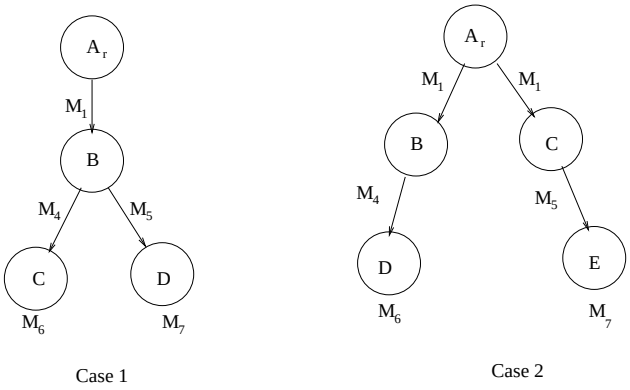

Figure 4: Nondistributivity of $\otimes$ over $\oplus$

Theorem 4 *The concatenation operator $\otimes$ does not distribute over the aggregation operator $\oplus$.*

The impact of malicious agents should be limited.

Theorem 5 *Let $Z = (B, D, U)$ and $Z^{-1} = (R, S)$ be defined as in Definitions 2 and 3. Let $M_i = (b_i, d_i, u_i), 0 \leq i \leq n$. Let $M = M_0 \otimes (M_1 \oplus M_2 \oplus \cdots \oplus M_n)$, and $M' = (\frac{b_0}{2}, \frac{b_0}{2}, 1 - b_0)$. We have*

$$R(M) + S(M) \leq 2R(M') \tag{5}$$

The theorem is interpreted as follows. M' is the trust with certainty b_0. The binary event corresponding to M' has an equal number of positive outcomes and negative outcomes, which is $R(M')$. By Theorem 2 in (Wang & Singh 2006), when the total number of outcomes is fixed, the certainty is minimized when the number of positive outcomes equals the number of negative outcomes. Since the certainty of M is less than b_0, in effect the binary event corresponding to M has fewer total number of outcomes than the binary event corresponding to M'.

Suppose the malicious agent B recommends n bad witnesses to A. Let $M_i = (b_i, d_i, u_i), 1 \leq i \leq n$ be the trust in A_g placed by those witnesses. M_0 is the trust in B placed by A. Theorem 5 shows that the effect of those bad witnesses through the recommendation of B is less than a body of evidence, which has a total number of $2R(M')$ experiences.

For example, suppose A's trust in B is (b_0, d_0, u_0). Consider the following numerical examples of how certainty

propagates. Recall that b_0 is the weight of the belief of A in B's trustworthiness.

Case 1. B lies to A that he knows 100 witnesses and each witness had 1 million positive experiences with the target,

Case 2. The 100 witnesses are referred by 100 different agents in whom A places the same trust as in B.

If $b_0 = 0.25$, then in case 1, in A's view, B's recommendation is the same as of an honest agent who tells A that he only has one positive experience with the target, while in case 2, the impact is the same as of an honest agent who tells A that he has 100 positive experiences with the target.

If $b_0 = 0.7$, then in case 1, in A's view, B's recommendation is the same as of an honest agent who tells A that he only has 9.2 positive experiences with the target, while in case 2, the impact is the same as of an honest agent who tells A that he has 916 positive experiences with the target.

If $b_0 = 0.9$, then in case 1, in A's view, B's recommendation is the same as of an honest agent who tells A that he only has 46 positive experiences with the target, while in case 2, the impact is the same as of an honest agent who tells A that he has 4,600 positive experiences with the target.

If $b_0 = 0.99$, then in case 1, in A's view, B's recommendation is the same as of an honest agent who tells A that he only has 91 positive experiences with the target, while in case 2, the impact is the same as of an honest agent who tells A that he has 9,100 positive experiences with the target. From the above examples, we can readily see that our method limits the impact of a malicious agent who introduces a large number of biased witnesses.

Discussion

This paper treats reputation management as a well-defined path algebra problem. Each agent only needs to maintain the trust ratings for its neighbors and to use referral networks and path algebra to merge and combine the trust ratings and obtain the trust in the service provider with which it has no direct interactions. Our algorithm scales well, since the trust ratings are merged in a bottom up fashion.

The underlying notion of a reference associated with a trust rating is extremely general. It accommodates referrals, neighborhood relationships in peer-to-peer computing, symbolic references across information resources, and potentially even physical neighborhood relationships such as on certain kinds of ad hoc networks.

Literature

Richardson *et al.* (2003) have studied the trust management for the semantic web. Each user maintains trust in a small number of other users. A user's trust in any other user can be computed by using the existing web of trust recursively. Richardson *et al.* first enumerate all paths between the user and every other user who has a local based belief in a given statement, then calculate the belief associated with each path by using a predefined *concatenation function* along each path and the belief held by the final user, and those beliefs associated with all paths can be combined by using a predefined *aggregation function*. Since Richardson *et al.* use the path algebra interpretation on the whole network, it is not appropriate in a distributed network where agents in the network are autonomous, since the topology may change constantly and some agents may not be cooperative.

Yu and Singh (2002), (2003) have studied distributed reputation management in a social network of agents where the agents cooperate with each other in finding trustworthiness of the other agents. Yu and Singh build their work on referral networks. They use Dempster-Shafer theory to represent the agent's trust on the service provider. When an agent wants to find the trustworthiness of a service provider, it uses the referral network to find witnesses, and combines the beliefs of those witnesses regarding the service provider. Yu and Singh assign weights to each witness to detect and penalize deceptive agents. There are some limitations to this approach. First, a service with medium quality is treated as unknown quality, not as a known medium quality. Second, suppose some witnesses are found. Let's consider two cases. In one case, all witnesses are referred by the same agent, while in the other case, all witnesses are referred by different agents. We should expect more evidence from the second case. By assigning weights to each witness these two cases are not differentiated.

Huynh *et al.* (2004) introduced a trust model, FIRE, which has four components: interaction trust, role-based trust, witness reputation, and certified reputation. FIRE incorporates all those components to provide a combined trust. Referrals were used to obtain the witnesses, each witness is assigned a weight. But we have not seen any work on the biased information referred by the malicious acquaintances. Further, it does not scale well since a weight for each witness needs to be maintained.

Jøsang *et al.* (2003) analyzed transitive trust topologies, specified three basic topology dimensions: trust origin, trust target and trust purpose. They also described principles for recommendation, measuring and computing trust based on those topologies. Whereas detailed mathematical computation formula for computation of the combined trust has not been found in his work. Jøsang *et al.* used cryptography to provide authenticity and integrity of referrals, but key management is still a major and largely unsolved problem on the Internet today.

Conclusions

Although our concatenation and aggregation operators are intuitively obvious, and similar to what other researchers have followed, their technical definitions are different. One advantage of these technical definitions is that they are based on an approach to trust that we recently introduced (Wang & Singh 2006). That approach treats trust in terms of probabilities of probabilities, and is thus more principled than ad hoc heuristic approaches, which are prevalent in the literature. The second advantage of the definitions of operators that we propose here is that they support further intuitive properties of the propagation of trust in a network, and avoid the undesirable property of distributing concatenation over aggregation. The third advantage of our definitions is that they limit the impact of a malicious agent, who might otherwise produce false witnesses to corrupt the assignment of trust.

We are building a testbed to evaluate our work empirically.

References

Aberer, K., and Despotovic, Z. 2001. Managing trust in a peer-2-peer information system. *CIKM* 310–317.

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, 43–52.

Casella, G., and Berger, R. L. 1990. *Statistical Inference*. Duxbury Press.

Dellarocas, C. 2004. Online reputation mechanisms. In Singh, M. P., ed., *Practical Handbook of Internet Computing*. Baton Rouge: Chapman Hall & CRC Press. chapter 20.

Huynh, D.; Jennings, N. R.; and Shadbolt, N. R. 2004. Developing an integrated trust and reputation model for open multi-agent systems. In *Proceedings of the Workshop on Trust in Agent Societies at AAMAS*.

Jøsang, A.; Gray, E.; and Kinateder, M. 2003. Analysing topologies of transitive trust. In *Proceedings of the Workshop of Formal Aspects of Security and Trust (FAST)*.

Jøsang, A. 1998. A subjective metric of authentication. In *Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS)*.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. 1994. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 175–186. ACM Press.

Richardson, M.; Agrawal, R.; and Domingos, P. 2003. Trust management for the semantic Web. In *The Semantic Web: Proceedings of the 2nd International Semantic Web Conference (ISWC)*, volume 2870 of *LNCS*, 351–368. Springer-Verlag.

Wang, Y., and Singh, M. P. 2006. Trust via evidence combination: A mathematical approach based on certainty. TR 2006-11, North Carolina State University.

Xiong, L., and Liu, L. 2004. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering* 16(7):843–857.

Yu, B., and Singh, M. P. 2002. Distributed reputation management for electronic commerce. *Computational Intelligence* 18(4):535–549.

Yu, B., and Singh, M. P. 2003. Detecting deception in reputation management. In *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 73–80.

Proofs of Theorems

Proof of Theorem 1 The concatenation operator $\otimes$ is associative.

Suppose $M_1 = (b_1, d_1, u_1)$, $M_2 = (b_2, d_2, u_2)$, $M_3 = (b_3, d_3, u_3)$, $M_1 \times (M_2 \times M_3) = (b, d, u)$, and $(M_1 \times M_2) \times M_3 = (b', d', u')$, Then
$b = b_1(b_2 b_3) = b_1 b_2 b_3$ and
$b' = (b_1 b_2)b_3 = b_1 b_2 b_3$, so we have
$b = b'$
$d = b_1(b_2 d_3) = b_1 b_2 d_3$ and $d' = (b_1 b_2)d_3 = b_1 b_2 b_3$, so we have $d = d'$, and
$u = 1 - b - d = 1 - b' - d' = u'$
Which completes the proof.

Proof of Theorem 2
Aggregation ($\oplus$) is associative. Since the belief space is equivalent to the evidence space. So the aggregation operator in the belief space is equivalent to the plus operator in the evidence space by our definition, and the plus operator is associative, so the aggregation operator in the belief space is also associative.

Proof of Theorem 3
Aggregation ($\oplus$) is commutative.

Since the belief space is equivalent to the evidence space. Since the aggregation operator in the belief space is equivalent to the plus operator in the evidence space by our definition, and the plus operator is commutative, so the aggregation operator in the belief space is also commutative.

Proof of Theorem 5
Assume $M_1 \oplus M_2 \oplus \cdots \oplus M_n = (b, d, u)$. Then
$M = (b_0 b, b_0 d, 1 - b_0 b - b_0 d)$
Let the equivalent event corresponding to M be (r, s), where $r = R(M)$ and $s = S(M)$. Define $c(r, s)$ be the certainty associated with the binary events (r, s), then
$c(r, s) = b_0 b + b_0 d \leq b_0$. By Theorem 2 in (Wang & Singh 2006), we have
$c(\frac{r+s}{2}, \frac{r+s}{2}) \leq c(r, s) \leq b_0$. since $R(M') = S(M')$, so
$c(R(M'), R(M')) = \frac{b_0}{2} + \frac{b_0}{2} = b_0$. So
$c(\frac{r+s}{2}, \frac{r+s}{2}) \leq c(R(M'), R(M'))$ and by Theorem 1 in (Wang & Singh 2006), we have $\frac{r+s}{2} \leq R(M')$, that is $R(M) + S(M) \leq 2R(M')$.