IEICE TRANS. INF. & SYST., VOL.E93–D, NO.3 MARCH 2010

PAPER Special Section on Trust, Security and Privacy for Pervasive Applications

HopRec: Hop-Based Recommendation Ability Enhanced Reputation Ranking in P2P Networks

Yufeng WANG[†,††a)], ***Nonmember***, **Akihiro NAKAO**[††,†††], ***Member***, ***and*** **Jianhua MA**[††††], ***Nonmember***

SUMMARY As a concept stemmed from social field, we argued that, in P2P networks, peers' recommendation behaviors and functional behaviors should be explicitly separated, thus we propose the HopRec scheme which uses hop-based recommendation ability to improve the accuracy of reputation ranking in P2P networks. Our contributions lie in the following aspects: firstly, we adopt the simple but effective idea to infer peer's recommendation ability (RA): the farer away that peer is from the initial malicious seeds, the higher RA that peer should have; Then, the computation of reputation rankings appropriately reflects peer's different RA. The simulation results show that, in comparison with Eigentrust-like algorithms, HopRec can be robust to sybils and front peers attacks, and achieve significant performance improvement. Moreover, we compare HopRec with two related schemes, Poisonedwater and CredibleRank, and found that: in hospitable P2P environment, HopRec can obtain better performance than Poisonedwater, and can achieve the comparable performance as CredibleRank, with less computation overhead then CredibleRank. Finally, we also show that, if the initial good and malicious seeds could be selected based on peers' degrees, then HopRec and CredibleRank can achieve perfect performance.

key words: *P2P, reputation rankings, recommendation ability*

1. Introduction

The rapid growth of distributed and autonomous communication networks such as Peer-to-Peer (P2P) and wireless mesh networks etc., has spurred the development of numerous collaborative applications. Reputation and trust play a pivotal role in such applications, which can assist participants in deciding whether or not to transact with others, through aggregating ratings about a given participant to derive a trust or reputation score. Obviously, the concept of reputation is closely linked to that of trustworthiness, but there is a clear and important difference. The most distinguished difference lies in that, trust systems produce a score that reflects the trusting entity's subjective view about trusted entity's trustworthiness, whereas reputation is referred to as a single value (more technically, a social evaluation) that represents what the community as a whole thinks about a certain user. In the context of collaborative applications such as P2P systems, reputation represents the opinions that nodes in the system have about their peers and peer-provided resources. Briefly, reputation can be considered as a collective measure of trustworthiness based on the referrals or ratings from members in a community, and an individual's subjective trust can be derived from a combination of partner's reputation and its personal experience [1], [3]. The objective of this paper is to infer relatively accurate global reputation ranking.

Reference [2] identifies the following three dimensions as being fundamental to any reputation system: formulation, calculation and dissemination. Formulation represents the ideal mathematical underpinnings of the reputation metric and the sources of input to that formulation; calculation is the algorithm used to calculate the mathematical formulation for a given set of constraints (physical distribution of participants, type of communication substrate, etc.); dissemination includes the mechanism that allows system participants to obtain the reputation metrics resultant from the calculation. Such a mechanism may involve storing the values and disseminating them to the participants. The most important component of the formulation dimension is the mathematical or algorithmic representation of the reputation, which produces a metric encapsulating reputation for a given domain for each identity. These metrics seek to generate an accurate assessment in potentially adversarial environments.

Most social-network based reputation metrics adopt link analysis to infer peer's reputation according to peer's position in trust graph (the seminal paper of Brin and Page [4], and Kleinberg [5] introduce the area of link analysis to rank web pages, in which the link from page i to j is viewed as some vote for page j's quality from page i). Analogous to the PageRank measure for web pages, EigenTrust [6] and TrustRank [7] assign a universal measure of reputation ranking to each peer in P2P system according to the underlying trust graph (based on the intention: good peers will put high trust value on other good peers). Since high reputation scores can bring benefits for peers, it is expected that malicious peers would try to distort the correctness of the algorithm. Considering the autonomy and openness of most P2P networks, for the reputation metrics, the most serious attacks are sybils and front peers [8], [9]. Front peers represent these malicious colluding peers always cooperate with others in order to increase their reputation. They then provide misinformation to promote actively mali-

Manuscript received June 29, 2009.
Manuscript revised October 1, 2009.

[†]The author is with Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, China.
[††]The authors are with National Institute of Information and Communications Technology (NICT), Tokyo, 113–0001 Japan.
[†††]The author is with The University of Tokyo, Tokyo, 113–0033 Japan.
[††††]The author is with Hosei University, Koganei-shi, 184–8584 Japan.
a) E-mail: wfwang@nict.go.jp
DOI: 10.1587/transinf.E93.D.438

cious peers. The name, a front peer, is after its act of standing in the front of this colluding group of malicious peers. It is argued that this form of attack is particularly difficult to prevent [9]; Sybils attack means that, in near zero-cost of peer identity (it is the case that we focus on, and it is also the extreme charming point of P2P systems), new identities, or sybils, may be created cheaply and in large numbers. Thus, a peer can create an unlimited number of sybils that may link to (or perform fake transactions with) each other so that a peer may improve his/her own reputation.

It is argued that using social network structure with reputation management (always combined with personalized trust metric) can solve sybils attack effectively [10], [11]. The key intention in social-network based reputation ranking schemes lies in that: the false identities would either only connect to their old friends or remain disconnected, in which cases they will have poor social ranking (reputation ranking) since they cannot receive enough reputations from other good peers to increase their own. For example, Eigentrust can effectively combat sybils attack, if the pre-trusted peers are carefully selected, such that no malicious peers are included. But the implicit assumption in Eigentrust that the quality of a peer (functional reputation) and the quality of a peer's links (recommendation ability) are highly correlated is vulnerable to front peers attack, because, through providing good services, front peers can accumulate high reputation value, and propagate most of its reputation value to their malicious neighbors, which lead to malicious peers having high reputation values.

In this paper, we argue that, in order to improve the robustness of global reputation ranking, it is imperative to differentiate two social concepts: RA and reputation about actually being as good service providers, and properly use peers' RA to calculate peers' reputation rankings. Intuitively, we adopt the simple but effective method to infer peers' RA: the farer away the peer is from malicious seeds (a small set of malicious seeds could be initially identified), the higher RA the peer should have. And then, peers' RA is integrated into the procedure of calculating the peers' reputation ranking, to properly reflect peers' different recommendation ability.

The rest of paper is organized as follows: Sect. 2 briefly introduces some related work, and their drawbacks, including Eigentrust, Poisonedwater and CredibleRank. In Sect. 3, we propose the HopRec scheme, which includes two phases: the inference of RA based on hops away from initial malicious seeds, and the computation of reputation scores based on peers' RA. Section 4 gives the simulation settings and results, and compares the performance of HopRec with the above existing schemes in various P2P environments. Section 5 discusses several issues in HopRec. Finally, we briefly conclude this paper.

2. Related Work

Traditionally, trust relationship among peers (or web structure) is represented as a directed graph $G = (V; E)$, in which, V is the set of peers (pages), and the edges E represent direct trust between peers (pages). That is, edge $(i, j) \in E$ connecting peers (pages) i and j, denotes (conveys) the direct trust of peer (page) i on peer (page) j, which, in our paper, is assigned continuous weight $w(i, j) \in [0, 1]$. Generally, local trust values can be generated by Bayesian learning or by an average rating based on peer satisfaction [16], [17], which are beyond the scope of our paper. Note that, in Web system, the adjacent matrix W of the web graph is: $w(i, j)$ equals 1 if there is a hyperlink from page i to page j, or 0, otherwise.

In our paper, the row normalized adjacent matrix, denoted as RW, is defined as follows: the sum of each *row* in adjacent matrix W equals 1. That is:

$$rw(i, j) = w(i, j) \Big/ \sum\nolimits_j w(i, j) \tag{1}$$

Note that, just as most previous work, in RW definition, we will insert a self-loop for all peers with outdegree 0.

RW^T have been used in the Eigentrust and Pagerank to iteratively calculate the peer's reputation ranking (page's quality) according to the following equation:

$$PR^{(k+1)} = \alpha \cdot RW^T \times PR^k + (1 - \alpha) \cdot p \tag{2}$$

where vector p is the initial trust value on some pre-trusted peers (pages), so-called personalized Eigentrust (Pagerank). The constant parameter α, termed as trust dampening factor, always equals 0.85. When the iterative number k is large, the final reputation vector PR will converge to the principal eigenvector of matrix $[\alpha \cdot RW + (1-\alpha) \cdot e \cdot p^T]^T$, where e is the N-vector whose elements are all $e_i = 1$, and N is the number of peers in P2P systems.

But, as illustrated in Sect. 4, Eigentrust-like algorithm is vulnerable to front peers attack, and the reason of this vulnerability lies in that two different roles, peer's functional reputation and peer's recommendation ability, are entangled in Eigentrust-like algorithms. That is, a peer ranked high will be unlikely to contain local high quality links to bad peers.

A natural extension of the idea of the conveyance of trust is that of the conveyance of distrust. Reference [12]–[14] investigate the possibility of propagating distrust among web pages to demote more spam sites than the sole use of trust values. Specifically, the above work relies on an inverse PageRank-style model to assign a spam proximity value to every source in the Web graph, (similar to the BadRank approach for assigning in essence a "negative" PageRank value to spam [15], badness).

Our previous work also adopts the idea of inverse propagation of distrust (called poisoned water in [18]), and designs the logistic way to convert the badness into the adaptive spreading factor. Poisonedwater replaces the trust dampening factor in traditional reputation ranking algorithms (the constant parameter α, 0.85 in Eq. (2)) with the adaptive spreading factor to compute peers' reputation rankings, and shows that the performance of Poisonedwater is significantly better than Eigentrust in hostile network environments. Moreover, in Poisonedwater, we also investigate

the effect of intentional selection of initial good and malicious seeds on the ranking performance. Surprisingly, we drew the following conclusion: considering the randomized property used in logistic model, the way to randomly select initial good and malicious seeds performs better than that of selection based on peers' degree. The difference of HopRec from the Poisonedwater lies in that, instead of using the propagation of "poisoned water" along the reverse indegree direction, and the logistic model to infer peers' recommendation ability, HopRec simply uses the number of hops to initial malicious seeds to infer peers' RA, and can achieve better performance than Poisonedwater in hospitable network environments. More important, in HopRec, if we can intentionally select initial malicious seeds according to their degrees, the ranking performance of HopRec is almost perfect (extremely better than degree-based Poisonedwater).

Recently, in web field, Ref. [14] introduces the concept of link credibility, and develops a credibility-based Web ranking algorithm, CredibleRank, which incorporates credibility information into the quality assessment of each page on the Web. Specifically, the credibility of each page equals the product of two components: the first component evaluates the credibility of a page in terms of the quality of a random walk originating from the page and lasting for up to k steps. Intuitively, the above component models a random walker who, when arriving at a spam page, becomes stuck and ceases his random walk, and for all other pages the walker continues to walk, for up to k hops; due to the size of the Web, the cost of crawling all pages, and only small partial spam seed pages, the authors introduce the second component, credibility penalty factor, into page's credibility, and discuss several ways to compute the credibility penalty factor (the hop-based method is included). In a sense, the recommendation ability in HopRec is like the credibility (especially the second component) in CredibleRank, but, interestingly, we found that, the time and resource consuming component of random walking just brings trivial improvement of reputation ranking, and the key factor in improving the reputation ranking is the second component, the credibility penalty factor, which is called hop-based recommendation ability in our paper. The reason behind the above phenomenon is that, according to the CredibleRank, page's credibility calculated in the first component is almost same for all peers, so that it could not effectively distinguish pages' credibility.

Based on the above observation, we design the simple reputation ranking algorithm, HopRec, which uses the intuition that the farer away from the malicious seeds, the higher the recommendation ability. The advantage of HopRec over CredibleRank lies in that, we achieve comparable performance as CredibleRank, but without the time and resource consuming k-step random walking. Furthermore, we show that if, the initial seeds could be intentionally selected based on peers' degrees, HopRec and CredibleRank can achieve same perfect performance.

3. HopRec Scheme

Generally, given that some malicious and good seeds could be initially identified, HopRec scheme is composed of two phases: the first phase is to infer peers' RA based on the number of hops to initial malicious seeds; second phase is to update the adjacent matrix, to reflect peer's RA in trust propagation, and uses the updated trust matrix to calculate peer's reputation ranking.

3.1 Hop-Based RA Inference

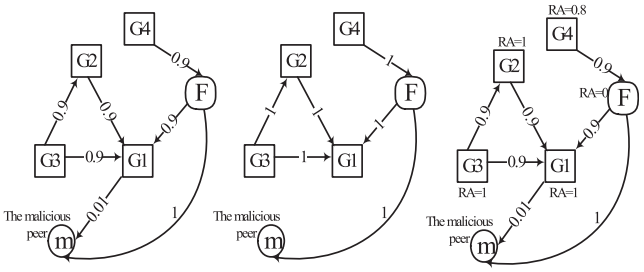
Unlike the hyperlink structure in web systems, in P2P trust graph, the link weight between two neighboring peers is assigned a continuous value in [0, 1]. Thus, we can not simply view as one hop between two neighboring peers only if there exist link between those two peers. For example, assume that good peer i has trust link weight 0.01 on a malicious peer j (for the malicious peer provides 1% good service, or for the good peer mistakenly rates), but we can not say that, there exist one hop from peer i and peer j, and peer i would have bad recommendation ability. Thus, in P2P reputation ranking, we should firstly preprocess the trust adjacent matrix, before we use the hop-based method to infer peers' RA. Simply we could set the low link weights in trust graph as zero, which actually means that there exist no link between those neighbors, (generally, those kinds of links denote the ratings between good peers and malicious peers), and promote the high link weights to 1. This above simple way has the following advantages: we could avoid the following attacks: some malicious peers bedim their presence in the network by occasionally providing good services in small percentage (like 1%) of all transactions (note that front peers provide good service in almost 100% of all transactions) [23]; and we could accommodate unintentional mistakes made by good peers. Naturally, the threshold 0.5 is good candidate, for it is too costly for the malicious peers to make cooperation in 50% of all their transactions, and even though, those peers do exist, and according to the preprocessing method, we would promote their weights to 1, but, HopRec could detect those as front peers.

Briefly, we construct an $N*N$ matrix NW that modifies the original trust adjacent matrix in the following way:

$$NW_{ij} = \begin{cases} 1, & \text{if } W_{ij} \geq 0.5 \\ 0, & \text{if } W_{ij} < 0.5 \end{cases} \tag{3}$$

Then, based on the direct graph defined by the matrix NW, we define the hop-h bad path of peer i as follows: there exist at least one path of length h from peer i to any (or more) peer(s) in the set of initial malicious seeds. We represent the influence of hop-h bad path on peer i's recommendation ability as $RA_h(i)$, then the hop-based recommendation ability of peer i, $RA(i)$, can be calculated as the product of the constituent $RA_h(i)$, that is,

$$RA(i) = \prod_{h=1}^{L} RA_h(i) \tag{4}$$



(a) The initial trust graph. (b) The modified trust graph. (c) The RA of peers based on HopRec.

Fig. 1 The instructive illustration of HopRec.

where $RA_h(i)$ can exponentially dampen in the hop to the initial malicious seeds. Similar as CredibleRank, we define:

$$RA_h(i) = 1 - \varphi^{h-1} \tag{5}$$

Procedure RecInference
INPUT: the initial malicious seeds MS; the revised trust adjacent matrix NW; $\varphi = 0.2$; the maximal hop numbers $L = 6$.
OUTPUT: recommendation ability of each peer i, $RA(i)$
Initialize: for each peer i, let $RA(i) = 1$;
FOR $h = 1$: L
  midmatrix = NW^h
  FOR $i = 1$: N
    FOR each j in MS
      IF (midmatrix(i, j) > 0)
        $RA(i) = RA(i) * (1 - \varphi^{h-1})$
        BREAK
      ENDIF
    ENDFOR
  ENDFOR
ENDFOR

Algorithm 1: Computation of peers' RA (pseudocode)

The above pseudocode can easily compute the RA for each peer. Note that, considering the popularly known "six degrees of separation" in social systems, we set the maximal hop number as 6. Moreover, in our simulations, we set $\varphi = 0.2$ (corresponding to relatively small decaying factor for RA computation). The underlying intention lies in that, for our previous work, Poisonedwater can effectively deal with the hostile P2P environments, thus, in HopRec scheme, we attempt to effectively solve front peers attack in relatively hospitable P2P environments, which generally means that the percentage of good peers is larger than that of non-good peers, and naturally we use the small decaying value. Of course, intuitively, if the value of φ can be adaptively determined, and then HopRec could deal with various networking environments, which should be deeply investigated in future work.

Figure 1 instructively illustrates the procedure of RecInference, in which peer m is initial malicious peer, peer F is front peer, and peers from G1 to G4 are good. Figure 1 (b) shows the modified trust graph using Eq. (3). Figure 1 (c) simply shows the inferred RA for each peer based on the number of hops away from the initial malicious seed m. Obviously, F's RA is zero. Note that even though G4 is same as G1, G2 and G3 (all are good peers), but the recommendation ability of G4 is 0.8 (other good peers' RA is 1). In a sense, it is fair, because, there exist two-hop path from G4 to the initial malicious peer m, which means that G4 indirectly (implicitly) recommends the malicious peer m, naturally, the G4's RA should be lower than other good peers.

Theoretically, there also exist drawback with hop-based RA inference, that is, malicious peer could create long sybil chain to dampen the effect of hop-based RA inference. But, actually, since malicious peers are initially identified by good peers, and it is extremely difficult for sybils to get some trust links from good peers, thus the probability of the above attack is very small.

3.2 Reputation Ranking Inference

Traditionally, in Eigentrust, the following equation is used to iteratively update peers' reputation ranking:

$$FR^{(k)} = \alpha \cdot RW^T \times FR^{(k-1)} + (1 - \alpha) \cdot FR^{(0)} \tag{6}$$

$FR^{(0)}$ represents the initial trust value on the set of pre-trusted peers (initial good seeds).

After we obtain the each peer's RA computed by the above procedure RecInference, we first revise the row normalized adjacent matrix RW used in reputation propagation to seamlessly integrate peer's RA. Let

$$RW_{ij} = RW_{ij} * RA(i), \qquad (i,\ j = 1, \ldots, N) \tag{7}$$

where N is the number peers in system.

Note that, the semantic meaning of each item RW_{ij} in row normalized trust matrix RW implies that peer i should propagate RW_{ij} of its reputation value to peer j. Naturally, we can use the Eq. (7) to integrate recommendation ability of each peer i in propagating reputation.

In matrix form, we can define the matrix RW_{hop} used in HopRec:

$$RW_{hop} = RA_{hop} \times RW \tag{8}$$

where RA_{hop} is diagonal matrix, in which each element is the corresponding peer's recommendation ability. Then, through replacing matrix RW in Eq. (6) with matrix RW_{hop}, HopRec can be used to infer peer's reputation ranking.

4. Simulation Settings and Results

4.1 Simulations Settings

This subsection describes the simulation setups, including the algorithm parameters, network environment types, peer behaviors, and the procedure of generating trust network. Shown in Table 1, there exist two kinds of peers in our simulation settings: good peer and non-good peer including malicious peer and front peer. Note that we roughly use hostile

Table 1 Algorithm parameters, network environment types and behavior patterns in our simulations.

	Parameter and Value range
Algorithm parameters	Number of peers in the network: 1000~2000
	Threshold of iterative process ,ε=10^{-4}
	Percentage of initially selected good seeds and malicious seeds: 1%, 2% and 3%
	(optional) Each front peer and malicious peer can create 10 sybils, and form links with them
	Methods to select initial good and malicious seeds: ● Random: select those peers randomly; ● Degree-related: select those peers according to their degree
Hostile network environmen t	Percentage of good peers: 30% Percentage of non-good peers: 70% ● Percentage of front peers: 20% ● Percentage of malicious peers: 50%
Hospitable network environmen t	Percentage of good peers: 60% Percentage of non-good peers: 40% ● Percentage of front peers: 20% ● Percentage of malicious peers: 20%
Peer behavior patterns	Good Peer (G): always provide truthful feedback about the transaction partner. $w(G, G)$=$w(G, F)$=0.9, and $w(G, M)$=0.01
	Front peer (F): like good peer, except providing false feedback about the malicious peer. $w(F, G)$= $w(F, F)$= $w(F, M)$=0.9
	Malicious peer (M): provide bad feedback for good peer, and provide good feedback for malicious peer and front peer. $w(M, F)$=$w(M, M)$=0.9, and $w(M, G)$=0.01

P2P environments to denote that the percentage of non-good peers is larger than percentage of good peers, and hospitable P2P environments, vice verse. Moreover, all results in simulations are the average of five runs.

Naturally, occurring trust networks takes a long time to gain a large number of users, and the topological properties are relatively fixed, thus it is necessary to be able to automatically generate trust networks. It is argued that reputation feedback in trust network complies with power-law distribution [19]. Thus, we use the Barab'asi-Albert model, to construct experimental trust graph. Specifically, the detailed construction procedure is given as follows:

① Growth: Starting with 50 nodes, at each round we add 10 new nodes, each with 10 edges;

② Preferential Attachment: A naive simulation of the preferential attachment process is quite inefficient. In order to attach to a vertex in proportion to its degree we normally need to examine the degrees of all vertices in turn, a process that takes O(n) time for each step of the algorithm. Thus the generation of a graph of size n would take O(n^2) steps overall. A much better procedure, which works in O(1) time per step and O(n) time overall, is given as follows [20]. In this paper, we maintain a list, in an integer array for instance, that includes d_i entries of value i for each peer i. Then in order to choose a target peer for a new edge with the correct preferential attachment, one simply chooses a number at random from this list. When new peers and edges are added, the list is updated correspondingly.

Considering that, the main objective of HopRec is to investigate the effect of separation between recommendation behaviors and functional behaviors on the accuracy of reputation ranking, so we use simple and idealized method to construct the trust relationship among peers. That is, the direct trust weight between two neighbors is determined by both peers' types. Shown in Table 1, the term, $w(G, F)$ represents the local trust value from a peer of type G to another peer of type F. Basically, we select the value, 0.9 to represent the high trustworthiness that one peer puts on another peer, and 0.01, the low trustworthiness. The above values can accommodate some rating error related to local trust weight. Note that, our simulations also adopt other similar values to denote high and low trustworthiness, and the similar results can be obtained. Some more complicated ways to generate local trust weight through realistic transactions, can be found in [21], [22]. We argue that the methods used to generate the local trust relationships among peers would not affect the main results in this paper, for the objective of all those methods is to put high local trust weights on trustworthiness peers, and low weights on malicious peers by good peers (similar to our idealized way to generate local trust weights).

4.2 Measurement

In ideal reputation ranking algorithms, good peers should occupy those first positions in ranking list, and the less malicious peers are assigned into those first positions, the better is the reputation ranking algorithm. Note that some front peers may occupy those first positions, for they also provide good service for other peers. In a sense, HopRec scheme can be viewed as a classification problem. We define ranking error ratio—among the peers from the top of the reputation ranking (set A), how many peers are actually malicious (set B)—to evaluate the performance of HopRec. That is, $ranking - error - ratio = |B|/|A|$. Specifically, in HopRec scheme, set A denotes the high reputation ranking list returned by HopRec and Eigentrust schemes, and set B represents the list of malicious peers included in set A. Moreover, the size of list A equals the number of good peers in the system.

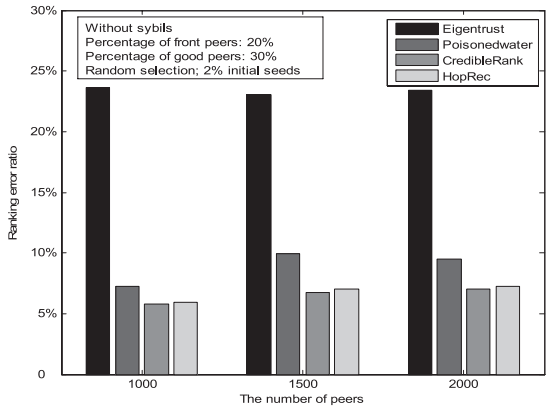
Note that, in [14], the authors introduce the measurement of rank-based spam resilience, SR, to evaluate the quality of a candidate ranking algorithm versus a baseline ranking algorithm. For our paper, the candidate ranking algorithms is the HopRec, and the baseline ranking is Eigentrust. Thus, specifically, in our paper, SR could be similarly defined as follows: the sum of rankings of malicious peers in HopRec divided by the sum of rankings of malicious peers in Eigentrust, then minus 1. So, a candidate ranking algorithm that induces a more spam-resilient ranking will result in positive SR value, and negative values indicate that the candidate algorithm is less spam-resilient

than the baseline. In our paper, we also investigated the SR values of HopRec, and observed that, the SR value are ever positive. Considering that, in most transactions of P2P networks, users mainly care about whether the high reputation peer list includes malicious peers or not (probably, we do not care about most malicious peers' rankings, for they should be low). So, in this paper, we view the reputation ranking as a classification problem, and use ranking error ratio as the measurement (in ideal ranking, the ranking error ratio should be 0, for the size of set A in our measurement equals the number of good peers in the systems).

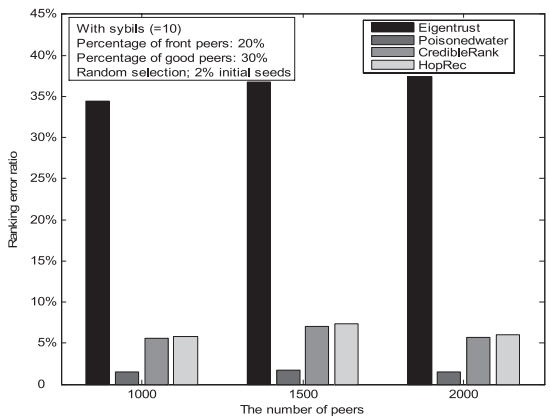
4.3 Simulation Results

In this subsection, we conducted extensive simulations to illustrate the qualitative property of HopRec in comparison with other existing schemes (for completeness, we also implemented the Poisonedwater and CredibleRank) in various P2P environments (hostile and hospitable environment, without and with sybils), which almost are neglected by related work.

Figure 2 (a) shows that the performances of Eigentrust, Poisonedwater, CredibleRank and HopRec, with the change of total peers in the systems when there exist no sybils, and Fig. 2 (b) shows those performances when each non-good peer can create 10 sybils (in those simulations, additional 20% of total peers are regarded as the pool of sybils, each non-good peer randomly select 10 sybils from the pool to form links). Note that, in those simulations, the percentage of good peers is set as 30%, the so-called hostile environment. From Fig. 2 (a) and (b), we can draw the following conclusions:



(a) Without sybils.



(b) With sybils.

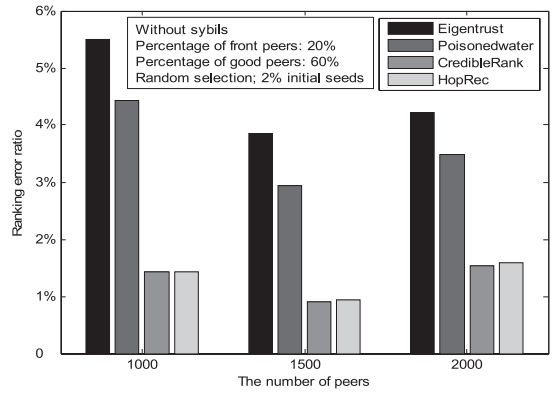
Fig. 2 Ranking error ratio vs. the number of peers in hostile environments.

- In all the scenarios (with and without sybils), Poisonedwater, CredibleRank and HopRec perform better than Eigentrust, and front peer plus sybils attack is more harmful to Eigentrust than only front peer attack, but has no negative effect on Poisonedwater, CredibleRank and HopRec. The reason lies in that, those three schemes can detect front peers, and assign low recommendation ability to those front peers, which prevent front peers from passing their reputation values to sybils and malicious peers, such that sybils and malicious peers can not promote each other.
- Interestingly, without sybils, the CredibleRank and HopRec slightly do better than Poisonedwater (Fig. 2 (a), but, with sybils, on the contrary, Poisonedwater does better than CredibleRank and HopRec (Fig. 2 (b)). The reason lies in that, the more hostile P2P environment is, the more accurately that Poisonedwater predict peers' recommendation ability (considering Poisonedwater uses nonlinear least-squares regression to estimate the logistic model, and infers each peer's recommendation ability). Actually, in Fig. 2 (b), additional 20% sybils increase the proportion of non-good peers in P2P systems, while leads to the more hostile P2P environment.

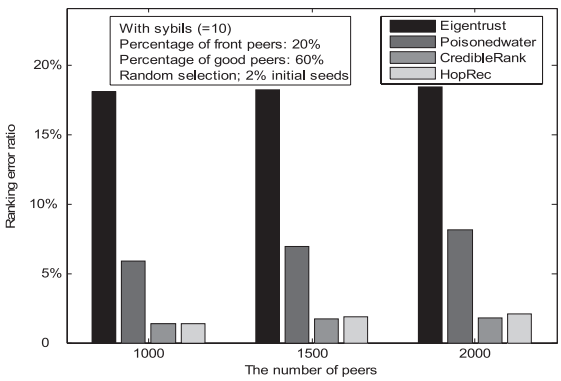
Similarly, Fig. 3 (a) and (b) respectively illustrate those four schemes' performances (with sybils and without sybils) in hospitable P2P environment (the percentage of good peers is 60%). From those figures, we can infer the following results:

- Similar to performance in the hostile network environment, in all the hospitable scenarios, Poisonedwater, CredibleRank and HopRec perform better than Eigentrust, and front peer plus sybils attack brings more harm to Eigentrust, but has no much negative effect on Poisonedwater, CredibleRank and HopRec.
- On the contrary with the hostile environments, in hospitable environments, CredibleRank and HopRec perform better than Poisonedwater in two scenarios (with and without sybils). The reason is that, based on logistic model, in hospitable environment, Poisonedwater almost assigns same recommendation ability to each peers (like constant α in traditional Eigentrust), thus, in hospitable environments, Poisonedwater can not effectively distinguished peers' different recommendation ability. On the contrary, hop-based RA inference method can be effective in hospitable environment.

Note that, from the above Fig. 3 and Fig. 4, we can see that our proposed HopRec has almost the same perfor-



(a) Without sybils



(b) With sybils

Fig. 3 Ranking error ratio vs. the number of peers in hospitable environments.

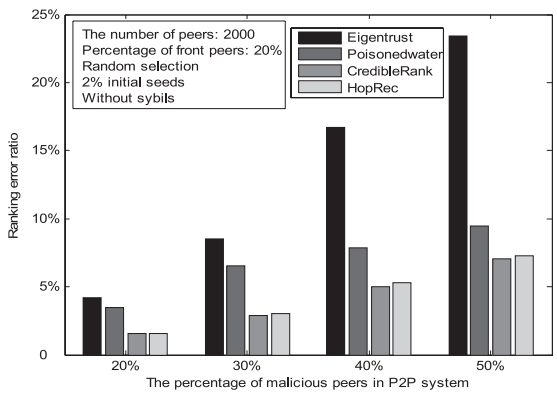
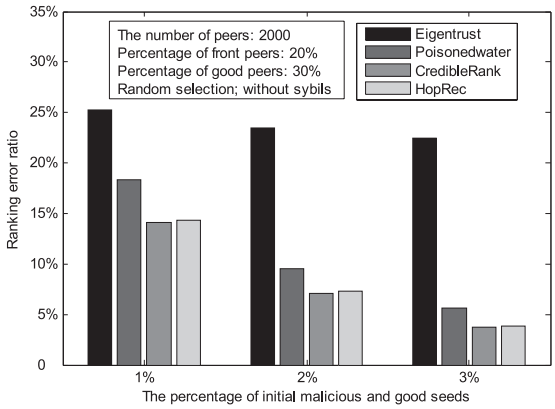
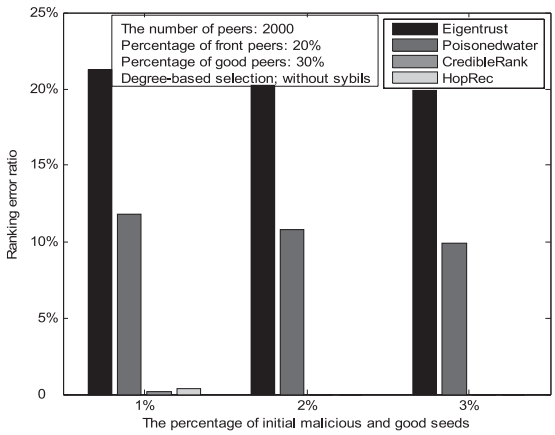


Fig. 4 Ranking error ratio vs. the percentage of malicious peers.

mance (ranking error ratio) as the available CredibleRank algorithm. The reason is that, in CredibleRank, peer's credibility calculated in the first component is almost same for all peers, so that it can not effectively distinguish peers' credibility, which also implies that the key factor in CredibleRank is the second component. In brief, our scheme, HopRec achieves the comparable performance as CredibleRank, with less computation overhead then CredibleRank. For the randomness of those schemes, we conducted a great deal of experiments with various range of parameters, and, qualitatively, we can obtain the similar conclusion: the difference between CredibleRank and



(a) Random selection.



(b) Degree-based selection.

Fig. 5 Ranking error ratio vs. the percentage of initial good and malicious seeds.

HopRec is trivial. The slight advantage of CredibleRank over HopRec comes with significant cost: the time and resource consuming component in CredibleRank: random walking.

Moreover, those figures also show that: Poisonedwater are extremely effective when P2P environment is very hostile (especially, in hostile environment with sybils, in comparison with Poisonedwater, CredibleRank and HopRec perform worse, shown in Fig. 2 (b)), but, in hospitable environments, the advantage of Poisonedwater is not so much (slightly better than Eigentrust), and, here we show that it is worse than CredibleRank and HopRec. The above results also give us a meaningful hint: there exist many schemes to improve the performance of P2P reputation ranking, but, probably, some schemes are more appropriate for some specific P2P environment than other schemes. It is also better, in real applications, to simultaneously use several different methods to infer the reputation ranking.

Figure 4 illustrates the variation of ranking error ratio with the change of the percentage of malicious peers (from 20% to 50%). We can see that, with the increasing of percentage of malicious peers, the performance of Eigentrust becomes worse greatly, but the ranking error ratio in other schemes only get worse slightly. Furthermore, in the all the scenarios (the hospitable P2P environ-

ments), the CredibleRank and HopRec performs slightly better than Poisonedwater, and the performance difference between CredibleRank and HopRec is trivial.

Figure 5 (a) and (b) show the ranking error ratio with the change of the percentage of initial malicious and good seeds selected at random and selected based on peer's degree respectively. That is, in Fig. 5 (a), our experiments randomly select small percentage of malicious and good peers as seeds, and Fig. 5 (b) intentionally selects initial seeds according to their degrees. We can see that, first, when there exist front peers in P2P system, the different selection ways have not much effect on the Eigentrust. The reason for this phenomenon is that, the way to select initial good seeds can not prevent front peers from passing their reputation values to their malicious friends, and ironically, front peers can get many links from good peers, because they always offer good service; Second, both figures show that, as the percentage of initial seeds increases, the performances of Poisonedwater, CredibleRank and HopRec became better. Interestingly, the degree-based selection of initial seeds has significantly positive effect on the performance of CredibleRank and HopRec. Specifically, from Fig. 5 (b), we can see that, when we can intentionally select 2% (or more than 2%) of initial malicious and good seeds, the ranking error ratio in CredibleRank and HopRec is zero in all experiments (for ranking error ratio is zero, the bars of CredibleRank and HopRec do not appear in Fig. 5 (b)). The reason behind the Fig. 5 (b) is straightforward: in hop-based recommendation ability inference, malicious seeds with high degree act like "powerful nodes", and their negative influence on other peers' RA can be easily and quickly propagated.

5. Discussion

A. The convergence properties of Eigentrust and HopRec algorithms

Note that, Eigentrust has elegant mathematical implication: the final reputation vector will converge to the principal eigenvector of matrix $[\alpha \cdot RW + (1-\alpha) \cdot e \cdot p^T]^T$, where e is the N-vector whose elements are all $e_i = 1$, and N is the number of peers in P2P systems, and the constant parameter α, always equals 0.85. Considering that, in HopRec, we use the inferred peers' recommendation ability to properly revise the row normalized matrix RW, thus, obviously, HopRec algorithm should have the similar mathematical implications: when the iterative number k is large, for HopRec, the final reputation vector will converge to the principal eigenvector of $[\alpha \cdot RW_{hop} + (1-\alpha) \cdot e \cdot p^T]^T$, where RW_{hop} are defined as Eq. (8).

B. The selection of initial good and malicious seeds

Like other personalized reputation ranking algorithms (Eigentrust, etc.), the pre-trusted peers and initial malicious seeds are essential to HopRec approach. Generally, we have to use other extraneous factors to select those peers, like, initially, the founders of the P2P social networks, which are commonly known to be trustworthy, and some peers that firstly join the system (usually known to be trusted by other peers). Then, the real social life relationships of those peers may be used to recommend other pre-trusted and initial malicious peers. Moreover, from long-term viewpoint, as the P2P social network evolves and grows, the designers of the P2P social networks can observe and identify some good peer (and malicious peers), and form the whitelist (and blacklist) that can be used as the candidates for the initial seeds. However, the above ways may be over optimistic in a distributed computing environment, for in the virtual world built on computer network, pre-trusted peers and initial malicious seeds may not last forever, and decision on the trustworthiness of those seeds is also required. In this environment, designing human-assistant semi-automatic methods to identify the pre-trusted and initial malicious peers is challenging and interesting work.

C. Distributed calculation of HopRec algorithm

Generally, HopRec only focuses on the reputation metrics in P2P system, and in principle, needs the whole trust graph at one place. However, there are some situations in which a global computation on the entire graph is impractical, e.g., if the link information of the whole network is not easily accessible, and we need a quick estimation for a particular peer. Clearly, distributed approaches based on partitioned graph and information exchanging among some peers, are needed. Reference [24] presents the JXP algorithm for computing Pagerank-style authority scores of Web pages that are arbitrarily distributed over many sites of a P2P network. The basic idea in JXP is that, each peer maintains partial view of whole Web system (different peers may have overlapping fragments), and locally computes authority scores with information obtained from other peers by means of random meeting with other peers. The authors also prove that the JXP scores converge to the true Pagerank scores that one would obtain by a centralized Pagerank computation on the global graph.

Intuitively, for distributed calculation of peer's reputation ranking, two main problems must be solved: accuracy of the computed values and computation speed. The above approach could be adapted to compute reputation ranking in distributed way, which should be deeply investigated in the future work.

6. Conclusion

Most of the popular link analysis based P2P reputation ranking algorithms compute unique reputation value for each peer through aggregating all peers' ratings. Those work implicitly assumed that the quality of a peer and the quality of a peer's links are strongly correlated: a peer ranked higher will be unlikely to contain local high quality links to bad peers. However, this assumption opens doors for front peers attack. As a concept stemmed from social field, reputation in P2P networks should exhibit multi-faceted features, that is, the peers' recommendation behaviors and functional behaviors should be explicitly separated. In this paper, HopRec adopts the simple idea to infer peers' rec-

ommendation ability: the farer that a peer is away from the initial malicious seeds, the higher recommendation ability that the peer should have; Then, the computation of reputation rankings appropriately reflects peer's different recommendation ability. We conduct extensive simulations, and compare HopRec with the traditional Eigentrust, and two enhanced reputation ranking schemes: Poisonedwater and CredibleRank, and found that: HopRec performs significantly better than Eigentrust; HopRec performs better than Poisonedwater in hospitable P2P environment; and, HopRec can achieve the comparable performance as CredibleRank, but without the time and resource-consuming component of random walking in CredibleRank; Furthermore, when the initial seeds can be intentionally selected according their degrees, then, with small percentage of initial seeds, HopRec and CredibleRank can achieve perfect performance. The above results also give us the meaningful hint: there exist many schemes to improve the performance of P2P reputation ranking, but, probably, some schemes are more appropriate for some specific P2P environments than other schemes.

Acknowledgement

This research is partially support by the 973 Program 2007CB310607, 863 Projects 2007AA01Z206 and 2006AA01Z235, NSFC Grants 60802022 and 60772062; the authors thank the anonymous reviewers for their suggestion on how to improve the previous draft of the articles; their comments were of great help.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, vol.43, pp.618–644, 2007.
[2] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," ACM Comput. Surv., vol.42,pp.1–31, 2009.
[3] C.-N. Ziegler and G. Lausen, "Propagation models for trust and distrust in social networks," Information Systems Frontiers, vol.7, pp.337–358, 2005.
[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Technical Report, Stanford Digital Library Technologies Project, 1998.
[5] J.M. Kleinberg, "Authoritative sources in hyperlinked environment," J. ACM, vol.46, no.5, pp.604–632, 1999.
[6] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," Proc. WWW, pp.640–651, 2003.
[7] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen, "Combating Web spam with TrustRank," Proc. 30th VLDB, pp.576–587, 2004.
[8] J.R. Douceur, "The sybil attack," Proc. International Workshop on Peer-to-Peer Systems, pp.251–260, 2002.
[9] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," Comput. Netw., vol.50, no.4, pp.472–484, 2006.
[10] T. Hogg and L. Adamic, "Enhancing reputation mechanisms via online social networks," Proc. ACM Conference on Electronic Commerce, pp.236–237, 2004.
[11] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman, "SybilGuard: Defending against Sybil attacks via social networks," Proc. SIGCOMM, pp.267–278, 2006.
[12] B. Wu, V. Goel, and B. Davison, "Propagating trust and distrust to demote web spam," Models of Trust for the Web (MTW), 2006.
[13] J. Caverlee, S. Webb, and L. Liu, "Spam-resilient Web rankings via influence throttling," Proc. 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp.26–30, 2007.
[14] J. Caverlee and L. Liu, "Countering Web spam with credibility-based link analysis," Proc. PODC 2007.
[15] B. Wu and B. Davison, "Identifying link farm spam pages," Proc. 14th International World Wide Web Conference (WWW), pp.820–829, 2005.
[16] Z. Liang and W. Shi, "Analysis of ratings on trust inference in open environments," Elsevier Performance Evaluation, vol.65, no.2, pp.99–128, Feb. 2008.
[17] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," IEEE Trans. Knowl. Data Eng., vol.16, no.7, pp.843–857, 2004.
[18] Y. Wang and A. Nakao, "Poisonedwater: An improved approach for accurate reputation ranking in P2P networks," in Future Generation Computer Systems (FGCS), Elsevier, 2009.
[19] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted P2P computing," IEEE Trans. Parallel and Distrib. Syst., vol.18, no.4, pp.460–473, 2007.
[20] M.E.J. Newman, "The structure and function of complex networks," SIAM Review, vol.45, pp.167–256, 2003.
[21] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on Epinions.com community," Proc. Twentieth National Conference on Artificial Intelligence (AAAI), pp.121–126, 2005.
[22] S. Rubin, M. Christodorescu, V. Ganapathy, N. Kidd, L. Kruger, and H. Wang, "Anomaly detection as a reputation system for online auctioning," Proc. 12th ACM Conference on Computer and Communications Security (CCS), 2005.
[23] D. Donato, M. Panicci, M. Selis, C. Castillo, G. Cortese, and S. Leonardi, "New metrics for reputation management in P2P networks," Proc. 3rd international workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2007.
[24] J. Xavier-Parreira, C. Castillo, D. Donato, S. Michel, and G. Weikum, "The JXP method for robust PageRank approximation in a peer-to-peer Web search network," VLDB Journal, vol.16, 2007.

Yufeng Wang received Ph.D degree in State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT), China, on July 2004. From Jul. 2006 till Apr. 2007, he worked as Postdoctoral researcher in Kyushu University, Japan. From May, 2007, he acted as associate Professor in Nanjing University of Posts and Telecommunications, China. Presently, he is expert researcher in National Institute of Information and Communications Technology (NICT), Japan. He also acts as guest researcher in State Key Laboratory of Networking and Switching Technology, BUPT, China. E-mail: wfwang@nict.go.jp; wfwang@njupt.edu.cn

Akihiro Nakao received B.S. (1991) in Physics, M.E. (1994) in Information Engineering from the University of Tokyo. He was at IBM Yamato Laboratory/at Tokyo Research Laboratory/at IBM Texas Austin from 1994 till 2005. He received M.S. (2001) and Ph.D. (2005) in Computer Science from Princeton University. He has been teaching as an Associate Professor in Applied Computer Science, at Interfaculty Initiative in Information Studies, Graduate School of Interdisciplinary Information Studies, the University of Tokyo since 2005. (He has also been an expert visiting scholar/a project leader at National Institute of Information and Communications Technology (NICT) since 2007.) E-mail: nakao@iii.u-tokyo.ac.jp

Jianhua Ma received his B.S. and M.S. degrees of Communication Systems from National University of Defense Technology (NUDT), China, in 1982 and 1985, respectively, and the PhD degree of Information Engineering from Xidian University, China, in 1990. He has joined Hosei University since 2000, and is currently a professor at Digital Media Department in the Faculty of Computer and Information Sciences, in Hosei University, Japan. Dr. Ma is a member of IEEE and ACM. He has edited 10 books/proceedings, and published more than 150 academic papers in journals, books and conference proceedings. E-mail: jianhua@hosei.ac.jp