Personalized EigenTrust with the Beta Distribution

Daeseon Choi, Seunghun Jin, Younho Lee, and Yongsu Park

This letter presents an enhancement of EigenTrust. Using the beta distribution, local trust values can be more correctly evaluated. Simulation shows that the proposed scheme calculates the local trust more correctly by up to 8%. For personalization, the proposed scheme provides cumulative transitive values from the local trust to the global trust with mathematically guaranteed convergence.

Keywords: Trust management, peer-to-peer network, social network, security, trust evaluation.

I. Introduction

With the advance of the Internet and high-speed wired/wireless networks, people are able to interact online much more frequently and easily than offline. It is possible to visit strangers' blogs, chat with others, and carry out important transactions with unfamiliar people over the Internet.

One of the most crucial prerequisites is trust establishment for these online transactions or services. If we can correctly evaluate another person's trustworthiness online, we can avoid the diverse and numerous frauds and dangers, such as downloading malicious programs, purchasing defective products, having inappropriate interactions, and so on.

For several years, there has been much focus on decentralized trust management, and many trust management schemes have been devised. However, most rely on heuristics, and only a few schemes are based on concrete theories. EigenTrust [1] is one of them, and it is one of the most widely used up to now.

Manuscript received Aug. 25, 2009; revised Dec. 31, 2009; accepted Jan. 18, 2010.

This work was supported by the IT R&D program of MKE/KIAT [2007-S-601-03, User Control Enhanced Digital Identity Wallet System], and also supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2007-313-D00760).

Daeseon Choi (phone: +82 42 860 1308, email: sunchoi@etri.re.kr) and Seunghun Jin (email: jinsh@etri.re.kr) are with the Software Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Younho Lee (corresponding author, email: yhlee@ynu.ac.kr) is with the Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Rep. of Korea.

Yongsu Park (email: yongsu@hanyang.ac.kr) is with the Department of Computer Science and Engineering, Hanyang University, Seoul, Rep. of Korea.

doi:10.4218/etrij.10.0209.0354

EigenTrust is a distributed protocol to compute the converged global trust values of peers by applying each peer's local trust value transitively. Unlike other models, EigenTrust relies on linear algebraic theory to guarantee obtaining the converged global trust values. Moreover, it is very simple and does not depend on complex parameters or assumptions.

Recently, personalized schemes have appeared in which each peer organizes an initial trust group and gradually extends the group and updates trust information [2]-[6] to provide personalized trust information in distributed environments. However, such schemes seem infeasible since the underlying relation graph cannot be used for general cases or [2], [3] have an exponential order of time complexity [7].

We propose an enhancement of EigenTrust. Using the beta distribution, the enhanced scheme can calculate the local trust value more correctly. We also provide the cumulative transitive values from local trust to the global trust, where each peer can calculate the combined values of his/her local trust with global ones. We mathematically prove that the combined trust values with α=1 converge to those of EigenTrust. Simulation results show that the enhance scheme can calculate local trust values more correctly than Eigentrust.

II. Enhanced EigenTrust

1. Overview of EigenTrust

EigenTrust is a distributed algorithm for each peer to compute other peers' global trust values in P2P environments. To do this, each peer first calculates local trust values of others as follows.

Assume that each peer i has already had transactions with another j, and s_{ij} is initialized to 0. For each transaction between i and j, s_{ij} is incremented by 1 if i is satisfied with the transaction with j (otherwise, decremented by 1). By using s_{ij}, i can compute the direct trust value of j as

$$c_{ij} = \max(s_{ij}, 0) / \sum_j \max(s_{ij}, 0), \quad (1)$$

where c_{ij} represents j's direct trust value from the point of view of i. We call c_{ij} the local trust value hereafter. Note that $0 \le c_{ij} \le 1$. A one-hop indirect trust value of another peer k from i's point of view, (t_{ik}) can be calculated as $t_{ik} = \Sigma_j c_{ij} c_{ik}$. If we define C to be a matrix $[c_{ij}]$, $\vec{t_l}$ is a vector containing t_{ik} for all k, and $\vec{c_l}$ is a vector containing c_{ik} for all k, then, $\vec{t_l} = \vec{c_l}\, C$. If we consider two-hop indirect trust values, that is, recommendees' recommendations, $\vec{t_l} = \vec{c_l}\, C^2$. Similarly, for n-hop indirect recommendations from the initial direct trust, $\vec{t_l} = \vec{c_l}\, C^n$. Because C is aperiodic and irreducible, as n goes towards infinity, for all i, $\vec{t_l}$ is converged to a single vector, C's eigenvector. EigenTrust regards this eigenvector as global trust values for all peers. The distributed EigenTrust algorithm has score managers, where, for each peer i, different M score managers are related, and they calculate the global trust value for i. To assign score managers, a distributed hash table (DHT) such as CAN or Chord is used (see [1]).

2. Enhancement Using Beta Distributions

Recall (1) to calculate c_{ij}, where s_{ij} means the number of satisfactory transactions subtracted by the number of unsatisfactory transactions. Assume that there are two disjoint events, x and $\sim x$. We define r and s to be the numbers of occurrences of x and $\sim x$ up to now, respectively. If we define α and β as $\alpha = r+1$ and $\beta = s+1$, the probability density function f for event x is

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)+\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}.$$

The expected value for f is $(r+1)/(r+s+2)$. In 2002, Jøsang and others devised the Beta reputation system [8] using this distribution. However, [8] does not consider the convergence of transitive trust, which makes it difficult to measure the global trust of peers.

Assume that r_{ij} and s_{ij} are the numbers of past satisfactory and unsatisfactory transactions from i to j, respectively. If we use f to calculate local trust value, $c_{ij} = (r_{ij}+1)/(r_{ij}+s_{ij}+2)$, which means the probability that peer i can expect a satisfactory result for the next transaction with j.

However, we should use normalization for the local trust value for convergence in EigenTrust. Hence, we calculate the local trust value as

$$c_{ij} = \left(\frac{r_{ij}+1}{r_{ij}+s_{ij}+2}\right) \Bigg/ \Sigma j \left(\frac{r_{ij}+1}{r_{ij}+s_{ij}+2}\right). \qquad (2)$$

3. Personalized EigenTrust

If all the peers perform transactions with others evenly, EigenTrust works very well. However, in practice, there are some closed community groups, such as university students, that EigenTrust's global trust values do not reflect their trustworthiness very well [9].

Some recent schemes [2]-[6] provide personalized trust information, where each peer has an initial trust group and then gradually extends the group and updates the trust information. However, these methods have some limitations or high computational complexities [7].

In this section, we provide personalized EigenTrust to calculate the combined value of each peer's local trust and global trust. If we provide personalization, peers in a closed community group can calculate each other's trustworthiness more accurately (even though others may underestimate those peers' trust values).

One naive method adds another variable α ($0 \le \alpha \le 1$) for combination: α · (local trust value) + $(1-\alpha)$ · (global trust value), where a similar method is described in [1]. This naive method provides a simple combined value of local and global trust because the convergence speed of EigenTrust is very fast [1].

To reflect the recent personalization approach in which each peer has an initial trust group and then gradually extends the group and updates the trust information, we assign different weights for different numbers of transitive relations. More specifically, we assign the highest weight to the direct trust and the second highest to the one-hop-indirect trust. In this way, as the number of transitive operations increases, we decrease the weight for trust calculation as follows:

$$T_t = (IC + C^2 + \cdots + C^t)/t = (\Sigma_{k=1}^{t} C^k)/t. \qquad (3)$$

In the matrix T_t, the i-th vector is peer i's personalized trust vector $\vec{t_l}$. By using (3), we consider two improvements over the original algorithm. First, we assign a lower weight as more transitive operations are applied as in [2], [3], and [7] where the number of transitive operations is limited. Second, we slow down the convergence speed to weigh personal direct trust more heavily and to reflect the change of trust relations (from personal direct trust) conservatively.

Theorem 1 demonstrates that if t goes to infinity, T_t finally converges to the original EigenTrust.

Theorem 1. Assume that $\lim_{t\to\infty} C^t = C^\infty$. Then, $\lim_{t\to\infty} T_t = (\Sigma_{k=1}^{t} C^k) C^\infty / t$.

Proof. Without loss of generality, let $c_{ij}^{(k)}$ be the entry of C_k in the i-th row and j-th column. Then, $\lim_{t\to\infty} C_{ij}^{(t)} = C_{ij}^{\infty}$ since all of the entries in C^∞ are converged to fixed values. Therefore, we just need to prove the entry $d_{ij}^{(t)}$, which is in the i-th row and j-th column of T_t, converges to $c_{ij}^{(\infty)}$ in order to prove the theorem. By the definition of limit, there exists a natural number k_1 s. t. for any $\varepsilon/2 > 0$, $t \ge k_1 \Rightarrow \left|c_{ij}^{(t)} - c_{ij}^{(\infty)}\right| < \varepsilon/2$.

Let M be the maximum value among $\left|c_{ij}^{(t)}-c_{ij}^{(\infty)}\right|,\cdots,\left|c_{ij}^{(k_1)}-c_{ij}^{(\infty)}\right|$. Then, $\left|d_{ij}^{(t)}-c_{ij}^{(\infty)}\right| \le k_1 M/t+(t-k_1+1)/t\cdot\varepsilon/2 \le k_1 M/t+\varepsilon/2$. Since $\lim_{t\to\infty} k_1 M/t=0$, there exists a natural number k_2 s. t. for any $\varepsilon/2<0$, $t \ge k_2 \Rightarrow \left|k_1 M/t\right| < \varepsilon/2$. Thus, for any $\varepsilon>0$, $t>\max(k_1,k_2)\Rightarrow\left|d_{ij}^{(t)}-c_{ij}^{(t)}\right|<\varepsilon$. Therefore, $\lim_{t\to\infty} d_{ij}^{(t)}=C_{ij}^{(\infty)}$.

□

If we modify (3) as follows, we can achieve personalized trust values more flexibly by adjusting α:

$$T_t'=(\alpha^0 IC+\alpha^1 C^2+\cdots+\alpha^{t-1}C^t)/(\alpha^0+\alpha^1+\cdots+\alpha^{t-1}). \quad (4)$$

In (4), when $\alpha=1$, $T_t'=T_t$ of (3). If $\alpha=0$, $T_t'=IC=C$, that is, local trust values. $0<\alpha<1$, T_t' contains combined values between local trust and global trust.

III. Experimental Results

To evaluate the enhanced scheme we conducted two experiments. In the first experiment, we focused on calculation of local trust values. We assumed that there were 3 peers *A*, *B*, and *C*, where *A* and *B* perform transactions with *C*, and *C* computes the local trust values of *A* and *B* using EigenTrust and the enhanced scheme, respectively. We considered every possible case where each peer has provided satisfactory/unsatisfactory transactions. Experiments show that, for some cases, relative values of the original local trust show inconsistent inequality relationships, that is, the local trust values of EigenTrust are $c_{CA}>c_{CB}$, whereas those of the enhanced scheme are $c_{CA}<c_{CB}$. Because the enhanced scheme is based on the beta distribution that reflects relative local trustworthiness correctly, this difference indicates the case where the original algorithm computes incorrect local trust relations. Figure 1 shows the simulation results. The *x* and *y* axes are the numbers of transactions of *A* and *B*, respectively, and the *z* axis

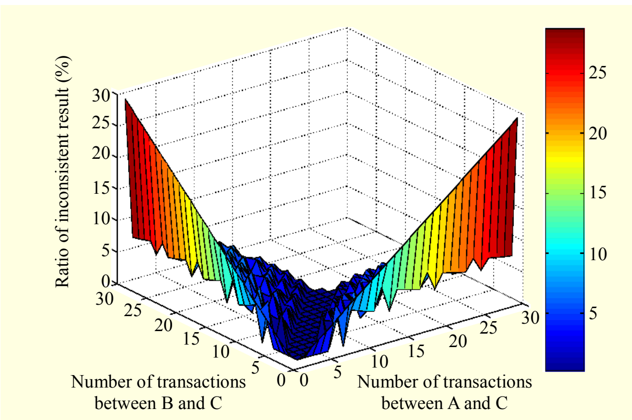

Fig. 1. Comparison of local trust values of EigenTrust and those of the enhanced scheme.

is the probability of such an inconsistent occurrence. Figure 1shows that such an inconsistency occurs with a probability of up to 8.7%.

In the next experiment, we compared the convergence rates between EigenTrust and the enhanced scheme. We set the number of peers is 50 and defined a round as each case in which a peer performs a transaction and exchanges local trust information with others evenly. Experimental results show that EigenTrust has fast convergence within 5 rounds, while our personalized EigenTrust has slower convergence (from personal local trust to global trust), taking 15 to 20 rounds, which reflects our conservative approach. We also conducted experiments for various α values, which confirmed that for the case of $0<\alpha<1$, the computed trust values have intermediate results between global and local trust values.

IV. Conclusion

In this letter, we presented an enhancement of EigenTrust by using the beta distribution. Simulation results showed that the enhanced scheme can correct inaccurate cases, which are 2% to 8% on average. Moreover, experimental results show that the proposed algorithm produces personalized trust values by adjusting α.

References

[1] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. WWW Conf.*, 2003, pp. 640-651.

[2] U. Maurer, "Modelling a Public-Key Infrastructure," *Proc. ESORICS*, 1996, pp. 325-350.

[3] A. Gutscher, "A Trust Model for an Open, Decentralized Reputation System," *Proc. IFIPTM*, 2007, pp. 285-300.

[4] G. Shafer, *A Mathematical Theory of Evidence*, Princeton Univ. Press, 1976.

[5] K. Sentz and S. Ferson, *Combination of Evidence in Dempster-Shafer Theory*, SANDIA Tech. Report, SAND2002-0835, 2002.

[6] A. Jøsang, "Artificial Reasoning with Subjective Logic," *Proc. AWCR*, 1997, pp. 204-211.

[7] A. Gutscher, J. Heesen and O. Siemoneit, "Possibilities and Limitations of Modeling Trust and Reputation," *Proc. CEUR Workshop*, 2008.

[8] A. Jøsang and R. Ismail, "The Beta Reputation System," *Proc. BCEC*, 2002, pp. 324-337.

[9] Q. Lian et al., "Robust Incentives via Multilevel Tit-for-Tat," *Concurrency and Computation: Practice & Experience*, vol. 20, no. 2, 2008, pp. 167-178.