PatTrust: A Pattern-based Evaluation Approach for Trust and Distrust in Internetware

Yuan Yao
State Key Laboratory for Novel Software Technology, Nanjing University, China
yyao@smail.nju.edu.cn

Feng Xu
State Key Laboratory for Novel Software Technology, Nanjing University, China
xf@nju.edu.cn

Yiling Yang
State Key Laboratory for Novel Software Technology, Nanjing University, China
ylyang@smail.nju.edu.cn

Jian Lu
State Key Laboratory for Novel Software Technology, Nanjing University, China
lj@nju.edu.cn

ABSTRACT

Finding reliable services by trustworthiness evaluation is necessary for Internetware and other Web-based systems to function properly. In these environments, distrust is at least as important as trust. However, existing evaluation approaches either ignore distrust or treat it in a fixed way. In view of this, we propose a novel pattern-based approach *PatTrust* for trustworthiness evaluation. The advantage of this approach is the flexibility in handling both trust and distrust, which is achieved by introducing the *trust pattern* concept. Based on trust patterns, we propose three trust models derived from social psychology theories. Through a set of experiments on a real data-set, we demonstrate the flexibility and effectiveness of our proposed approach. Furthermore, the experimental results also show some evidence for studied theories, such as "the enemy of my enemy is my friend".

Keywords

Trust and Distrust, Trust Pattern, Trustworthiness Evaluation, Service Selection, Internetware

1. INTRODUCTION

In Internetware, services need to be composed to complete a specific task [19, 18]. Consequently, selecting reliable services as components is necessary for the whole system to function properly. Such service selection is especially difficult when few direct experiences of the specific service are available. To this end, researchers have proposed to select reliable services by evaluating the trustworthiness of the service providers for Internetware and other Web-based systems, based on the trust or distrust statements between participants over the system [13, 28].

In these environments where trust plays a critical role, many approaches for trustworthiness evaluation have been proposed, including *recommender-based approaches* [25, 23, 29, 3] and *flow-based approaches* [14, 27, 31, 26]. Recommender-based approaches focus on measuring the recommendation credibility of available recommenders. On the other hand, flow-based approaches compute trust values by finding connected paths or connected components to the service provider being evaluated, based on the transitivity property [8] of trust. However, the aforementioned approaches either ignore distrust or treat it in a fixed way, while distrust is known at least as important as trust in Internetware and several other Web-based applications, including e-commerce [9], recommendation system [20], Web spamming protection [21], and edge sign prediction [16].

Additionally, existing approaches tend to compute a continuous trust value as the evaluation result, while one of the ultimate objectives of trustworthiness evaluation is to help decision making for end-users [24]. In our case, the decision is either to select the service or not to. This is analogous to the edge sign prediction problem which predicts a positive or negative attitude on a latent edge of a graph [16]. In fact, by regarding the trust and distrust statements over the system as a *trust graph*, trustworthiness evaluation is a generalized edge sign prediction problem: Instead of predicting a positive or negative sign, trustworthiness evaluation aims to compute a continuous trust value of the edge. To distinguish between two subproblems, we name the problem to compute a continuous trust value as *trust computation problem* and the problem to determine a trust sign as *trust decision problem*.

In this paper, we propose a *pattern-based approach PatTrust* to evaluate the personalized trustworthiness for a *trustee* from the point of view of a *trustor*. PatTrust takes the trust graph as input, and computes a trust value or decides a trust sign for the trustee-trustor pair. In our approach, trust or distrust statements that satisfy certain requirements can be assembled and these semblings are regarded as *trust patterns*. One of the major advantages of PatTrust is its flexibility: Various trust patterns that contain distrust can be characterized and learned under PatTrust. Based on these

trust patterns, PatTrust predicts the final trust value for a trustor-trustee pair by regression models, or predict the final trust decision by classification models. Note that these models can be easily substituted under PatTrust.

To further demonstrate the flexibility and effectiveness of PatTrust, we propose three trust models derived from social psychology theories. Based on these theories, we systematically study trust and distrust by characterizing several trust patterns for each theory. The studied theories include *balance theory* [4], *weak balance theory* [7], *status theory* [11], and their variations for continuous settings. PatTrust is flexible enough to accommodate these theories, and experimental results on a real data-set show that PatTrust can achieve low mean absolute error in trust computation and high predictive accuracy in trust decision. In addition, the experiments also show some evidence for studied theories, such as the inappropriateness of "the enemy of my enemy is my friend" in trust-based networks.

The rest of the paper is organized as follows. Section 2 covers related work of trustworthiness evaluation, and describes how distrust is treated in their work. Section 3 describes our PatTrust approach and studies several theories that contain trust and distrust. Section 4 presents the experimental results on the studied theories under PatTrust. Section 5 concludes the paper.

2. RELATED WORK

The focus of early *recommender-based approaches* for trustworthiness evaluation is on directly measuring the recommendation credibility of available recommenders. For example, PeerTrust system for peer-to-peer e-commerce environment proposes two measures for the recommendation credibility, namely, reputation-based measure and similarity-based measure [29]. In mobile ad-hoc networks, Buchegger and Boudec use a deviation test to eliminate unreliable recommendations from the final computation of trust [3]. In multi-agent systems, Sabater and Sierra determine the recommendation credibility by applying fuzzy logic [30] on the social relationships between the recommenders and the trustee [25]. Travos system in multi-agent system evaluates the recommendation credibility by tracking the past recommendation behavior of recommenders [23]. The trust value is then computed based on the recommendation content and the credibility of these recommendations.

Later, with the popularity and availability of underlying social networking structures, researchers begin to explore *flow-based approaches* that compute trust values by finding a set of connected paths or a connected component from the trustor to the trustee. Trust then flows from the trustor to the trustee over these paths or components. For example, in peer-to-peer networks, EigenTrust assumes that trust is recursively transitive through the network [14]. The EigenTrust algorithm computes global trust values by calculating the left principal eigenvector of the matrix representing the local trust values. Similar to EigenTrust, PowerTrust also uses a flow-based trust calculation approach, with special consideration of the power-law distribution of feedback [31]. In multi-agent system, Wang et al. define two operators to deal with trust propagation along a path, and trust aggregation among different paths [27]. To avoid the loss of information, FlowTrust uses the network flow theory on a connected component instead of finding connected paths [26].

Different from the previous two categories of approaches, we introduce a novel *pattern-based approach* PatTrust for trustworthiness evaluation. PatTrust is based on the characterization of trust patterns, and various applications can tailor different trust patterns and therefore produce different trust models. Similar to flow-based approaches, PatTrust also tries to find the connected paths from trustor to trustee. In contrast to flow-based approaches, PatTrust identifies whether the found paths satisfy certain characterized trust patterns. PatTrust counts these satisfied paths, and performs predictor training by considering these patterns as features. Finally, PatTrust predicts the trust value or trust decision of the trustor-trustee pair based on the trained predictor.

The aforementioned existing proposals put their focus only on trust information, while distrust is known at least as important as trust. Some other proposals realize the importance of distrust. However, they treat distrust in a fixed way. Among them, Guha et al. study one-step propagation and repeated propagation of distrust, and find that one-step propagation can achieve better results in predictive accuracy [9]. Ziegler and Lausen also incorporate distrust into their trust computation [32]. They argue that one-step propagation is questionable, and distrust cannot be propagated in any form. Nordheimer et al. use a threshold to distinguish trust from distrust, and accordingly generate one trust graph and one distrust graph for trustworthiness evaluation [22].

In contrast to these distrust proposals, PatTrust treats distrust in a more flexible manner by allowing several trust patterns with each of them containing different trust and distrust semblings. As a result, PatTrust can be used as a framework to study different trust patterns. Moreover, the importance of these trust patterns can be compared under PatTrust, and the comparison results provide evidence for the validity of distrust proposals. In this paper, we characterize and study several trust patterns from social psychology theories.

In machine learning and data mining domains, there are several pieces of work that focus on edge sign prediction. Leskovec et al. formulate the edge prediction problem and connect the problem to the theories of balance and status [16]. We generalize the problem to continuous settings, namely, PatTrust allows both binary and continuous input and can produce both binary and continuous output. Chua et al. propose a probabilistic model for online rating system to predict trust and distrust based on the ratings towards items [6]. In contrast, PatTrust only needs trust and distrust opinions between participants and can be applied to the systems where item ratings are unavailable.

In summary, one of the major advantages of PatTrust is its flexibility: 1) trust patterns can be characterized according to different requirements of different domains, 2) various distrust considerations can be learned and compared, 3) the framework can be applied to both binary and continuous settings, and 4) different predictors can be used to train the patterns. To the best of our knowledge, we are the

first to introduce the concept of trust pattern and develop a pattern-based approach for trustworthiness evaluation.

3. PATTRUST

In this section, we describe our PatTrust approach. The key idea of PatTrust is the concept of trust pattern. Based on the concept of trust pattern, PatTrust characterizes trust patterns and counts the number of these patterns between a trustor-trustee pair. After that, these patterns and corresponding numbers are used as features for prediction models which include regression models and classification models. Before characterizing trust patterns and describing trust models from studied theories, we first formulate the concept of trust pattern.

3.1 Trust Pattern Definition

The trust and distrust statements between participants in the network can be modeled as a weighted directed trust graph $G(V, E, T)$, where V is the set of participants, E is the set of trust and distrust edges between participants, and T is the corresponding trust opinion on every edge. This trust opinion could be binary or continuous. For each edge $e(u, v) \in E$ where $u, v \in V$, $T_{(u,v)}$ is the trust or distrust opinion on the edge. $T_{(u,v)} > 0$ if u trusts v, $T_{(u,v)} < 0$ if u distrusts v, and $T_{(u,v)} = 0$ if there is no trust or distrust statement from u to v. Based on the graph $G(V, E, T)$, we now define path on the graph:

Definition 1. Given $G(V, E, T)$, $P(u, v)$ is the *path* from u to v consisting of a set of edges $e_i(v_i, v_{i+1}) \in E$, for $i = 1, \cdots n$, where $v_1 = u$ and $v_{n+1} = v$.

To accommodate the theories of balance and status, we also define *unsigned path* as follows:

Definition 2. Given $G(V, E, T)$, the *unsignedpath* $P_u(u, v)$ from u to v consists of a set of node pairs (v_i, v_{i+1}), for $i = 1, \cdots n$, where $v_1 = u$, $v_{n+1} = v$, and either $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$.

In other words, unsigned path is the path from u to v in the undirected version of the directed input graph.

We now give the definition of trust pattern. We employ the syntax of the regular expression to define the concept of trust pattern as following, and different trust patterns can be characterized based on the concept.

Definition 3. A *trust pattern* $\mathcal{P}$ can be defined in the form of

$$\mathcal{P} ::= \varnothing | \varepsilon | a | \mathcal{P}/\mathcal{P} | \mathcal{P} \cdot \mathcal{P} | \mathcal{P}^*,$$

with $\varnothing$ denoting the empty set, ε denoting an empty word, and $a \in \Sigma$.

Σ contains the labels which indicate requirements on a single edge of the graph. For example, in binary settings, Σ could be $\{\overrightarrow{+}, \overleftarrow{+}, \overrightarrow{-}, \overleftarrow{-}, \}$ where $\overrightarrow{+}$ indicates a forward trust sign and $\overleftarrow{-}$ indicates a backward distrust sign. In continuous settings, Σ could be the form of $\{\overrightarrow{\succcurlyeq_{t_1}}, \overleftarrow{\preccurlyeq_{t_2}}\}$, where t_1 and t_2 are specified thresholds. $\overrightarrow{\succcurlyeq_{t_1}}$ means the forward trust value on the edge is no less than t_1, and $\overleftarrow{\preccurlyeq_{t_2}}$ means the backward trust value on the edge is no greater than t_2. Note that if $t_1 \leqslant t_2$, all the edges that satisfy $\succcurlyeq_{t_2}$ also satisfy $\succcurlyeq_{t_1}$, namely, $\succcurlyeq_{t_2} \subseteq \succcurlyeq_{t_1}$.

A characterized trust pattern contains the requirements on a sequence of edges, such as $\overrightarrow{+}\overrightarrow{-}$ and $\overrightarrow{\succcurlyeq_{t_1}}\overrightarrow{\preccurlyeq_{t_2}}$. To illustrate the expressive power of regular expression for the characterization of trust patterns, we present two examples as follows. The first is the one-step distrust propagation studied by Guha et al. [9]. It can be characterized as $\mathcal{P} = (\overrightarrow{+})^*\overrightarrow{-}$ with $\Sigma = \{\overrightarrow{+}, \overrightarrow{-}\}$. This trust pattern requires all the edges of a path to be positive except the last one, because only the last edge is considered as direct experience with the trustee. As another example, suppose a user is tired of moderate opinions and is only interested in extreme opinions, and suppose the scale of trust statement is $(-0.5, 0.5)$. The user may characterize a trust pattern $\mathcal{P} = (\succcurlyeq_{0.4} / \preccurlyeq_{-0.4})^*$ with $\Sigma = \{\succcurlyeq_{0.4}, \preccurlyeq_{-0.4}\}$.

With the the syntax definition of trust pattern, we now define the semantics of our trust pattern concept. First, let $\mathcal{L}(\mathcal{P})$ denote the language corresponding to $\mathcal{P}$, and we define $\mathcal{L}(\mathcal{P})$ as follows:

$$\begin{aligned}
&\mathcal{L}(\varnothing) = \varnothing \\
&\mathcal{L}(\varepsilon) = \{\varepsilon\} \\
&\mathcal{L}(a) = \{a\} \\
&\mathcal{L}(\mathcal{P}_1/\mathcal{P}_2) = \mathcal{L}(\mathcal{P}_1) \cup \mathcal{L}(\mathcal{P}_2) \\
&\mathcal{L}(\mathcal{P}_1 \cdot \mathcal{P}_2) = \mathcal{L}(\mathcal{P}_1)\mathcal{L}(\mathcal{P}_2) \\
&\mathcal{L}(\mathcal{P}^*) = \textstyle\bigcup_{i \in \mathbb{N}} \mathcal{L}(\mathcal{P})^i
\end{aligned}$$

Here, $\mathcal{P}^*$ denotes the iterator which is recursively interpreted as follows:

$$\begin{aligned}
&\mathcal{L}(\mathcal{P})^0 = \{\varepsilon\} \\
&\mathcal{L}(\mathcal{P})^{i+1} = \{uv | u \in \mathcal{L}(\mathcal{P}), v \in \mathcal{L}(\mathcal{P})^i\} \ (i \in \mathbb{N})
\end{aligned}$$

The semantics of trust pattern can be defined as:

Definition 4. A path $P(u, v)$ in $G(V, E, T)$ satisfies a trust pattern $\mathcal{P}$ can be defined as:

$$P(u, v) \models \mathcal{P} \overset{def}{=\!=} W(P(u, v)) \in \mathcal{L}(\mathcal{P})$$

Here, $W(P(u, v))$ indicates the sequence of labels on the edges of path $P(u, v)$, and each label indicates the trust value or sign of the edge satisfies the certain requirement. In other words, a path $P(u, v)$ satisfies the characterized trust pattern $\mathcal{P}$ when $W(P(u, v))$ is a word in the language $\mathcal{L}(\mathcal{P})$.

With this definition, we can determine whether a path satisfies a characterized trust pattern. For example, in binary settings, we have a trust pattern characterization that defines a language $\{\overrightarrow{+}\overrightarrow{-}, \overrightarrow{-}\overrightarrow{+}\}$. On the other side, There is a path $P(u, v)$ with $W(P(u, v)) = \overrightarrow{+}\overrightarrow{-}$ which means the path $P(u, v)$ consists of two ordered edges with trust and distrust signs on them. Then we say the path $P(u, v)$ satisfies the corresponding trust pattern.

Things are similar in continuous settings. For example, assume we have a characterized pattern with $\Sigma = \{\overrightarrow{\succcurlyeq_{0.3}}, \overrightarrow{\preccurlyeq_{-0.3}}\}$ and $\mathcal{L}(\mathcal{P}) = \{\overrightarrow{\succcurlyeq_{0.3}}\overrightarrow{\succcurlyeq_{0.3}}\overrightarrow{\preccurlyeq_{-0.3}}, \overrightarrow{\preccurlyeq_{-0.3}}\overrightarrow{\succcurlyeq_{0.3}}\overrightarrow{\succcurlyeq_{0.3}}\}$. A path $P(u, v)$

consists of three ordered edges with trust values 0.4, 0.4, and -0.4 on them. When determining whether the path satisfies the pattern, we need to label each edge of the path with the letters from Σ that the edge satisfies. The resulting $W(P(u,v))$ is $\overrightarrow{\succcurlyeq_{0.3}}\overrightarrow{\succcurlyeq_{0.3}}\overrightarrow{\preccurlyeq_{-0.3}}$ and $W(P(u,v)) \in \mathcal{L}(\mathcal{P})$ which means the path satisfies the corresponding pattern. Note that the semantics can also handle other kinds of paths, such as unsigned path $P_u(u,v)$.

3.2 Trust Pattern Characterization

To interpret the flexibility of PatTrust and systematically investigate trust and distrust, we characterize several classes of trust patterns from theories of social psychology, including balance theory, weak balance, status theory. The trust patterns we characterize for each theory is limited within two hops in this subsection. That is, we only consider the $u-w-v$ triads for a given node pair (u,v). We first introduce *16Triads* [17] as a baseline.

16Triads: There are 16 types of triads by adding two directions and two signs on an unsigned path $u-w-v$. These triads are represented as trust patterns in Table 1. As shown in the table, the directions of the edge are indicated by $\rightarrow$ and $\leftarrow$, and the signs are indicated by $+$ and $-$. In continuous trust patterns, the signs are substituted by $\succcurlyeq_\alpha$ and $\preccurlyeq_\beta$, with $\succcurlyeq_\alpha$ indicating the trust value of the edge is no less than α, and $\preccurlyeq_\beta$ the trust value is no greater than β.

We will study these patterns under PatTrust as a baseline to compare the results from following theories.

Balance: Balance theory has four principles, i.e., "the friend of my friend is my friend", "the enemy of my friend is my enemy", "the friend of my enemy is my enemy", and "the enemy of my enemy is my friend". These principals indicate that if the relationship between u and v is positive (trust), then the number of positive edges in (u,w) and (w,v) should be even where w forms a triad with u and v, regardless of the edge direction [4, 16]. This reduces the 16 triads to trust patterns $+-$, $-+$, $++$, and $--$, with the former two indicate a positive relationship between u and v and the latter two indicate a negative relationship.

We generalize balance theory to handle continuous input by introducing a threshold. Corresponding to the $+$ and $-$ signs in binary case, trust patterns in continuous case are in the form of $\succcurlyeq_\alpha$ and $\preccurlyeq_\beta$ where α and β could be defined by different applications. The four continuous trust patterns of balance are $\succcurlyeq_\alpha\preccurlyeq_\beta$, $\preccurlyeq_\beta\succcurlyeq_\alpha$, $\succcurlyeq_\alpha\succcurlyeq_\alpha$, and $\preccurlyeq_\beta\preccurlyeq_\beta$.

Weak Balance: A controversial principle in balance theory is "the enemy of my enemy is my friend". Weak balance deals with the controversy by removing this principle [7]. We will also study this alternative balance theory in PatTrust. As shown in Table 1, the characterized trust patterns are the same with those of balance theory, except for the last one.

Status: In the theory of status, a positive edge from u to v means that v has a higher status than u [11]. As a result, a positive edge from u to v is equal to a negative edge from v to u. Based on this principle, we can reverse all the negative edges to positive ones, and the 16 triads can be reduced to $\rightarrow\rightarrow$, $\rightarrow\leftarrow$, $\leftarrow\rightarrow$, and $\leftarrow\leftarrow$. The trust pattern $\rightarrow\rightarrow$ indicates a positive relationship between u and v, and $\leftarrow\leftarrow$ indicates a negative relationship.

Similar to balance theory, continuous versions the trust patterns can be characterized as $\overrightarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}$, $\overrightarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}$, $\overleftarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}$, and $\overleftarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}$. All the characterized trust patterns are shown in Table 1, and the theoretical output is shown in the third column of Table 4.

3.3 Trust Pattern Extension

We have characterized several trust patterns with path length restricted to two in the previous subsection. Actually, we can define three trust models based on these trust patterns under PatTrust. Before describing how these trust models operate, we first extend the studied theories to handle arbitrary path length by introducing several operators in this subsection.

Usually, only the last edge of a path is considered as direct experience with the trustee [9, 10]. Consequently, our operators only function on the rest of the edges of the path. Finally, paths with arbitrary length are reduced to length-two trust patterns in Table 1.

Balance: For balance theory, the four principles can be combined arbitrarily and these combinations result in several trust patterns. We apply an operator Ⓑ to the edges of the path except for the last edge to reduce these patterns. Given two consecutive edges (u,w) and (w,v) without considering their directions, we can define the operator Ⓑ as follows:

Definition 5. Let $+$ indicate trust edge and $-$ indicate distrust edge, and let S_1 and S_2 be the trust signs of two consecutive edges (u,w) and (w,v) on an unsigned path,

$$S = S_1 Ⓑ S_2 = \begin{cases} + & \text{if } S_1 = S_2 \\ - & \text{otherwise} \end{cases}$$

Based on the definition, we have the following theorem which enables us to merge the trust signs in both bottom up and up bottom way.

Theorem 1. The Ⓑ operator is associative.

The proof is straightforward as the sign of the path only depends on the parity of the distrust edge number.

The Ⓑ operator reduces the paths of arbitrary length to trust patterns of length two. Using the one-step distrust propagation [9] as an example again, whose trust pattern is $(\overrightarrow{+})^*\overrightarrow{-}$. Ⓑ would reduce this trust pattern to $\overrightarrow{+}\overrightarrow{-}$. The continuous case is similar as $\succcurlyeq_\alpha$ can be mapped to $+$ and $\preccurlyeq_\beta$ can be mapped to $-$.

Weak Balance: As to weak balance, those paths contain consecutive distrust edges are not considered, and we can define an operator Ⓦ which has the associative property as well.

Definition 6. Let $+$ indicate trust edge and $-$ indicate distrust edge, and let S_1 and S_2 be the trust signs of two

Table 1: Trust patterns of 16Triads, Balance, Weak Balance, and Status.

Theory	Binary trust patterns	Continuous trust patterns
16Triads	$\overset{\rightarrow}{+}\overset{\rightarrow}{+}, \overset{\rightarrow}{+}\overset{\rightarrow}{-}, \overset{\rightarrow}{-}\overset{\rightarrow}{+}, \overset{\rightarrow}{-}\overset{\rightarrow}{-},$ $\overset{\rightarrow}{+}\overset{\leftarrow}{+}, \overset{\rightarrow}{+}\overset{\leftarrow}{-}, \overset{\rightarrow}{-}\overset{\leftarrow}{+}, \overset{\rightarrow}{-}\overset{\leftarrow}{-},$ $\overset{\leftarrow}{+}\overset{\rightarrow}{+}, \overset{\leftarrow}{+}\overset{\rightarrow}{-}, \overset{\leftarrow}{-}\overset{\rightarrow}{+}, \overset{\leftarrow}{-}\overset{\rightarrow}{-},$ $\overset{\leftarrow}{+}\overset{\leftarrow}{+}, \overset{\leftarrow}{+}\overset{\leftarrow}{-}, \overset{\leftarrow}{-}\overset{\leftarrow}{+}, \overset{\leftarrow}{-}\overset{\leftarrow}{-}$	$\overrightarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}, \overrightarrow{\succcurlyeq_\alpha}\overrightarrow{\preccurlyeq_\beta}, \overrightarrow{\preccurlyeq_\beta}\overrightarrow{\succcurlyeq_\alpha}, \overrightarrow{\preccurlyeq_\beta}\overrightarrow{\preccurlyeq_\beta},$ $\overrightarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}, \overrightarrow{\succcurlyeq_\alpha}\overleftarrow{\preccurlyeq_\beta}, \overrightarrow{\preccurlyeq_\beta}\overleftarrow{\succcurlyeq_\alpha}, \overrightarrow{\preccurlyeq_\beta}\overleftarrow{\preccurlyeq_\beta},$ $\overleftarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}, \overleftarrow{\succcurlyeq_\alpha}\overrightarrow{\preccurlyeq_\beta}, \overleftarrow{\preccurlyeq_\beta}\overrightarrow{\succcurlyeq_\alpha}, \overleftarrow{\preccurlyeq_\beta}\overrightarrow{\preccurlyeq_\beta},$ $\overleftarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}, \overleftarrow{\succcurlyeq_\alpha}\overleftarrow{\preccurlyeq_\beta}, \overleftarrow{\preccurlyeq_\beta}\overleftarrow{\succcurlyeq_\alpha}, \overleftarrow{\preccurlyeq_\beta}\overleftarrow{\preccurlyeq_\beta}$
Balance	$++, +-, -+, --$	$\succcurlyeq_\alpha\succcurlyeq_\alpha, \succcurlyeq_\alpha\preccurlyeq_\beta, \preccurlyeq_\beta\succcurlyeq_\alpha, \preccurlyeq_\beta\preccurlyeq_\beta$
Weak Balance	$++, +-, -+$	$\succcurlyeq_\alpha\succcurlyeq_\alpha, \succcurlyeq_\alpha\preccurlyeq_\beta, \preccurlyeq_\beta\succcurlyeq_\alpha$
Status	$\rightarrow\rightarrow, \rightarrow\leftarrow, \leftarrow\rightarrow, \leftarrow\leftarrow$	$\overrightarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}, \overrightarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}, \overleftarrow{\succcurlyeq_\alpha}\overrightarrow{\succcurlyeq_\alpha}, \overleftarrow{\succcurlyeq_\alpha}\overleftarrow{\succcurlyeq_\alpha}$

consecutive edges (u,w) and (w,v) on an unsigned path,

$$S = S_1 ⓦ S_2 = \begin{cases} + & \text{if } S_1 = S_2 = + \\ - & \text{if } S_1 \neq S_2 \end{cases}$$

Theorem 2. The ⓦ operator is associative.

Note that ⓦ only works on paths without consecutive distrust edges, and weak balance will discard the paths whose reduced trust pattern is $--$.

Status: As mentioned above, in the theory of status, a positive edge from u to v means that v has a higher status than u. This principle is transitive: if v has a higher status than u, and u has a higher status than w, then v has a higher status than w. Based on this transitive principle, the trust patterns of this theory can be characterized as $(\rightarrow)^*$ and $(\leftarrow)^*$.

We first transform all negative edges to positive ones, and following balance, we also define a ⓢ operator to reduce the trust patterns to length two.

Definition 7. Let $\rightarrow$ indicate forward trust edge and $\leftarrow$ indicate backward trust edge, and let S_1 and S_2 be the trust directions of (u,w) and (w,v),

$$S = S_1 ⓢ S_2 = \begin{cases} \rightarrow & \text{if } S_1 = S_2 = \rightarrow \\ \leftarrow & \text{if } S_1 = S_2 = \leftarrow \end{cases}$$

Theorem 3. The ⓢ operator is associative.

The proof is straightforward with the transitivity property of the status principle. Note that ⓢ only considers paths with one direction except for the last edge. For example, a trust pattern $(\rightarrow)^*\leftarrow$ will be reduced to $\rightarrow\leftarrow$ by ⓢ, and $\rightarrow\leftarrow\rightarrow$ will be discarded by ⓢ.

3.4 Trust Model

As indicated in previous subsections, the three trust models are based on balance, weak balance, and status, respectively. Based on the trust patterns, these models first search available paths and then determine whether the paths satisfy certain patterns. Our operators will be used if the path length is greater than two. Trust models under PatTrust can be divided into two phases of model training and trustworthiness evaluation.

Given a trustor and trustee, by considering the network of trust and distrust statements between participants in the system as the weighted graph $G(V,E,T)$, the task of evaluating the trustworthiness of trustee is transformed to predicting the trust value or trust sign of the corresponding latent edge in the graph. Overall, given characterized trust patterns $\mathcal{P}_i, i = 0,1,...,n$, Trust models under PatTrust evaluate the trustworthiness of trustee t for trustor s as follows:

1. Training:
   (a) For each edge in the graph, say $e(u,v)$, PatTrust finds all the paths or unsigned paths between u and v that satisfy $\mathcal{P}_i$, based on the semantics of trust pattern. The number of the found paths are counted as c_i.
   (b) For each $\mathcal{P}_i$ and c_i on edge $e(u,v)$, PatTrust then records them as features, and at the same time records the trust sign or trust value on the edge $e(u,v)$.
   (c) PatTrust would repeat step 1.(a) and 1.(b) for all edges in the graph, and train the prediction model based on the recorded information. Particularly, regression models for continuous output and classification models for binary output.
2. Evaluation: PatTrust predicts the trust value or trust sign for s and t based on the trained model, by counting the paths for s and t as step 1.(a).

For efficiency, PatTrust can complete the training part in advance, and update the trained model periodically. In addition, there is evidence showing that the model trained on one data-set can generalize across other data-sets for edge sign prediction [16]. The generalization ability of PatTrust is left as future work.

4. EXPERIMENTS

In this section, we evaluate PatTrust on a real data-set from *Advogato* [1]. We follow the experimental framework of Guha et al. [9] and Leskovec et al. [16]. That is, the edge of which the trust value being computed or the trust sign being decided is suppressed, and we are aiming to evaluate the trustworthiness of the this edge.

4.1 Data-Set Description

Epinions and Slashdot are usually used for sign prediction [15, 16]. However, these data-sets only have binary attitude of

Table 3: Experimental results of various schemes under PatTrust.

Theory	Input	Output	Em = 0	Em = 10	Em = 25
16Triads	Binary	Trust computation (MAE)	0.1642	0.1552	0.1515
		Trust decision (Accuracy)	0.832	0.8904	0.8729
	Continuous	Trust computation (MAE)	0.1559	0.1382	0.1185
		Trust decision (Accuracy)	0.7756	0.9251	0.9011
Balance	Binary	Trust computation (MAE)	0.1745	0.1745	0.1729
		Trust decision (Accuracy)	0.7986	0.8601	0.8799
	Continuous	Trust computation (MAE)	0.1707	0.1607	0.1205
		Trust decision (Accuracy)	0.7707	0.9263	0.9109
Weak Balance	Binary	Trust computation (MAE)	0.1759	0.1748	0.1726
		Trust decision (Accuracy)	0.7862	0.8691	0.9039
	Continuous	Trust computation (MAE)	0.1709	0.1548	0.1112
		Trust decision (Accuracy)	0.7684	0.9379	0.9414
Status	Binary	Trust computation (MAE)	0.1721	0.1709	0.167
		Trust decision (Accuracy)	0.792	0.854	0.8751
	Continuous	Trust computation (MAE)	0.1641	0.1672	0.1602
		Trust decision (Accuracy)	0.7605	0.8947	0.8914

Table 2: High level statistics of Advogato, and trust and distrust network of Advogato.

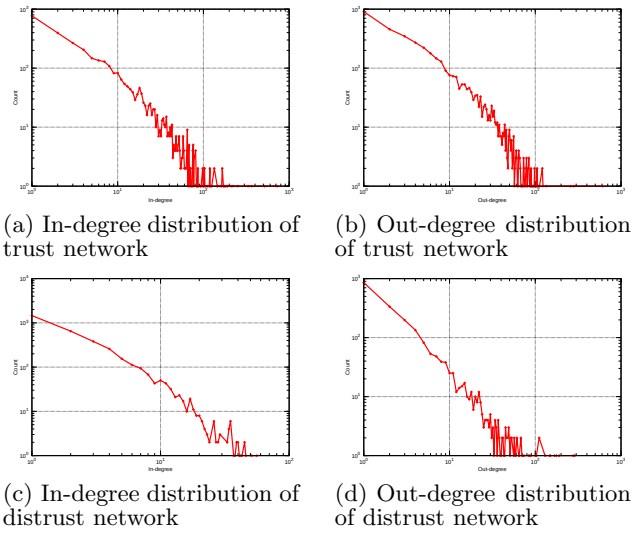
Statistics	Advogato	Trust	Distrust
Nodes	5,428	4,459	3978
Edges	51,493	38,630	12863
Avg. degree	18.97	17.33	6.47
Edge proportion	100%	75%	25%

trust or distrust, while PatTrust is also developed for continuous trust and distrust. In our experiments, we choose the Web of Trust data-sets Advogato.

Our Advogato data-set is a snapshot on June 23, 2011 which contains 5,428 nodes and 51,493 edges. There are 4 levels of trust statements in the data-set, i.e., "Observer", "Apprentice", "Journeyer", and "Master". These statements can be mapped into real values in $(-0.5, 0.5)$ where negative trust value means distrust and positive value means trust. In our experiments, we map them to -0.4, -0.1, 0.1, and 0.4, respectively. To verify our mapping, we study both the resulting trust network with positive values and the distrust network with negative ratings as shown in Table 2. The proportion of distrust edges is close to that of the Epinions and Slashdot data-sets [16]. Additionally, we find power-law distribution [2] in both trust and distrust networks as shown in Fig. 1. This distribution is also found in Epinions' trust and distrust network [20].

4.2 Experimental Setup and Results

We now describe our experimental setup and results. In our experiments, we first restrict the usage of local trust information within two hops ($hop = 2$), namely, we only consider the paths of length two. Based on the results, we compare the importance of different trust patterns, and discuss observations for trust computation and trust decision. We then remove the restriction of path length, and conduct experiments on the proposed extensions and operators. To avoid the excessive number of paths, we only search paths within three hops ($hop = 3$) and four hops ($hop = 4$). For



(a) In-degree distribution of trust network

(b) Out-degree distribution of trust network

(c) In-degree distribution of distrust network

(d) Out-degree distribution of distrust network

Figure 1: Power-law distributions on trust and distrust network of Advogato.

all experiments, the mean absolute error and the predictive accuracy are reported for trust computation and trust decision, respectively.

We take into account of the following four parameters in our experiments:

1. For trust patterns, we characterize three classes of trust patterns from social psychology theories, as interpreted in section 3.2. When $hop = 2$, we additionally consider 16Triads.
2. For input, we consider binary and continuous cases as illustrated in Table 1. Specially, we set $\alpha = 0.3$, $\beta = -0.3$ in the following experiments.
3. For output, we check PatTrust on both trust computation and trust decision problems, and report the mean

Table 4: The linear regression coefficients of Balance, Weak Balance, and Status with continuous input and $Em = 10$.

Theory	Trust pattern	Theory output	Coefficient
Balance	$\succcurlyeq_{0.3}\succcurlyeq_{0.3}$	1	0.0042
	$\succcurlyeq_{0.3}\preccurlyeq_{-0.3}$	-1	-0.0108
	$\preccurlyeq_{-0.3}\succcurlyeq_{0.3}$	-1	-0.0287
	$\preccurlyeq_{-0.3}\preccurlyeq_{-0.3}$	1	-0.0281
Weak Balance	$\succcurlyeq_{0.3}\succcurlyeq_{0.3}$	1	0.0023
	$\succcurlyeq_{0.3}\preccurlyeq_{-0.3}$	-1	-0.0154
	$\preccurlyeq_{-0.3}\succcurlyeq_{0.3}$	-1	-0.0329
Status	$\rightarrow\rightarrow$ $\succcurlyeq_\alpha\succcurlyeq_\alpha$	1	0.0163
	$\rightarrow\leftarrow$ $\succcurlyeq_\alpha\succcurlyeq_\alpha$	0	0.0011
	$\leftarrow\rightarrow$ $\succcurlyeq_\alpha\succcurlyeq_\alpha$	0	0.0047
	$\leftarrow\leftarrow$ $\succcurlyeq_\alpha\succcurlyeq_\alpha$	-1	-0.0235

absolute error for the former one and the predictive accuracy for the latter one. In our experiments, we choose linear regression model to compute a continuous trust value, and LIBSVM [5] model to predict the binary trust sign.

4. When $hop = 2$, we consider an additional parameter *embeddedness*. Embeddedness is defined as the size of common neighbors between two nodes in an undirected manner [16]. Specially, we study three subsets of edges with minimum embeddedness of 0 ($Em = 0$), 10 ($Em = 10$), and 25 ($Em = 25$).

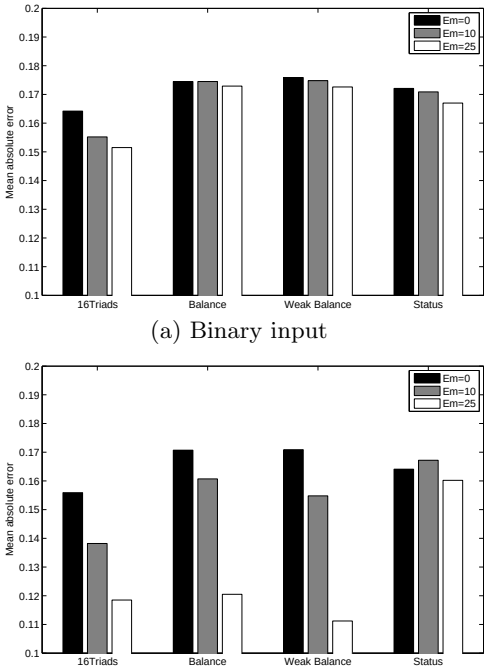
These four dimensions produce $4 * 2 * 2 * 3 = 48$ experimental schemes when $hop = 2$, and the results are shown in Table 3. The best performance of both trust computation and trust decision is achieved by the weak balance theory with continuous input. This result provides support for the weak balance theory and shows the advantage of continuous input. When $hop = 3$ and $hop = 4$, there are $3 * 2 * 2 = 12$ experimental schemes, respectively. The results are shown in Fig. 4.

4.2.1 Pattern Importance Comparison

Before further discussing the results of Table 3, we first compare the importance of trust patterns in different theories, especially those patterns containing distrust. For simplicity, we only report the coefficients of linear regression models for Balance, Weak Balance, and Status with continuous input and $Em = 10$, as shown in Table 4.

There are several observations from the table. First, regression coefficients of all the patterns are consistent with the theory output, except for the $\preccurlyeq_{-0.3}\preccurlyeq_{-0.3}$ pattern. This result provides further evidence for the inappropriateness of "the enemy of my enemy is my friend" principle in trust-based networks. However, this principle may play role in some other networks such as the political networks [12].

Second, for patterns that contain distrust, such as $\succcurlyeq_{0.3}\preccurlyeq_{-0.3}$ and $\preccurlyeq_{-0.3}\succcurlyeq_{0.3}$, the absolute values of the coefficients are significantly greater than those patterns only consider trust, such as $\succcurlyeq_{0.3}\succcurlyeq_{0.3}$. This observation again reveals the importance of distrust information for trustworthiness evaluation.



(a) Binary input

(b) Continuous input

Figure 2: Mean absolute error of trust computation on Advogato.

Third, the absolute values of the coefficients for $\succcurlyeq_{0.3}\preccurlyeq_{-0.3}$ and $\preccurlyeq_{-0.3}\succcurlyeq_{0.3}$ in weak balance theory are greater than those corresponding coefficients for balance theory. Considering the fact that weak balance performs better than balance, we can conjecture that these patterns which contain distrust play an important role for improving the performance of overall prediction.

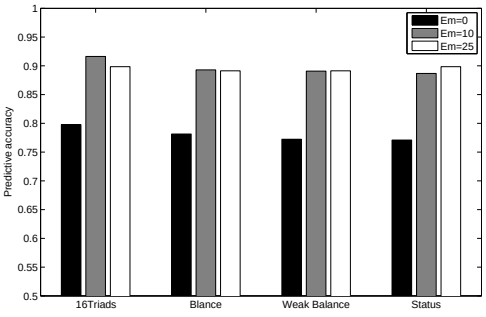
Finally, the results of status support the validity of the theory in trust-based networks. Further combination of status and weak balance could be developed to support trustworthiness evaluation with distrust information.

4.2.2 Trust Computation

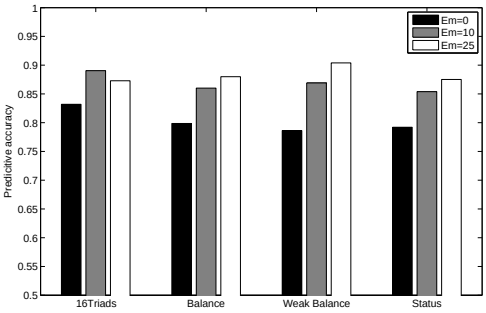
We now discuss the results for trust computation as shown in Fig. 2. The mean absolute errors for the three dimensions of theory, input, and Em are plotted.

First, the mean absolute errors of 16Triads are smaller than the other theories in general. This means both trust directions and trust signs are important for trust computation, because the balance theory and the weak balance theory ignore trust directions to a certain extent, and the status theory only considers trust directions by treating negative edges the same way as the opposite positive edges.

Second, PatTrust performs better with continuous input. This suggests that trust evaluation based on continuous or multilevel trust opinions can be more accurate than only based on positive and negative attitude. Many web sites are introducing multilevel trust ratings, and we believe that the multilevel or continuous trust can help to improve the performance of trustworthiness evaluation.



(a) Binary input



(b) Continuous input

Figure 3: Predictive accuracy of trust decision on Advogato.

Third, PatTrust becomes more effective with the increase of embeddedness, especially with continuous input. This is consistent with our intuition: more common neighbors can help to improve trustworthiness evaluation by providing more trust and distrust information. It is noteworthy that with only neighborhood information, the mean absolute error can be reduced to around 0.1.

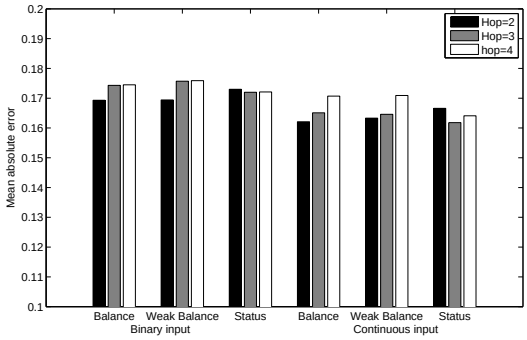
4.2.3 Trust Decision

Fig. 3 shows the results for predicting edge signs, and the results are similar to trust computation.

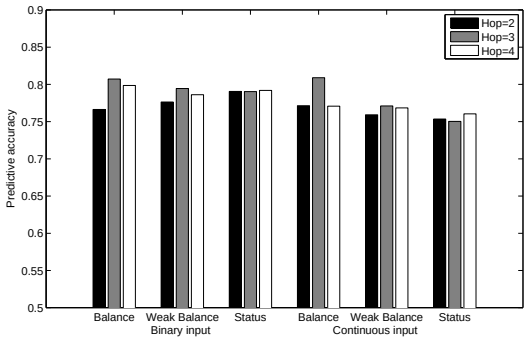
First, as shown in the figure, the accuracy results obtained from these four theories are close to each other, with lowest error rate 5.86%. We do not observe the superiority of 16Triads indicating that trust directions and trust signs in trust decision are not as important as they are in trust computation.

Second, different from trust computation, the accuracy results of $Em = 0$ with continuous input are lower than that with binary input. This is probably because of the lack of paths that satisfy the continuous trust patterns. However, when $Em = 10$ and $Em = 25$, the results of continuous input become better.

Third, similar to trust computation, we also observe that with greater embeddedness, the predictive accuracy increases in most of the cases. This increase is especially dramatic from $Em = 0$ to $Em = 10$ with continuous input. However, from $Em = 10$ to $Em = 25$, the predictive accuracy only increases in weak balance and declines in other theories. This is due to sparsity of the training set with large embeddedness.



(a) Trust computation



(b) Trust decision

Figure 4: The results of operators on Advogato with $hop = 2$, $hop = 3$, and $hop = 4$.

4.2.4 Operators

We now discuss the results when applying our operators on paths with length greater than two. The results for trust computation and trust decision with $hop = 3$ and $hop = 4$ are shown in Fig. 4. Note that $hop = 4$ means we only search paths within 4 hops. The results with $hop = 2$ are also plotted for comparison. Note that the left half of the figures are the results of binary input and the right half of continuous input.

First, we observe that the results of trust computation with $hop = 3$ and $hop = 4$ are not as good as the case when $hop = 2$. This is consistent with the intuition that trust information from neighborhood is more reliable than that from several hops away. Second, the trust decision accuracy increases when hop changes from 2 to 3. However, overall, we observe that the results of $hop = 3$ and $hop = 4$ are close to the results when $hop = 2$ in both trust computation and trust decision.

5. CONCLUSIONS

In this paper, we have proposed a novel pattern-based approach PatTrust for trustworthiness evaluation in Internetware and other Web-based systems. PatTrust can handle trust and distrust by characterizing different trust patterns. One of the major advantages of PatTrust is its flexibility in trust pattern characterizations, distrust management, and predictive power. It can also deal with binary and continuous input for both trust computation and trust decision.

Extensive experiments of trust patterns and trust models derived from social psychology show the flexibility and effectiveness of PatTrust. Several comparisons also show some interesting evidence, such as the inappropriateness of "the enemy of my enemy is my friend" principle, and the advantage of continuous input or multilevel input in trust-based networks.

Future directions include more extensive experiments on the theories, for example, to study how the hop constraint affects the effectiveness of PatTrust. We are also interested in more elaborate design of pattern-based approach for trustworthiness evaluation, for example, to generalize PatTrust to accommodate more complicated trust evaluation approaches from literature. We also plan to test the generalization ability of PatTrust across more real data-sets, e.g., test one data-set with the prediction model trained from another data-set.

6. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 60736015, 61073030), and the National 973 Program of China (2009CB320702).

7. REFERENCES

[1] Advogato dataset, available at *http : //www.trustlet.org/wiki/Advogato_dataset*.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for mobile ad-hoc networks. Technical report, KTH Royal Institute of Technology, Theoretical Computer Science Group, 2004.

[4] D. Cartwright and F. Harary. Structural balance: a generalization of heider's theory. *Psychological Review*, 63(5):277–293, 1956.

[5] C. Chang and C. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

[6] F. C. T. Chua and E.-P. Lim. Trust network inference for online rating data using generative models. In *Proc. of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'10)*, pages 889–898. ACM, 2010.

[7] J. Davis. Structural balance, mechanical solidarity, and interpersonal relations. *American Journal of Sociology*, pages 444–462, 1963.

[8] J. A. Golbeck. *Computing and applying trust in web-based social networks*. 2005. Ph.D. Dissertation.

[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of the 13th International Conference on World Wide Web (WWW'04)*, pages 403–412. ACM, 2004.

[10] C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proc. of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)*, pages 1025–1032, May 2009.

[11] F. Heider. Attitudes and cognitive organization. *Journal of psychology*, 21(1):107–112, 1946.

[12] T. Hogg, D. Wilkinson, G. Szabo, and M. Brzozowski. Multiple relationship types in online communities and social networks. In *Proc. of the 23rd National Conference on Artificial Intelligence (AAAI'08)*, pages 30–35, 2008.

[13] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th International Conference on World Wide Web (WWW'03)*, pages 640–651. ACM, 2003.

[15] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proc. of the 18th international conference on World Wide Web (WWW'09)*, pages 741–750. ACM, 2009.

[16] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proc. of the 19th international conference on World Wide Web (WWW'10)*, pages 641–650. ACM, 2010.

[17] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proc. of the 28th international conference on Human factors in computing systems (CHI'10)*, pages 1361–1370. ACM, 2010.

[18] J. Lü, X. Ma, X. Tao, F. Xu, and H. Hu. Research and progress on Internetware. *Science in China (Series E)*, 36(10):1037–1080, 2006.

[19] J. Lü, X. Tao, X. Ma, H. Hu, F. Xu, and C. Cao. Research on agent-based Internetware. *Science in China (Series E)*, 35(12):1–21, 2005.

[20] H. Ma, M. Lyu, and I. King. Learning to recommend with trust and distrust relationships. In *Proc. of the 3rd ACM conference on Recommender Systems*, pages 189–196. ACM, 2009.

[21] P. Metaxas. Using propagation of distrust to find untrustworthy web neighborhoods. In *Proc. of the 4th International Conference on Internet and Web Applications and Services*, pages 516–521. IEEE, 2009.

[22] K. Nordheimer, T. Schulze, and D. Veit. Trustworthiness in networks: A simulation approach for approximating local trust and distrust values. In *Proc. of IFIP International Conference on Trust Management*, pages 157–171. Springer-Verlag, 2010.

[23] J. Patel, W. L. Teacy, N. R. Jennings, and M. Luck. A probabilistic trust model for handling inaccurate reputation sources. In *Proc. of the 3rd International Conference of Trust Management (iTrust'05)*, pages 193–209. Springer-Verlag, 2005.

[24] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[25] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 475–482. ACM, 2002.

[26] G. Wang and J. Wu. Flowtrust: trust inference with network flows. *Frontiers of Computer Science in China*, 5(2):181–194, 2011.

[27] Y. Wang and M. P. Singh. Trust representation and aggregation in a distributed agent system. In *Proc. of*

the 21st National Conference on Artificial Intelligence (AAAI'06), volume 21, pages 1425–1430, 2006.

[28] Y. Wang and J. Vassileva. A review on trust and reputation for web service selection. In *27th International Conference on Distributed Computing Systems Workshops*, page 25. IEEE, 2007.

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

[30] L. A. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30(3):407–428, 1975.

[31] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):460–473, 2007.

[32] C. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4):337–358, 2005.