PIN-TRUST: Fast Trust Propagation Exploiting Positive, Implicit, and Negative Information

Min-Hee Jang
Hanyang University
Seoul, Korea
zzmini@hanyang.ac.kr

Christos Faloutsos
Carnegie Mellon University
Pittsburgh, USA
christos@cs.cmu.edu

Sang-Wook Kim*
Hanyang University
Seoul, Korea
wook@hanyang.ac.kr

U Kang
Seoul National University
Seoul, Korea
ukang@snu.ac.kr

Jiwoon Ha
Hanyang University
Seoul, Korea
jiwoonha@hanyang.ac.kr

ABSTRACT

Given "who-trusts/distrusts-whom" information, how can we propagate the trust and distrust? With the appearance of fraudsters in social network sites, the importance of trust prediction has increased. Most such methods use only explicit and implicit trust information (e.g., if Smith likes several of Johnson's reviews, then Smith implicitly trusts Johnson), but they do not consider distrust.

In this paper, we propose PIN-TRUST, a novel method to handle all three types of interaction information: *explicit* trust, *implicit* trust, and *explicit distrust*. The novelties of our method are the following: (a) it is carefully designed, to take into account positive, implicit, and negative information, (b) it is *scalable* (i.e., linear on the input size), (c) most importantly, it is *effective* and accurate. Our extensive experiments with a real dataset, `Epinions.com` data, of 100K nodes and 1M edges, confirm that PIN-TRUST is scalable and outperforms existing methods in terms of prediction accuracy, achieving up to 50.4 percentage relative improvement.

Keywords

Belief propagation, graph mining, trust prediction

1. INTRODUCTION

In a social network, how can users find other reliable users? In many social network sites, users share various information through interactions with other users. *Trust* plays a vital role for users who seek reliable users in social networks [4, 5, 10, 13, 25]. In particular, in product review sites and e-commerce sites such as `Epinions.com` and `eBay.com`, it is very important to find trustworthy users [3, 22, 24]. In order to find trustworthy users from such a huge social network, *trust prediction* methods, which predict future trust relationships of users, have been proposed [6, 20, 22, 24].

Previous methods predict future trust relationships of a target user based on *interaction information* between users. There exist three types of the interaction information: *explicit trust relationship*, *explicit distrust relationship*, and *ratings* [6, 11, 22]. A user explicitly makes trust or distrust relationships with others when the user decides to trust or distrust them. Also, a user can rate other users' content as well. In many previous methods, ratings are regarded as *implicit trust information* indicating the likelihood of a user to make a trust relationship with another user [20, 22].

Shortcomings of existing methods. The existing methods based on interaction information for trust prediction, have following three problems. First, most previous methods consider only *partial information* of the interaction for trust prediction. The methods in [12, 17, 20, 22, 24] consider the trust relationship and ratings only, but not the distrust relationship. Some methods [11, 18] consider the distrust relationship for trust prediction; however, none of them consider all three types of interaction information together.

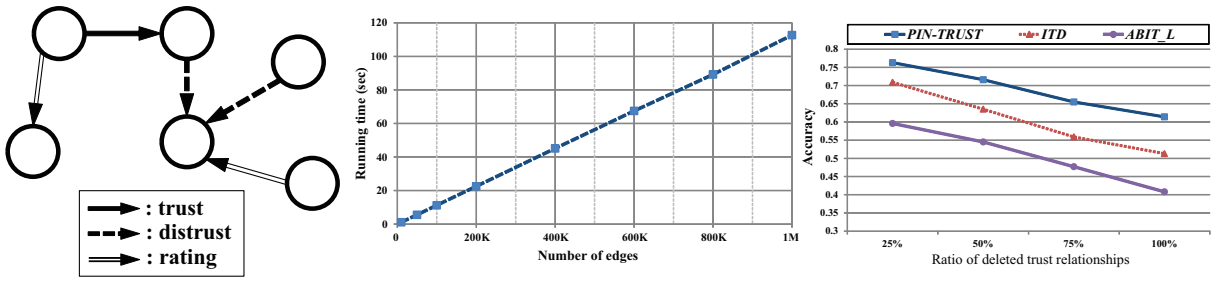
Second, if a target user lacks adequate interaction information, previous methods tend to fail in predicting the target user's trust relationships. Most existing methods rely on link prediction which infers the probability of having a trust relationship between two users based on some criteria (e.g., amount of interaction, number of common neighbors, indirect trust relationships, etc.) [8, 12, 22]. If one of the two users does not have enough interaction information, the previous methods have difficulties in computing the probability between the two users, causing low prediction accuracy for the target user.

Third, previous methods do not take *trust reciprocation* into account. The norm of the trust reciprocation is that people should help those who help them [7, 23]. For example, if user u_i has helped user u_j, then u_j would help u_i back. In a social network, u_i may trust u_j in the hope of getting the trust back. The trust reciprocation exists in many different social networks and should be exploited in trust prediction [21, 23].

*Corresponding author

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983753



(a) Our PIN model. (b) Scalability. (c) Effectiveness.

Figure 1: Advantage of our method. Our method captures all types of interaction information (see (a)), is scalable (see (b)), and outperforms competitors (see (c)).

Our main idea. In this paper, to solve the problems mentioned above, we propose PIN-TRUST, a novel trust prediction method. The proposed method first builds a trust network that has *two types of directed edges* to consider *all kinds* of interaction information. The *positive edge* is created when a user makes a positive relationship (e.g., trust relationship and rating) with another user. The *negative edge* is created when a user makes a distrust relationship with another user. Previous study revealed that negative relationships would be helpful in analyzing and predicting the behaviors of users [16]. By exploiting all three types of interaction information among users in trust prediction, we can alleviate the lack of information successfully, thereby improving the prediction accuracy.

PIN-TRUST measures the degree of trustworthiness of each user for a target user by using "network effects", specifically, *belief propagation* (BP) [26]. We propose message passing strategies that propagate different trust (or distrust) messages according to interaction information between two users. Also, the proposed method defines a new notion of *reverse edges* having the opposite direction of the normal edges to consider the *trust reciprocation*. The proposed method computes the degree of trustworthiness of each node for the target node in a network, and then returns the top-k most trustworthy nodes as a prediction result for the target node.

We summarize our main contributions as follows:

- **PIN Model**: We propose PIN-TRUST, a novel method to predict a target user's future trust relationships. The proposed method models users' behaviors in a social network to consider all types of interaction information (explicit trust, explicit distrust, and implicit ratings; see Figure 1(a)) and also the trust reciprocation.
- **Scalability**: The time complexity of our PIN-TRUST is *linear* on the network size (Figure 1(b)).
- **Effectiveness**: PIN-TRUST is effective and accurate. Applied on real-world data, PIN-TRUST outperforms existing methods with up to 19.7 and 50.4 percentage relative improvement in prediction accuracy compared with *ITD* [22] and *ABIT_L* [20], respectively (Figure 1(c)).

Table 1 shows a summary of comparisons with existing methods, *ITD* and *ABIT_L*. The existing methods cannot consider the distrust relationship and the trust reciprocation. PIN-TRUST takes all of them into account, thereby increasing the accuracy of trust prediction significantly.

Table 1: Comparisons of trust prediction methods. Our proposed PIN-trust is the most accurate, and satisfies all the desired properties while others do not.

	PIN-TRUST	*ITD*	*ABIT_L*
Trust relationship and rating	√	√	√
Distrust relationship	√	-	-
Trust reciprocation	√	-	-
Scalability	√	√	√
Accuracy (in worst case)	61.4%	51.3%	40.8%

The outline of the paper is as follows: Sections 2 and 3 present the preliminaries and our method in detail, respectively. Section 4 shows the experimental results. Section 5 presents an overview of the related work. Finally, Section 6 summarizes and concludes the paper.

2. PRELIMINARIES

We define our trust prediction problem as follows:

PROBLEM 1 (TRUST PREDICTION). ***Given:***

- *Target user* u_q
- *Set of all users in a social network* $U = \{u_1, u_2, ..., u_n\}$
- *Trust relationship matrix* $T = [t_{ij}]$ *where* t_{ij} *indicates the trust relationship (a binary class) from* u_i *to* u_j
- *Distrust relationship matrix* $D = [d_{ij}]$ *where* d_{ij} *indicates the distrust relationship (a binary class) from* u_i *to* u_j
- *Rating count matrix* $R = [r_{ij}]$ *where* r_{ij} *is the number of ratings from* u_i *to* u_j
- *Average rating matrix* $M = [\mu_{ij}]$ *where* μ_{ij} *is the average score of all ratings from* u_i *to* u_j

Find: *The trustworthiness scores of all other users for the target user* u_q*, and her top-*k *most trustworthy users.*

A user in a social network is able to form three types of interaction with other users: the trust relationship, distrust relationship, and ratings. When a user u_i decides that a user u_j is trustworthy, a trust relationship t_{ij} is formed. On the contrary, when u_i decides u_j to be untrustworthy, a distrust relationship d_{ij} is formed. Also, a user is able to rate someone else's content. Let r_{ij} indicate the number of u_i's ratings on u_j's contents, and μ_{ij} represent the average of all rating scores from u_i to u_j. When U, u_q, and the interaction

information of users are given, trust prediction returns a set of the top-k trustworthy users as a prediction result for u_q. To do this, we measure the degree of trustworthiness of all users in a network based on BP.

BP is an algorithm that infers the state of a node in a network by computing the *belief score* of the node [2, 26]. The belief score of a node in a specific state means the probability that the node is in a specific state. In this paper, two states of a node are defined: <*trustworthy*, *untrustworthy*>. The belief score of a node is computed by exchanging *messages* between nodes. The message is a node's opinion about a neighboring node's possibility of being in a specific state. The opinion is determined by the probabilities, each of which indicates how much the node is likely to be in a state. The probability is inferred by its neighbors' opinions as well. The messages sent from one node to its neighbor are represented as a vector. The elements of the vector are the probabilities of two states of a node mentioned above. Each message that node u_i sends to node u_j is computed as follows:

$$msg_{ij}(x_p) \leftarrow \sum_{x_q \in X} \phi_i(x_q)\psi(x_q, x_p) \prod_{k \in N(i) \setminus j} msg_{ki}(x_q)$$

In this equation, x_p and x_q denote two states. $msg_{ij}(x_p)$ represents the message that u_i sends to u_j, indicating u_i's opinion about u_j's probability of being in state x_p. $\phi_i(x_q)$ denotes a *prior belief* which means the initial probability of u_i being in state x_q. In case the state of a node is already known, a higher prior belief is assigned to the corresponding state. Otherwise, the equal prior beliefs are assigned to both states. The *propagation matrix* ψ transforms a node's incoming messages into outgoing messages. $msg_{ij}(x_p)$ is computed by the product of the messages from u_i's neighbors except u_j. It is computed iteratively for a specific number of times or until the message values converge [26]. After the message computation, the belief scores of each node are computed. The belief scores are represented as a vector with the two states as well. Each belief score, the probability of node u_i in state x_p, is computed as follows:

$$b_i(x_p) = k\phi_i(x_p) \prod_{j \in N(i)} msg_{ji}(x_p),$$

where k is a normalization factor. To measure the degree of trustworthiness of all users in a network, we compute the belief scores of all nodes in the network based on BP.

3. PROPOSED METHOD

In this section, we propose PIN-TRUST, a method to predict trust relationships to be satisfied by a target user. PIN-TRUST employs the notion of the belief propagation (BP) to exploit the "network effects". In order for BP to be used, it is required to (1) model a trust network, (2) define propagation matrices, and (3) decide prior beliefs.

In Section 3.1, we first discuss how to build a trust network based on three kinds of interaction information so that it provides a richer resource in trust prediction. In Section 3.2 and Section 3.3, we discuss how to set propagation matrices and prior beliefs to be used for computing messages. Users tend to make trust relationships with other users who have behavioral characteristics similar to themselves [15]. We set the propagation matrices based on this notion, referred to as "homophily influence". Users tend to trust those users who are trusted by their trustees. For example, if a user u_i trusts a user u_j, and a user u_j trusts a user u_k, user u_i is likely to trust user u_k. Also, users tend to trust those users who are trusted by a number of other users. In Section 3.3, we discuss how to consolidate these two notions by using the prior beliefs in BP.

In addition, there exists reciprocation in trust relationships. Users tend to trust those users who trust them. In Section 3.4, we define a new type of reverse edges to consider the notion of trust reciprocation and discuss how to exploit the reverse edges in the BP settings.

3.1 Trust network modeling

To apply BP to trust prediction, one needs to construct a trust network. We construct a trust network where nodes and directed edges correspond to users and interactions between them, respectively. The direction of an edge is determined by the direction of the interaction represented by the edge. For example, when user u_i forms the trust relationship to user u_j, the direction of the edge is from u_i to u_j.

To consider all types of interaction information, the trust network has two types of edges: the *positive* and *negative edges*. A positive edge is created when a user makes a *trust* relationship with another user and/or *rates* her content. A negative edge is created when a user makes a *distrust* relationship with another user. It means that we regard the trust relationship and rating as the positive information, and the distrust relationship as the negative information. The trust and distrust relationships can be classified as *explicit* information since these relationships are declared *explicitly* by the user.

The rating is an expression of an *implicit trust*. Although a trust degree varies depending on the number of ratings and the average rating score between two users, the rating is regarded as *implicit positive information* in many existing methods [22, 20, 14]. For example, if u_i has given a number of good ratings to the content made by user u_j, u_i is likely to make a trust relationship with u_j. For this reason, we consider the ratings as positive information and create a positive edge when a user rates another user's content. Note that, although we make positive edges for ratings, the trust degree is dependent on the quality of the ratings: only good and many ratings affect the trust significantly. We discuss how to estimate the trust degree between u_i and u_j with respect to the varying number of ratings r_{ij} and the average rating score μ_{ij} in Section 3.2.

3.2 Propagation matrices

A user's belief scores on states should vary considerably depending on the interaction information that the user receives. To the end, we define two different *propagation matrices* for the positive and negative edges in order to propagate different messages according to the interaction information between users. The proposed propagation matrices are based on the concept of "*homophily influence*". Homophily refers to the tendency of a node to associate itself with its similar nodes in a network [15]. For example, in a trust network, a trustworthy user wants to establish trust relationships with other trustworthy users.

Table 2 indicates instantiations of the propagation matrices ψ for the positive and negative edges. In $\psi_{ij}(x_q,x_p)$, the row (source) represents x_q and the column (destination)

Table 2: Instantiations of propagation matrices.

(a) Positive edge

Positive edge		Destination: Trust.	Destination: Untrust.
Source	Trust.	$0.5+ \varepsilon\times(t + f(r))$	$0.5 - \varepsilon\times(t + f(r))$
	Untrust.	$0.5 - \varepsilon\times(t + f(r))$	$0.5+ \varepsilon\times(t + f(r))$

(b) Negative edge

Negative edge		Destination: Trust.	Destination: Untrust.
Source	Trust.	$0.5 - \varepsilon$	$0.5 + \varepsilon$
	Untrust.	$0.5+ \varepsilon$	$0.5 - \varepsilon$

represents x_p. We want to instantiate the propagation matrices so that 1) u_j's degree of trustworthiness should increase when trustworthy user u_i makes a positive relationship with u_j, and 2) if u_i is untrustworthy, then u_j's degree of trustworthiness ought to decrease. In the positive edge, the probability $\psi_{ij}(trustworthy, trustworthy)$ of a destination node u_j being in state *trustworthy* when a source node u_i is in state *trustworthy* is $0.5 + \varepsilon \times (t+f(r))$. On the other hand, $\psi_{ij}(untrustworthy, trustworthy)$ is $0.5 - \varepsilon \times (t+f(r))$. Below, we describe the meanings of the parameters and explain detailed reasons for the instantiations.

ε controls the influence of the source on the destination. t is a binary parameter to represent the existence of a trust relationship between them. If a trust relationship exists, $t = 1$; otherwise $t = 0$. $f(r)$ is the trust degree measured by the number of ratings r_{ij} and the average rating scores μ_{ij} from source node u_i to destination node u_j. Many good ratings can be regarded as the implicit trust between the two users. Therefore, as u_i gives more/higher ratings to u_j, u_i is more likely to trust u_j. To measure $f(r)$ quantitatively, we use a data-driven estimation [22, 20]. We examine the correlation between the number of ratings and the existence of a trust relationship in a user pair by using the `Epinions.com` dataset which has been widely used in many existing trust prediction methods [8, 12, 17] and trust-based recommendation systems [9, 27]. In the dataset, ratings are given as an integer value from 1 to 5.

Figure 2 shows the result. In Figure 2, the x-axis indicates the number r_{ij} of ratings in a user pair and the y-axis indicates the probability of the user pair having a trust relationship. The x-axis is divided into nine intervals with respect to varying r_{ij}. Each interval of r_{ij} is also subdivided into two depending on the range of the average rating μ_{ij} ($\mu_{ij} \leq 3$ and $\mu_{ij} > 3$). As shown in the figure, when $\mu_{ij} \leq 3$, trust relationships rarely occur in all r_{ij} intervals. On the contrary, when $\mu_{ij} > 3$, the probability of having trust relationships increases as r_{ij} increases. This observation indicates that the trust degrees should be measured by taking both r_{ij} and μ_{ij} into account. Based on the observation, the proposed method computes $f(r)$ as shown in Table 3. It implies that the more frequently u_i rates u_j's content favorably, the higher belief on the trustworthy state is sent from u_i to u_j by the propagation matrix in Table 2(a).

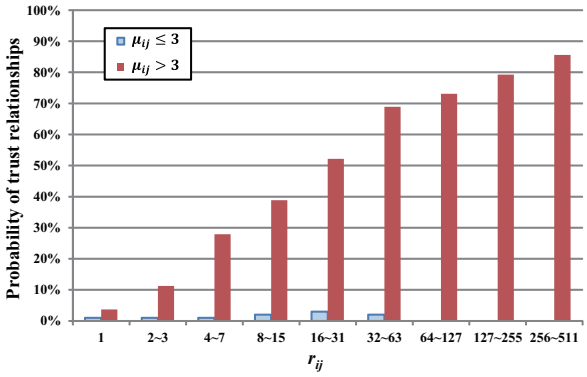


Figure 2: Probability of having trust relationship with varying r_{ij} and μ_{ij}.

Table 3: Estimation of trust degree with the number of ratings and the average rating score between two users.

r_{ij}	f(r): $\mu_{ij} \leq 3$	f(r): $\mu_{ij} > 3$
1	0.01	0.05
2-3	0.01	0.10
4-7	0.01	0.30
8-15	0.02	0.40
16-31	0.03	0.55
32-63	0.02	0.70
64-127	0	0.75
127-255	0	0.80
256-511	0	0.85

In the propagation matrix of the *negative edge*, $\psi_{ij}(trustworthy, trustworthy)$ is set to $0.5 - \varepsilon$. This is because normal users usually want to avoid the interactions with untrustworthy users [1]. On the contrary, $\psi_{ij}(untrustworthy, trustworthy)$ is set to $0.5 + \varepsilon$ since we assume that the untrustworthy user tries to debase normal users' trustworthiness. By using these propagation matrices, PIN-TRUST propagates different messages in accordance with a variety of interaction information between two users.

3.3 Prior beliefs

After setting the propagation matrices, we set the prior beliefs of each node. As mentioned in Section 2, the prior belief indicates the initial probability that the node is in a state among the two states: <*trustworthy*, *untrustworthy*>. We consider the following two aspects in setting the prior beliefs for nodes.

Target user's opinion. A user makes trust relationships based on her own opinions on other users. We assume the user's current interaction information reflects her opinions. For example, if u_i favorably rates u_j's content, u_i is highly likely to form the trust relationship with u_j. Also, a user trusts her trustworthy neighbors' opinions. If u_i trusts u_j and u_j trusts u_k, u_i is likely to trust u_k. Therefore, the node nearer to the target node via a series of positive edges should receive a higher degree of trustworthiness.

Majority's choice. Normally, we consider a user to be trustworthy if many other users trust her/him. That is, user u_i is likely to trust u_j when u_j is trusted by a number

Table 4: Parameters in PIN-trust.

Parameter	Definition
$0.5 + \alpha$	Prior belief of a target node
$0.5 + \beta$	Prior beliefs of the other nodes
ε	Influence in propagation matrices
$f(r)$	Trust degree based on rating information
RF	Factor to adjust influence of trust reciprocation

of other users. Therefore, a node that receives many trust relationships should get a high degree of trustworthiness.

In the BP computation, when a prior belief on the trustworthy state of node u_i is high, the message from the trustworthy state of u_i would be also high. In addition, because propagation matrices are defined based on the notion of "homophily influence", u_i's message on the trustworthy state to a node connected to u_i by the positive edge would be high. Consequently, the node is likely to be considered as the trustworthy state.

As mentioned earlier, in order to increase the degree of trustworthiness of a node (directly or indirectly) connected to the target node, we give a prior belief to the target node on a trustworthy state higher than on an untrustworthy state. That is, the prior beliefs to the target node are set as $<0.5+\alpha, 0.5-\alpha>$ for trustworthy and untrustworthy states, respectively. Since the prior belief on a trustworthy state is higher than that of an untrustworthy state, nodes close to the target node tend to get a high belief score on a trustworthy state as well.

In addition, in order to give high trustworthiness to a node that a number of nodes commonly trust, we set the prior belief for every node (except for the target node) on trustworthy state higher than that on an untrustworthy state. Specifically, the prior beliefs of every node (except the target node) are set as $<0.5+\beta , 0.5-\beta>$. As a result, the node receiving many trust relationships is likely to get a high belief score on a trustworthy state[1]. We note α is set bigger than β (i.e., $\alpha \gg \beta$) in order for the target node's opinion to be considered more than that of other nodes. Their detailed settings will be discussed again in Section 4.2.1.

3.4 Trust reciprocation

Trust reciprocation appears in existing social networks and plays an important role in trust prediction [21, 23]. If user u_i favorably evaluates user u_j, u_j is also likely to give favorable evaluation back to u_i. Likewise, if u_i evaluates u_j unfavorably, u_j is likely to evaluate u_i unfavorably as well.

To consider the trust reciprocation, we define a notion of *reverse edges* that have the opposite directions of their corresponding normal edges. Figure 3 shows an example of reverse edges. When u_1 trusts (and/or rates positively) u_2, a *reverse positive edge* is created from u_2 to u_1. Also, since u_1 distrusts u_3, a *reverse negative edge* is created from u_3 to u_1.

Based on the concept of the trust reciprocation, we also define two propagation matrices for the reverse edges. Table 5 shows instantiations of the propagation matrices of reverse edges. Notice that source and destination nodes are determined by considering the direction of *a normal edge rather than a reverse edge*. As a result, in the case of a reverse edge, a message is propagated from a destination to a source. Contrary to the propagation matrices in Table 2, in $\psi_{ij}(x_q,x_p)$, x_q represents the state of the destination node on the normal edge (row), and x_p represents the state of the source node on the normal edge (column). The concept of the trust reciprocation is "if you do something to me, then I'll do a similar thing to you". Thus, the propagation matrices for the reverse edges are similarly set as those for the normal edges. The matrices have RF, a parameter that controls the influence of trust reciprocation. RF is needed since all users do not always return the compliment.

[1]A user who receives many distrust relationships eventually would get a very low belief score on a trustworthy state through BP computations.

Table 5: Propagation matrices of reverse edges.

(a) Reverse positve edge

Positive		Source: Trust.	Source: Untrust.
Destn.	Trust.	$0.5+ \varepsilon\times(t + \mathrm{f}(r)) \times RF$	$0.5 - \varepsilon\times(t + \mathrm{f}(r)) \times RF$
	Untrust.	$0.5 - \varepsilon\times(t + \mathrm{f}(r)) \times RF$	$0.5+ \varepsilon\times(t + \mathrm{f}(r)) \times RF$

(b) Reverse negative edge

Negative		Source: Trust.	Source: Untrust.
Destn.	Trust.	$0.5 - \varepsilon \times RF$	$0.5+ \varepsilon \times RF$
	Untrust.	$0.5+ \varepsilon \times RF$	$0.5 - \varepsilon \times RF$

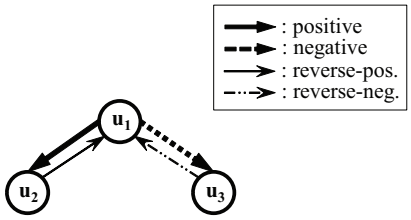


Figure 3: An example of reverse edges.

PIN-TRUST computes the belief scores of every node in a network using the strategies above, and returns the top-k nodes having the highest belief scores on the trustworthy state as a prediction result for the target user. Algorithms 1 and 2 show the pseudocode of PIN-TRUST and its subfunctions.

4. EXPERIMENTS

In this section, we perform extensive experiments to show the effectiveness and efficiency of PIN-TRUST. Basically, we design our experiments to answer the following questions:

- Q1 (Parameter): How robust is PIN-TRUST over the values of four parameters α, β, ε, and RF?
- Q2 (Distrust): How accurate is PIN-TRUST by modeling the distrust relationships?
- Q3 (Accuracy): How accurate is PIN-TRUST in comparison with existing methods?
- Q4 (Scalability): Is PIN-TRUST scalable?

Algorithm 1: PIN-TRUST

Input:
$U = \{u_1, u_2, ..., u_u\}$: Set of users (nodes)
$S = \{Trustworthy, Untrustworthy\}$: Set of possible states of nodes
P: Adjacency matrix with positive edges
N: Adjacency matrix with negative edges
PR: Adjacency matrix with reverse positive edges
NR: Adjacency matrix with reverse negative edges
$A = \{P, N, PR, NR\}$: Set of adjacency matrices
ϕ: Prior beliefs of each node
$\psi = \{\psi_P, \psi_N, \psi_{PR}, \psi_{NR}\}$: Set of propagation matrices for each adjacency matrix

Output:
Belief score matrix B

```
begin
    B = NewMatrix(U, S)
    foreach x ∈ A do
        MSG = NewMatrix(U, U)
        while not converged do
            foreach i ∈ U do
                msg = NewMessage(S)
                foreach k ∈ Neighbors(x, i) do
                    msg = MultiplyMessage(msg,
                      MSG[k][i], S)
                foreach j ∈ Neighbors(x, i) do
                    msg_j = DivideMessage(msg,
                      MSG[j][i], S)
                    MSG[i][j] = PropagateMessage(i,
                      msg_j, φ, x, S)
        foreach i ∈ U do
            msg = NewMessage(S)
            foreach j ∈ Neighbors(x, i) do
                msg = MultiplyMessage(msg,
                  MSG[j][i], S)
            foreach p ∈ S do
                msg[p] = msg[p] × φ_i[p]
            msg = NormalizeMessage(msg, S)
            foreach p ∈ S do
                B[i][p] = msg[p]
```

Algorithm 2: Functions

```
function NewMatrix(U, S):
begin
    return a matrix with ||U|| rows and ||S|| columns,
      having each element filled with 1

function NewMessage(S):
begin
    return a vector, having ||S|| elements filled with 1

function Neighbors(x, i):
begin
    return a set of neighbors of node i in the adjacency
      matrix x

function MultiplyMessage(msg, MSG[k][i], S):
begin
    foreach p ∈ S do
        msg[p] = msg[p] × MSG[k][i][p]
    return msg

function DivideMessage(msg, MSG[j][i], S):
begin
    foreach p ∈ S do
        msg[p] = msg[p] ÷ MSG[j][i][p]
    return msg

function PropagateMessage(i, msg_j, φ, x, S):
begin
    msg = NewMessage(S)
    sum = 0
    foreach p ∈ S do
        foreach q ∈ S do
            sum = sum + (φ_i[q] × ψ_x[q][p] × msg_j[q])
        msg[p] = sum
    return msg

function NormalizeMessage(msg, S):
begin
    sum = 0
    foreach p ∈ S do
        sum = sum + msg[p]
    foreach p ∈ S do
        msg[p] = msg[p] ÷ sum
    return msg
```

In Section 4.1, we describe the experimental setup. In Section 4.2, we present and analyze the results of experiments to answer the above questions.

4.1 Experimental setup

In our experiments, the `Epinions.com` dataset is used. It contains 131,828 users, 717,667 trust relationships, 123,705 distrust relationships, and 13,668,319 ratings; the ratings have integer values from 1 to 5. We measure the accuracy and the processing time of the proposed method. Target users are divided into three types of groups according to the amount of interaction information as follows:

- User group 1: Users who have rated more than 100 other users, or have more than 50 trust relationships
- User group 2: Users who have rated 50 to 100 other users, or have 20 to 50 trust relationships
- User group 3: Users who have rated 20 to 50 other users, or have 10 to 20 trust relationships

In our experiments, we compare the accuracy of the proposed method with those of two existing methods, *ITD* [22] and *ABIT_L* [20]. The accuracy of each method is computed as follows. First, we randomly select a target user in each user group; we delete the target user's n existing trust relationships (n = 25%, 50%, 75%, 100% of the total); the proposed method computes the belief score of every node, and the existing methods compute the probability of making trust relationships between the target node and every other node in the network; for each method, the scores computed for all the nodes are sorted in the descending order and the top-k nodes from the sorted list are selected as prediction results; the accuracy of each method is computed by comparing its predicted result with the ground truth, i.e., the

Table 6: Parameters and their values.

Parameters	Values
α	4×10^{-1}, 10^{-1}, $\mathbf{10^{-2}}$, 10^{-3}, 10^{-4}
β	10^{-3}, 10^{-4}, $\mathbf{10^{-5}}$, 10^{-6}, 0
ε	10^{-1}, 5×10^{-2}, 10^{-2}, $\mathbf{5\times10^{-3}}$, 10^{-3}
RF	1, $\mathbf{10^{-1}}$, 5×10^{-2}, 10^{-3}, 0

deleted trust relationships. The accuracy is measured by the ratio of correct predictions to all the predictions (i.e., precision), which has been used in many prediction domains [22, 20].

Our parameters can be flexibly tuned in order to reflect users' interactions. Table 6 gives such parameters: α, β, ε, and RF. We examine the accuracy with respect to different parameter values. While a parameter in Table 6 changes for different experimental settings, other parameters are fixed to the pivot values in boldface. For example, during our experiments are performed with different α of 4×10^{-1}, 10^{-1}, 10^{-2}, 10^{-3}, and 10^{-4}, other parameters, i.e., β, ε, and RF, are fixed to be 10^{-5}, 5×10^{-3}, and 10^{-1}, respectively.

We use the total elapsed time as the measure of a processing time while excluding the time for file accesses. For accurate experimental results, we take the average of times for getting 100 prediction results. The experiments are performed on an I-5 2.50GHZ PC equipped with a Windows7 OS and 8GB of main memory.

4.2 Results and Analyses

4.2.1 Q1: Parameter

In the first set of experiments, we show the accuracy of PIN-TRUST with respect to varying parameter values to verify its robustness. Figure 4 indicates the results. In the experiments, we used user group 2 as a target user group because it has a moderate amount of interaction information. Figure 4(a) depicts the accuracy changes according to varying prior beliefs of the target node. The results reveal that PIN-TRUST has the highest accuracy when $\alpha = 10^{-2}$. The accuracies are also kept high when $10^{-1} \geq \alpha \geq 10^{-3}$.

The accuracy of the proposed method decreases when $\alpha > 10^{-1}$. This is because the higher prior belief on the trustworthy state makes the trustworthy message too strong. If the target user's prior belief on the trustworthy state is too high, the user propagates too high trustworthy message to her/his neighbors. As a result, the neighbors get high belief scores on the trustworthy state regardless of the interaction information with the target user. For example, even if user u_j receives a low average rating score ($\mu_{ij} \leq 3$) from the target user, u_j eventually would get a high trustworthy belief score due to the high prior belief of the target user.

The proposed method shows the lowest accuracy when $\alpha = 10^{-4}$. If the target user's prior belief on the trustworthy state is too low, the target user propagates too low trustworthy messages to her/his neighbors as well. It does not make a significant difference between the target user's messages and the other users' messages. In BP, the messages represent the user's opinions. That is, when $\alpha = 10^{-4}$, it is difficult to consider the target user's opinions more than those of the other users, and thus the prediction accuracy decreases rapidly. For this reason, we set α as 10^{-2} in the following experiments.

Figure 4(b) shows the accuracy results with respect to varying prior beliefs of all the nodes other than the node for a target user. As shown in the figure, the proposed method provides high accuracies when $10^{-4} \geq \beta \geq 10^{-6}$. When $\beta = 0$, the accuracy slightly decreases. In the case, the prior beliefs of each node are assigned as unbiased states <0.5, 0.5> and thus, the messages of each node are also assigned as unbiased states (except for the target node). Consequently, we cannot consider the other users' opinions when we use $\beta = 0$.

The results imply that a user basically decides to establish trust relationships with others based not only on her/his own opinions but also on other users' opinions. If β is bigger than 10^{-4}, the accuracy decreases with the increase of β. It is an inevitable result since the bigger β makes the smaller difference between the target user' message and the other user's messages. In the following experiments, we set β to 10^{-5}.

Figure 4(c) indicates the accuracy results with the respect to varying values of ε, which controls the influence of the message in the propagation matrices. The results show that the proposed method provides high accuracies when $10^{-2} \geq \varepsilon \geq 10^{-3}$. The accuracy decreases with the increase of ε. This is because the higher ε makes the higher influence of the message propagation. The messages of each node eventually affect the belief scores of distant nodes in the network, thereby decreasing the accuracy of the trust prediction. We set ε as 5×10^{-3} in the following experiments.

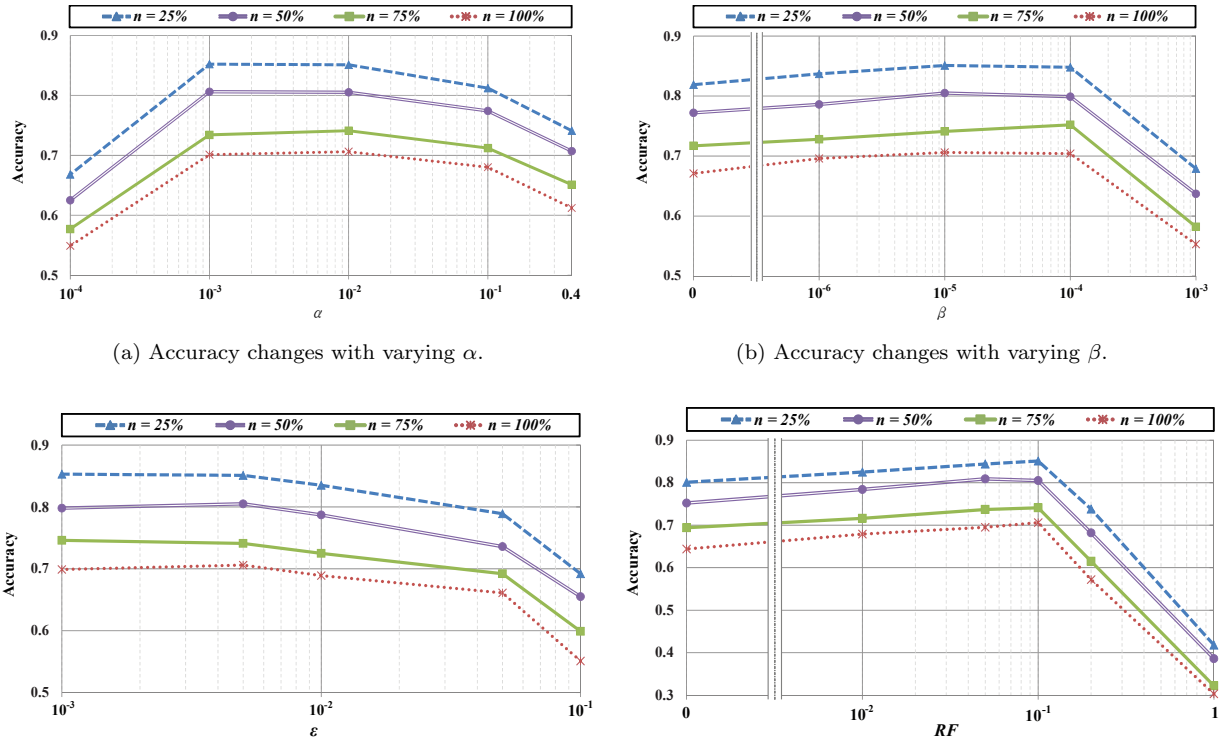
Lastly, Figure 4(d) shows the accuracy results according to varying values of RF, which controls the influence of the trust reciprocation on reverse edges. When $RF \approx 10^{-1}$, the proposed method shows the highest accuracy. Similar accuracies are also found when $RF \geq 10^{-2}$.

On the contrary, when $RF = 1$, the result shows the lowest accuracy: in the case, the messages on the reverse edges have the same influence as those on the normal edges. Therefore, all other nodes that trust the target node unconditionally would receive high trustworthy messages through the reverse edges from the target.

When $RF = 0$, the accuracy is lower than those when $10^{-1} \geq RF \geq 10^{-2}$ because the trust reciprocation is not considered at all. Since the trust reciprocation exists in many social networks, we need to consider it for trust prediction [21, 23]. We use RF of 10^{-1} to consider the trust reciprocation in the following experiments.

4.2.2 Q2: Distrust

In the second set of experiments, we measure the accuracy of PIN-TRUST with and without the distrust relationships in order to check whether the use of the distrust relationships is beneficial to accurate trust prediction. To measure the accuracy without employing them, we simply deleted all distrust relationships in the network. Table 7 shows the results. The accuracy of PIN-TRUST with distrust relationships is slightly better than the accuracy without them. Compared to other parameters above, the distrust relationship seems to be a less dominant factor to accuracy improvement. This is because, in `Epinions.com`, *users cannot see other users' distrust relationships*. Users' positive information such as the trust relationships and ratings can be seen to other users in `Epinions.com`, affecting other users' future trust relationships. On the contrary, users' distrust relationships cannot be seen to other users, so the distrust relationships cannot



(a) Accuracy changes with varying α.

(b) Accuracy changes with varying β.

(c) Accuracy changes with varying ε.

(d) Accuracy changes with varying RF.

Figure 4: Accuracy changes according to different values of parameters: PIN-trust shows its robustness against varying parameter values. Optimal parameter choices are $\alpha = 10^{-2}$, $\beta = 10^{-5}$, $\varepsilon = 5 \times 10^{-3}$, and $RF = 10^{-1}$.

Table 7: Distrust information helps: Accuracy results with and without distrust relationships.

	n=25%	*n*=50%	*n*=75%	*n*=100%
PIN-TRUST with distrust relationships	0.851	0.805	0.741	0.706
PIN-TRUST without distrust relationships	0.851	0787	0.720	0.673

affect other users' future relationships a lot. In other words, the distrust relationships of other nodes are not propagated over the network.

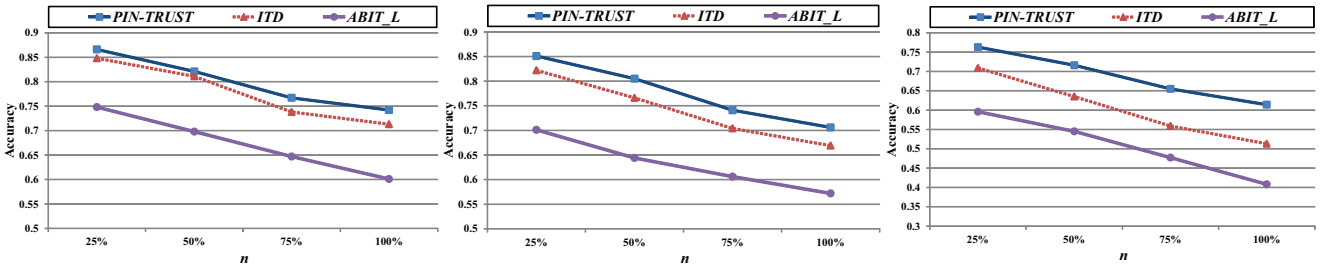
However, in case the distrust relationship is not used at all, users who received the distrust relationship from the target user may have the high trustworthy belief score through the indirect trust relationships. For example, a target user u_i trusts u_j and distrusts u_k, and also u_j trusts u_k because of her/his personal preferences. In this situation, if the distrust relationship is not used, u_k eventually would get a high trustworthy belief score through the indirect trust relationship from u_i. Using the distrust relationships is effective in eliminating such a problematic situation in advance, leading to higher prediction accuracy.

4.2.3 Q3: Accuracy comparison

In the third set of experiments, we compare the accuracy of PIN-TRUST with that of existing methods, *ITD* [22] and *ABIT_L* [20], with target users selected in each user group. Figure 5 shows the results of the accuracy compared. The results indicate that the proposed method provides the best accuracy in all cases.

Figure 5(a) shows the accuracy comparisons with user group 1. The accuracy of *ITD* is comparable to that of the proposed method. *ITD* computes the probability of making a trust relationship between two users by using both the trust relationship and ratings. Since user group 1 has an enough amount of interaction information, *ITD* works well in this case. However, since *ITD* does not consider the trust reciprocation and the distrust relationship, PIN-TRUST shows a slightly better accuracy. *ABIT_L* shows much lower accuracy than the others since it mainly focuses on the rating information between two users to infer the trust relationship [20].

Figure 5(b) shows the accuracy comparisons with user group 2. The accuracy gaps between PIN-TRUST and *ITD* are higher than those with user group 1. As shown in Section 4.1, user group 2 has a less amount of interaction information than user group 1. *ITD* regards two users to be in a trust relationship only if the probability level between the two users exceeds a predefined threshold [22]. If a target user does not have much interaction information, *ITD* has a difficulty in computing the probability level between the target and other users, thus providing a low accuracy. In contrast, even when the target user has small bits of interaction information, PIN-TRUST is able to compute the belief scores of all the nodes based on the proposed propagation strategies.



(a) User group 1. (b) User group 2. (c) User group 3.

Figure 5: Accuracy comparisons: PIN-trust universally shows better accuracy than existing methods. With the decrease in the amount of interaction information, the accuracy gaps between ours and the others increase.

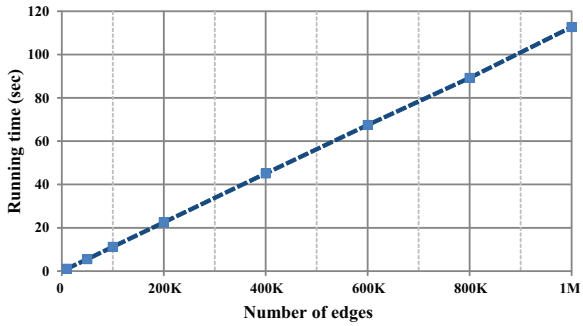


Figure 6: Processing time of PIN-trust: It is scalable with respect to the number of edges.

With the decrease in the amount of interaction information, the accuracy gaps increase even more. Figure 5(c) shows the accuracy comparisons with user group 3. Since user group 3 has the least amount of interaction information, the accuracy of all methods get decreased. The proposed method shows the best accuracy. Compared with *ITD* and *ABIT_L*, PIN-TRUST improves the prediction accuracy by up to 19.7 and 50.4 percentage relative improvements.

4.2.4 Q4: Scalability

Lastly, we show how the processing time changes as the number of edges in a network increases. Figure 6 shows the experimental results. The processing time grows linearly with the increasing number of edges. Note that PIN-TRUST is based on BP whose computational complexity is $O(|E|\, d^2 c)$ [1], where $|E|$ represents the number of edges in the network, d is the number of states of a node, and c is the number of iterations for convergence. In PIN-TRUST, we set $d = 2$ and $c = 30$. Consequently, the processing time of PIN-TRUST is linear on the network size in terms of the number of edges.

5. RELATED WORK

Recently, several methods on trust prediction have been proposed. Most existing methods are based on link prediction by regarding infers the trust relationship between two users as a link. They can be categorized into two classes: supervised prediction and unsupervised prediction [24, 8].

The supervised prediction method first extracts a set of features from every user pair in a network and infers the existence of the trust relationship by training a binary classifier. *Nguyen et al.* [20] used the trust antecedent framework (TAF) model, developed in the management and social science domain, to extract features for trust prediction. Based on the TAF model, they extracted a set of quantitative features: ability, benevolence, integrity, and trust propensity. *Chua et al.* [8] also used the TAF model to extract four key features: ability, propensity, expressiveness, and observability. Using the features, they predicted both trust and distrust relationships. *Ma et al.* [17] extracted two types of key features: ten features for user attributions and nine features for user interactions. These features were used in classification methods to construct a trust prediction model. Lastly, *Matsuo and Yamamoto* [19] explained the bidirectional effect of both trust relationships and ratings, and extracted seventy nine features used for trust prediction.

The unsupervised prediction method first computes the probability of making a trust relationship in each user pair based on some criteria and then ranks them in the order of the probability thus computed. *Oh et al.* [22] developed a trust prediction framework based on trust-message passing strategies, taking into account both trust relationships and ratings. In the framework, three types of trust prediction models are built: initial probability model, transpose trust model, and direct propagation model. *Huang et al.* [12] proposed a joint manifold factorization (JMF) model to predict both trust and distrust relationships. The JMF model explores a user group similarity to consider the social influence between users and then alleviates the sparsity problem in a trust network. *Tang et al.* [24] developed an approach to exploit the homophily effect in trust relationships to alleviate the sparsity problem as well. The homophily effect indicates that more similar users have a higher likelihood to trust one another. *Guha et al.* [11] proposed four strategies of trust relationship propagation: direction propagation, transpose trust, co-citation, and trust coupling. They also discussed distrust relationship propagation for trust prediction. *Massa and Avesani* [18] developed trust propagation strategies that incorporate trust and distrust relationships for trust prediction.

6. CONCLUSIONS

In this paper, we proposed PIN-TRUST, a novel method to predict a target user's future trust relationships. The proposed method measures each user's trustworthiness over other users in a network by using all three types of interaction information: the trust relationships, distrust relationships, and ratings. We described our message passing strategies which propagate different messages in accordance

with interaction information between two users. We defined the notion of reverse edges to consider the trust reciprocation and discussed message propagation strategies for the reverse edges as well. Our main contributions are summarized as follows:

- **PIN Model**: The proposed method is carefully designed to consider all types of interaction information and also the trust reciprocation.
- **Scalability**: The proposed method is scalable on the network size. We show that the processing time of PIN-TRUST grows linearly with the number of edges in the network.
- **Effectiveness**: The proposed method outperforms existing methods in terms of accuracy. Applied on real-world data, PIN-TRUST improves the prediction accuracy by up to 19.7 and 50.4 percentage compared with *ITD* and *ABIT_L*, respectively.

The proposed method, PIN-TRUST, is a modified belief propagation that is applied to a graph consisting of different types of edges and uses a different propagation matrix for each type of an edge. We showed this modification is practically effective in trust prediction. As a future work, we are going to study theoretical implications of the modification. In addition, we plan to extend our work to predict trust relationships of cold-start users and also to infer distrust relationships.

7. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A1A10054151).

8. REFERENCES

[1] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *AAAI ICWSM*, pages 2–11, 2013.

[2] F. Altarelli, A. Braunstein, L. Dall'Asta, A. Lage-Castellanos, and R. Zecchina. Bayesian inference of epidemics on networks via belief propagation. *Phys. Rev. Lett.*, 112(11):118701, 2014.

[3] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: An axiomatic approach. In *WWW*, pages 199–208, 2008.

[4] E. Ayday and F. Fekri. Iterative trust and reputation management using belief propagation. *IEEE TDSC*, 9(3):375–386, 2012.

[5] A. Benczur, K. Csalogany, and T. Sarlos. Link-based similarity search to fight web spam. In *AIRWEB*, 2006.

[6] L. Capra and M. Musolesi. Autonomic trust prediction for pervasive systems. In *IEEE AINA*, pages 481–488, 2006.

[7] J. Cheng, D. Romero, B. Meeder, and J. Kleinberg. Predicting reciprocity in social networks. In *IEEE PASSAT/SocialCom*, pages 49–56, 2011.

[8] F. Chua and E. Lim. Trust network inference for online rating data using generative models. In *ACM KDD*, pages 889–898, 2010.

[9] R. Forsati, I. Barjasteh, F. Masrour, A. Esfahanian, and H. Radha. Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In *ACM RecSys*, pages 51–58, 2015.

[10] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. In *ACM WSDM*, pages 241–250, 2010.

[11] R. Guha, R. Kumar, and P. Raghavan. Propagation of trust and distrust. In *WWW*, pages 403–412, 2004.

[12] J. Huang, F. Nie, H. Huang, and Y. Tu. Trust prediction via aggregating heterogeneous social networks. In *ACM CIKM*, pages 1774–1778, 2012.

[13] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *ACM RecSys*, pages 135–142, 2010.

[14] N. Korovaiko and A. Thomo. Trust prediction from user-item ratings. *Social Network Analysis and Mining*, 3(3):749–759, 2013.

[15] D. Koutra, T. Ke, U. Kang, D. Chau, H. Pao, and C. Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *ECML PKDD*, pages 245–260, 2011.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *ACM CHI*, 2010.

[17] N. Ma, E. Lim, V. Nguyen, A. Sun, and H. Liu. Trust relationship prediction using online product review data. In *ACM CNIKM*, pages 47–52, 2009.

[18] P. Massa and P. Avesani. Controversial users demand local trust metrics: an experimental study on epinions.com community. In *AAAI*, pages 121–126, 2005.

[19] Y. Matsuo and H. Yamamoto. Community gravity: Measuring bidirectional effects by trust and rating on online social networks. In *WWW*, pages 751–760, 2009.

[20] V. Nguyen, E. Lim, J. Jiang, and A. Sun. To trust or not to trust? predicting online trusts using trust antecedent framework. In *IEEE ICDM*, pages 896–901, 2009.

[21] V. Nguyen, E. Lim, H. Tan, J. Jiang, and A. Sun. Do you trust to get trust? a study of trust reciprocity behaviors and reciprocal trust prediction. In *SDM*, pages 72–83, 2010.

[22] H. Oh, J. Kim, S. Kim, and K. Lee. A probability-based trust prediction model using trust-message passing. In *WWW*, pages 161–162, 2013.

[23] A. Singhal, K. Subbian, J. Srivastava, T. Kolda, and A. Pinar. Dynamics of trust reciprocation in multi-relational networks. In *IEEE/ACM ASONAM*, pages 661–665, 2013.

[24] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *ACM WSDM*, pages 53–62, 2013.

[25] J. Tang, H. Gao, H. Liu, and A. Sarma. Etrust: Understanding trust evolution in an online world. In *ACM KDD*, pages 253–261, 2012.

[26] J. Yedidia, W. Freeman, and Y. Weiss. *Understanding belief propagation and its generalizations*. Morgan Kaufmann Publishers Inc., 2003.

[27] H. Zou, Z. Gong, N. Zhang, W. Zhao, and J. Guo. Trustrank: A cold-start tolerant recommender system. *Enterprise Information Systems*, 9(2):117–138, 2015.