TRUST ASSESSMENT IN ONLINE SOCIAL NETWORKS[*]

FULL VERSION

Guangchi Liu
Research & Development Department
Stratifyd, Inc.
Charlotte, NC, 28202
luke.liu@stratifyd.com

Qing Yang
Department of Computer Science and Engineering
University of North Texas
Denton, TX 76207
qing.yang@unt.edu

Honggang Wang
Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth
North Dartmouth, MA, 02747
hwang1@umassd.edu

Alex X. Liu
Department of Computer Science and Engineering
Michigan State university
East Lansing, MI, USA
alexliu@cse.msu.edu

ABSTRACT

Assessing trust in online social networks (OSNs) is critical for many applications such as online marketing and network security. It is a challenging problem, however, due to the difficulties of handling complex social network topologies and conducting accurate assessment in these topologies. To address these challenges, we model trust by proposing the three-valued subjective logic (3VSL) model. 3VSL properly models the uncertainties that exist in trust, thus is able to compute trust in arbitrary graphs. We theoretically prove the capability of 3VSL based on the Dirichlet-Categorical (DC) distribution and its correctness in arbitrary OSN topologies. Based on the 3VSL model, we further design the AssessTrust (AT) algorithm to accurately compute the trust between any two users connected in an OSN. We validate 3VSL against two real-world OSN datasets: Advogato and Pretty Good Privacy (PGP). Experimental results indicate that 3VSL can accurately model the trust between any pair of indirectly connected users in the Advogato and PGP.

***K*eywords** Trust Assessment · Online Social networks · Three-valued Subjective Logic · Trust Model

1 Introduction

Online social networks (OSNs) are among the most frequently visited places on the Internet. OSNs help people not only to strengthen their social connections with known friends but also to expand their social circles to friends of friends who they may not know previously. Trust is the enabling factor behind user interactions in OSNs and is crucial to almost all OSN applications. For example, in recommendation and crowdsourcing systems, trust helps to identify trustworthy opinions and/or users [1, 2]. In online marketing applications[3], trust is used to identify trustworthy sellers. In a proactive friendship construction system [4], trust enables the discovery of potential friendships. In wireless network domain, trust can help a cellular device to discover trustworthy peers to relay its data [5]. In security domain, trust is considered an important metric to detect malicious users [6, 7, 8, 9]. Given the above-mentioned applications, one confounding issue is to what degree a user can trust another user in an OSN. This paper concerns the fundamental issue of trust assessment in OSNs: *given an OSN, how to model and compute trust among users*?

Trust is traditionally considered as reputation or the probability of a user being benign. In online marketing, users rate each other based on their interactions, so the trust of a user can be derived from aggregated ratings. In the network security domain, however, the trust of a given user is defined as the probability that this user will behave normally in the future. Based on results from previous studies [10, 11, 12, 13], we define trust as *the probability that a trustee will*

[*] This paper is the full version of "Trust Assessment in Online Social Network" published in *IEEE Transaction on Dependable and Secure Computing*.

behave as expected, from the perspective of a trustor. Here, both trustor and trustee are regular users in an OSN where the trustor is interested in knowing how trustworthy the trustee is. This general definition of trust makes it applicable for a wide range of applications. We also assume that trust in OSNs is determined by objective evidence, i.e., cognition based trust [14, 15, 16, 17], is not considered in this paper.

1.1 Problem Statements

We model a social network as a directed graph $G = (V, E)$ where a vertex $u \in V$ represents a user, and an edge $e(u, v) \in E$ denotes a trust relation from u to v. The weight of $e(u, v)$ denotes how much u trusts v, which is commonly referred to as *direct trust*. A trustor may leverage the recommendations from other users to derive a trustee's trust, which is called *indirect trust*. We are interested in computing the indirect trust between two users who have not established a direct trust previously. To solve this problem, we first need to design a trust model that works with both direct and indirect trust. Based on the assumption that trust is determined by objective evidence, designing a trust model can be stated as follows.

- **P1**: *Given the interactions between a trustor and a trustee, how to model the trust of the trustee, from the trustor's perspective?*

The second problem is to compute/infer indirect trust between users in an OSN. Solving this problem means the trust between two users, without previous interactions, can be computed. Because the indirect trust inference is available, a trustor can conduct a trust assessment of a trustee in an OSN. As such, the second problem is formulated as follows.

- **P2**: *Given a social network $G = (V, E)$, $\forall$ u and v, s.t. $e(u, v) \notin E$ and $\exists$ at least one path from u to v, how does one compute u's trust in v,* i.e.*, how should u trust a stranger v?*

1.2 Limitation of Prior Art

Existing trust models can be categorized as topology (or graph) based models [18, 19, 9, 8, 20, 21], PageRank based models [22, 23, 24], probability based models [25, 26, 27], and subjective logic based models [28]. None of them, however, are able to accurately model and compute trust in OSNs.

Topology based models [18, 19, 9, 8] treat trust assessment as a community detection problem and employ a random-walk method to identify users within the same community. These users are considered trustworthy to each other. The key limitation of these models is that the trust values of users within a community are indistinguishable [29], restricting their applications to only coarse trust assessments. Graph based models [20, 21, 30, 31] assign a different real number, ranging from 0 to 1, to every edge in a social network, and employ various graph searching algorithms to evaluate the trust of users. The major limitation of these models is that trust is represented as a single real number, ignoring the uncertainty in trust.

Unlike graph based models, PageRank based models, e.g., TrustRank and EigenTrust [22, 32, 23], apply the idea of PageRank to rank users based on their trust values. A user's trust is obtained by calculating how likely it will be reached by a trustor in the network. In these models, the probability of a user being reached is determined by the connections between itself and the trustor. The key issue of these models is that they mistakenly treat the trust propagation process as a random walk process.

Probability based models [25, 26, 27] consider trust to be a probability distribution, i.e., a trustor uses its historical interactions with a trustee to construct a probabilistic model, to predict the trustee's future behavior. The major limitation of these models is that they only focus on modeling direct trust and do not explicitly consider the indirect trust assessment problem. Although the subjective logic based models [28, 33] make an attempt to jointly consider both direct trust and indirect trust inference, it can only handle series-parallel network topologies. Their performance degrades drastically in complex social networks.

1.3 Proposed Approach

To address problem **P1**, we propose the three-valued subjective logic (3VSL) model that accurately models the trust between a trustor and a trustee, based on their interactions. 3VSL is inspired by the subjective logic (SL) model [28], however, it is significantly different from SL.

The major difference between SL and 3VSL lies in the definitions of uncertainty in trust. SL believes the uncertainty in the trust of a trustee never changes, however, 3VSL considers the uncertainty increases as trust propagates among users in an OSN. Therefore, an extra state, called uncertainty state, is introduced in 3VSL to cope with the changing of uncertainty in trust.

The trust of a trustee, i.e., the probability that it will behave as expected, can be represented by a *Dirichlet-Categorical* (DC) distribution that is characterized by three parameters α, β and γ. Here, α is the number of positive interactions occurred, i.e., a trustor observed that the trustor behaved as expected for α times. β denotes the amount of negative interactions, indicating the trustee did not behave as expected. It is also quite possible that the behavior of the trust is ambiguous, i.e., it is impossible to determine whether it behaved as expected or not. In this case, we consider uncertain observations are made and use γ to record them. Uncertainty is generated not only when ambiguous behaviors are observed but also when trust propagates within an OSN, which will be elaborated in details in Section 3. The observations kept in α, β and γ are also called *evidence*, as they are used to judge whether the trustee is trustworthy or not. The major reason of introducing the uncertain state in 3VSL is to accurately capture the trust propagation process. When trust propagates from a user to another, certain evidence in α and β are "distorted" and "converted" into uncertain evidence. Given a DC distribution, it can be represented by a vector $\langle \alpha, \beta, \gamma \rangle$, which is also called *opinion*. On the other hand, the trustee's trust can be derived from a DC distribution; therefore, trust can be represented by an opinion. In the rest of this paper, we treat trust and opinion as interchangeable concepts, unless otherwise specified.

To address problem **P2**, we propose a trust assessment algorithm, called AssessTrust (AT), based on the 3VSL model. The AT algorithm decomposes the network between the trustor and trustee as a parsing tree that provides the correct order of applying trust operations to computer the indirect trust between the two users. Here, the trust operations available in trust computation are the discounting operation and combining operation. Leveraging these two operations, AT is proven to be able to accurately compute the trust between any two users connected in an OSN. Because 3VSL appropriately treats the uncertainty in trust, AT offers more accurate trust assessments, compared to the topology- and graph-based solutions. On the other hand, as AT aims at computing indirect trust between users, it outperforms the probability based models that focus only on direct trust. Experiment results demonstrate that AT achieves the most accurate trust assessment results. Specifically, AT achieves the F1 scores of 0.7 and 0.75, using the Advogato and Pretty Good Privacy (PGP) datasets, respectively. AT can rank users based on their trust values. We measure the accuracy of the ranking results, using the Kendall's tau coefficients. Experiment results show that, on average, AT offers 0.73 and 0.77 kendall's tau coefficients, in Advogato and PGP, respectively.

1.4 Technical Challenges and Solutions

The first technical challenge is that 3VSL needs to accurately model the trust propagation and fusion in OSNs. This is a challenge because trust propagation in OSNs is not well understood, although it is widely adopted by the research community. We address this challenge by using an opinion to represent trust and modeling trust propagation based on DC distribution and several commonly-accepted assumptions.

The second technical challenge is that 3VSL must be able to work on OSNs with non-series-parallel network topologies. This is a challenge because the only allowed operations in trust assessment are trust propagation and trust fusion. However, these two operations require that a network's topology must be either series and/or parallel. This requirement cannot be satisfied in real-world online social networks. We address this challenge by differentiating distorting opinions from original opinions. For example, if Alice trusts Bob and Bob trusts Charlie, then Alice's opinion on Bob is called the distorting opinion, and Bob's opinion on Charlie is the original opinion. We find that original opinions can be fused only once but distorting opinions can be combined any number of times. This discovery lays the foundation for the proposed recursive AssessTrust algorithm.

The third technical challenge is that 3VSL needs to handle social networks with arbitrary topologies, even with cycles. This is a challenge because it is impossible to test 3VSL in all possible network topologies. We address this challenge by mathematically proving 3VSL works in arbitrary networks. The proof is based upon the characteristics of Dirichlet distribution and the properties of different opinions in the trust computation process. In the end, the AssessTrust algorithm is designed to compute the trust between any two users in an OSN.

The rest of this paper is organized as follows. In Section 2, the background and terminologies of trust are introduced. In Section 3, we introduce the 3VSL model and define the trust propagation and fusion operations. We then differentiate discounting opinions from original opinions in Section 4, and prove 3VSL can handle arbitrary network topologies. In the same section, we detail the proposed AssessTrust algorithm. In Section 5, we validate the 3VSL model and the AssessTrust algorithm, using two real-world datasets. The related work is given in Section 6. We conclude the paper in Section 7.

2 Background

2.1 Terminology

In this section, we briefly introduce some terminologies frequently referred in this paper. Trust assessment is defined as the process that a *trustor* assesses a *trustee* on whether it will *perform a certain task as expected*. As such, trust can be either *direct* or *indirect* [29]. Direct trust is formed from a trustor's direct interactions with a trustee while indirect trust is inferred from others' recommendations. Typically, trust is represented as an *opinion*, indicating how much a trustor trusts a trustee.

To model trust propagation and trust fusion, two opinion operations, *i.e.*, the discounting operation and combining operation, are design to facilitate trust computation/assessment [29]. Trust fusion refers to combining different trust opinions to form a consensus trust opinion. Trust propagation refers to a trust opinion being transferred from a user to another. For example, if A trusts B, and B trusts C, then B's opinion on C will be discounted by A to derive an indirect opinion of C's trust.

2.2 Subjective Logic

To better understand 3VSL, we first briefly introduce the subjective logic (SL) [28]. Considering two users A and X, A's opinion on X's trust can be described by an *opinion*.

$$\omega_{AX} = \langle \alpha_{AX}, \beta_{AX}, 2 \rangle \left| a_{AX} \right.,$$

where α_{AX}, β_{AX}, and 2 denote the amounts of evidence supporting user X is trustworthy, untrustworthy, and uncertain, respectively. Based on α_{AX} and β_{AX}, a Beta distribution can be formed to model A's trust in X.

In SL, the amount of uncertain evidence in an opinion is always 2. a_{AX} is called the base rate and formed from existing impression without solid evidence, *e.g.*, prejudice, preference, or a general opinion obtained from hearsay. For example, if A always distrusts/trusts the users from a certain group where X belongs to, then a_{AX} will be smaller/greater than 0.5. Note that the opinion in SL [28] was defined as $\omega_{AX} = \langle b_{AX}, d_{AX}, u_{AX} \rangle \left| a_{AX} \right.$, which is another form of the trust opinion. The connection between these two opinion representations is mentioned in [28] as follows.

$$\begin{aligned} b_{AX} &= \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX} + 2} \\ d_{AX} &= \frac{\beta_{AX}}{\alpha_{AX} + \beta_{AX} + 2} \\ u_{AX} &= \frac{2}{\alpha_{AX} + \beta_{AX} + 2} \end{aligned} \tag{2.1}$$

Leveraging the property of Beta distribution, two opinions can be fused, by combining the corresponding Beta distributions, to yield a new opinion. For example, opinions $\omega_1 = \langle \alpha_1, \beta_1, 2 \rangle \left| a_1 \right.$ and $\omega_2 = \langle \alpha_2, \beta_2, 2 \rangle \left| a_2 \right.$ can be combined to produce $\omega_{12} = \langle \alpha_{12}, \beta_{12}, 2 \rangle \left| a_{12} \right.$, where α_{12}, β_{12} and a_{12} are calculated as follows.

$$\left\{ \begin{array}{l} \alpha_{12} = \alpha_1 + \alpha_2 \\ \beta_{12} = \beta_1 + \beta_2 \\ a_{12} = \dfrac{a_1 + a_2}{2} \end{array} \right. .$$

Let A and B denote two users where $\omega_1 = \langle \alpha_1, \beta_1, 2 \rangle \left| a_1 \right.$ is A's opinion about B's trust. Assume C is another user and $\omega_2 = \langle \alpha_2, \beta_2, 2 \rangle \left| a_2 \right.$ is B's opinion about C's trust. Then, the discounting operation is applied to compute A's indirect opinion about C's trust $\omega_{AC} = \langle \alpha_{12}, \beta_{12}, 2 \rangle \left| a_{12} \right.$ where

$$\left\{ \begin{array}{l} \alpha_{12} = \dfrac{\alpha_1 \alpha_2}{(\beta_2 + \alpha_2 + 2)(\beta_1 + \alpha_1 + 2)} \cdot \dfrac{2}{\kappa} \\ \alpha_{12} = \dfrac{\alpha_1 \beta_2}{(\beta_2 + \alpha_2 + 2)(\beta_1 + \alpha_1 + 2)} \cdot \dfrac{2}{\kappa} \\ a_{12} = a_2 \end{array} \right. ,$$

and

$$\kappa = 1 - \frac{(\alpha_1 \alpha_2 + \alpha_1 \beta_2)}{(\beta_2 + \alpha_2 + 2)(\beta_1 + \alpha_1 + 2)}.$$

3 Three-Valued Subjective Logic

The major limitation of the SL model is that the uncertainty in trust is considered a constant, however, the uncertainty in a trust opinion will be increased when it propagates from a user to another. To address this issue, we propose the three-valued subjective logic (3VSL) to model trust between users in an OSN, by redefining the uncertainty in trust. Designing the 3VSL model is a challenging task as trust propagation in OSNs is not well understood, although it is widely used in many applications. We address this challenge by modeling trust as an opinion, a representation of a probabilistic distribution over three different states, *i.e.*, trustworthy, untrustworthy, and uncertain. By investigating how these states of an opinion change during trust propagation, we redesign the trust discounting operation. Leveraging the Dirichlet distribution, we also redesign the combining operation. Moreover, we discover the mechanism of how to correctly apply these opinion operations on trust assessment within an OSN, leading to the design of the AssessTrust algorithm.

3.1 A Probabilistic Interpretation of Trust

Trust in 3VSL is defined as the probability that a trustee will behave as expected in the future. The probability is determined by the amounts of evidence that a trustor observed about a trustee's historical behaviors. A trustee may be observed behaving as expected, not expected, or in an ambiguous way. As a result, a trustor obtains positive, negative, and uncertain evidence accordingly. Based on the observed evidence, Bayesian inference is used to infer the probability of a trustee being trustworthy, or the probability that a trustee will behave as expected in the future. In summary, given more positive observed evidence, the probability of a trustee being trustworthy is larger.

The uncertainty state in 3VSL not only contains the observed uncertain evidence but also the distorted evidence when trust propagates in the network. Knowing how much evidence is distorted will give us an idea of how much positive (and negative) evidence left, which must be accurate so that the probability inference (of trust) could be precise. Without keeping track of uncertainty evidence, the amount of certain evidence in an opinion becomes incorrect, leading to erroneous trust assessments.

A trustee's future behavior can be modeled as a random variable x that takes on one of three possible outcomes $\{1, 2, 3\}$, *i.e.*, $x = 1$, $x = 2$ and $x = 3$ indicating the trustee will behave as expected, not as expected, or in an ambiguous way, respectively. As such, we are interested in the probability that $x = 1$, which is determined by the positive observed behaviors of the trustee. Therefore, the probability density function (pdf) of x follows the Categorical distribution.

$$f(x|\mathbf{p}) = \prod_{i=1}^{3} p_i^{[x=i]},$$

where $\mathbf{p} = (p_1, p_2, p_3)$ and $p_1 + p_2 + p_3 = 1$, p_i represents the probability of observing event i. The Iverson bracket $[x = i]$ evaluates to 1 if $x = i$, and 0 otherwise.

If the value of $\mathbf{p}$ is available, the pdf of x will be known and the probability of $x = i$ can be computed. Unfortunately, $\mathbf{p}$ is an unknown parameter and needs to be estimated based on the observations of x. We treat $\mathbf{p}$ as three random variables that follow the Dirichlet distribution.

$$\mathbf{p} \sim Dir(\alpha, \beta, \gamma),$$

where α, β, γ are hyper-parameters that control the shape of the Dirichlet distribution. We assume $\mathbf{p}$ follows Dirichlet distribution mainly because it is a conjugate prior of categorical distribution. In addition, because Dirichlet distribution belongs to a family of continuous multivariate probability distributions, we have various pdfs for $\mathbf{p}$ by changing the values of α, β, γ.

$$f(\mathbf{p}) = C{p_1}^{\alpha-1}{p_2}^{\beta-1}{p_3}^{\gamma-1}, \tag{3.1}$$

where C is a normalizing factor ensuring $p_1 + p_2 + p_3 = 1$. In this way, we use $\mathbf{p} \sim Dir(\alpha, \beta, \gamma)$ to model the uncertainty in estimating $\mathbf{p}$.

With the mathematical model in place, $\mathbf{p}$ can be estimated based on the observations of x, according to the Bayesian inference. Given a set of independent observations of x, denoted by $\mathbf{D} = \{x_1, x_2, \cdots, x_n\}$ where $x_j \in \{1, 2, 3\}$ and $j = 1, 2, \cdots, n$, we want to know how likely $\mathbf{D}$ is observed. This probability can be computed as

$$P(\mathbf{D}|\mathbf{p}) = \prod_{j=1}^{n} p_1^{[x_j=1]} p_2^{[x_j=2]} p_3^{[x_j=3]}.$$

Let c_i denote the number of observations where $x = i$, then the above equation becomes $p_1^{c_1} p_2^{c_2} p_3^{c_3}$. Based on Bayesian inference, given observed data $\mathbf{D}$, the posterior pdf of $\mathbf{p}$ can be estimated from

$$f(\mathbf{p}|\mathbf{D}) = \frac{P(\mathbf{D}|\mathbf{p})f(\mathbf{p})}{P(\mathbf{D})},$$

where $P(\mathbf{D}|\mathbf{p}) = p_1^{c_1} p_2^{c_2} p_3^{c_3}$ is the likelihood function, and $f(\mathbf{p})$ the prior pdf of $\mathbf{p}$. $P(\mathbf{D})$ is the probability that $\mathbf{D}$ is observed, which is independent of $\mathbf{p}$. Therefore, we have

$$f(\mathbf{p}|\mathbf{D}) \propto p_1^{c_1} p_2^{c_2} p_3^{c_3} \times p_1^{\alpha-1} p_2^{\beta-1} p_3^{\gamma-1}.$$

That means the posterior pdf $f(\mathbf{p}|\mathbf{D})$ can be modeled by another Dirichlet distribution $Dir(\alpha + c_1, \beta + c_2, \gamma + c_3)$. With the posterior pdf of $\mathbf{p}$, we have the following predicative model for x.

$$f(x|\mathbf{D}) = \int f(x|\mathbf{p}) f(\mathbf{p}|\mathbf{D}) d\mathbf{p}. \tag{3.2}$$

This function is in fact a composition of Categorical ($f(x|\mathbf{p})$) and Dirichlet ($f(\mathbf{p}|\mathbf{D})$) distributions, so it is called Dirichlet-Categorical (DC) distribution [34].

3.2 Opinion

In the previous section, we introduce how to model a trustee's future behavior by a DC distribution. From a DC distribution, the probability that the trustee is trustworthy can be derived from Eq. 3.2. Because the shape of a DC distribution is determined by three parameters, we use these parameters to form a vector to represent it. This vector is called *opinion* that expresses a trustor's opinion about a trustee's trust.

For a given DC distribution, the only undetermined parameters are α, β, γ. We set $\alpha = \beta = \gamma = 1$, if there is no observed data, *i.e.*, $\mathbf{D} = \emptyset$. In this case, the DC distribution yields a uniform distribution, *i.e.*, $p_1 = p_2 = p_3 = 1/3$. Assuming $\mathbf{p}$ initially follows uniform distribution is reasonable because we make no observation of x, and the best choice is to believe that x could be 1, 2, or 3 with equal probability. As more observations of x are made, the pdf of $\mathbf{p}$ becomes more accurate.

From Eq. 3.2, we can compute the probability of $x = 1$, *i.e.*, whether a trustee will behave as expected. In other words, we can use Eq. 3.2 to infer the trust of the trustee. Specifically, we can obtain the expectation of the probability that the trustee will behave as expected as follows.

$$\begin{aligned}
&P(x = 1|\mathbf{D}) \\
&= \int P(x = 1|p_1, p_2, p_3) P(p_1, p_2, p_3|c_1, c_2, c_3) d(p_1, p_2, p_3) \\
&= \frac{\Gamma(c_1 + c_2 + c_3)}{\Gamma(c_1)\Gamma(c_2)\Gamma(c_3)} \int p_1^{c_1-1} p_2^{c_2-1} p_3^{c_3-1} \\
&= \frac{\Gamma(c_1 + c_2 + c_3)\Gamma(c_1 + 1)\Gamma(c_2)\Gamma(c_3)}{\Gamma(c_1)\Gamma(c_2)\Gamma(c_3)\Gamma(c_1 + c_2 + c_3 + 1)} \\
&= \frac{c_1}{c_1 + c_2 + c_3},
\end{aligned} \tag{3.3}$$

where $\Gamma(n) = (n-1)!$ is the Gamma function. In the same way, the probabilities that the trustee will behave not as expected, or in an ambiguous way, can be computed from

$$P(x = 2|\mathbf{D}) = \frac{c_2}{c_1 + c_2 + c_3},$$

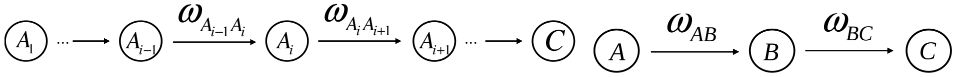
and

$$P(x = 3|\mathbf{D}) = \frac{c_3}{c_1 + c_2 + c_3}.$$

If the hyper-parameters α, β, γ equal to 1, the future behavior of the trustee is only determined by c_1, c_2, c_3, *i.e.*, the numbers of observations collected when the trustee behaved as expected, not as expected, or in an ambiguous way. We name these observations as positive, negative, and uncertain evidence. From a trustor A's perspective, a trustee X's future behavior can be modeled a DC distribution that is represented as an opinion.

$$\omega_{AX} = \langle \alpha_{AX}, \beta_{AX}, \gamma_{AX} \rangle \, | a_{AX}.$$

Here, ω_{AX} denotes A's opinion on X's future behavior, or A's trust in X behaving as expected. The parameters $\alpha_{AX}, \beta_{AX}, \gamma_{AX}$ refer to the amounts of observed positive, negative and uncertain evidence, respectively. We further name them as the *belief*, *distrust* and *uncertainty* parameters, in the rest of the paper. The subscripts of $\alpha_{AX}, \beta_{AX}, \gamma_{AX}$ differentiate them from the prior α, β, γ, *i.e.*, the former represents observed evidence while the latter is always $(1, 1, 1)$.



(a) A general illustration of series topology.

(b) A simple example of series topology.

Figure 1: Examples of series topologies

3.3 Discounting Operation

Trust propagation in OSNs was well-known, however, there is a lack of understanding about how to computationally model the process in practice. Trust propagation can be illustrated by a series topology, as shown in Fig. 1(a). In the figure, two edges are connected in *series* if they are incident to a vertex of degree 2. Trust propagation means that if user A_{i-1} trusts A_i and A_i trusts A_{i+1}, then A_{i-1} can derive an indirect trust of A_{i+1}, even if A_{i-1} did not interact with A_{i+1} before.

Based on existing literature on trust propagation [35, 36, 37, 38], it is commonly agreed that the following assumptions hold.

- A1: If A trusts B, B trusts C, then A trusts C.
- A2: If A trusts B, B does not trust C, then A does not trust C.
- A3: If A trusts B, B is uncertain about the trust of C, then A is uncertain about C's trust.
- A4: If A does not trust B, or A is uncertain about B, then A is uncertain about the trust of C.

It is worth mentioning that if A does not trust or is uncertain about B, then A is uncertain about C, and B's opinion on C cannot propagate to A. Based on the above-mentioned four assumptions, the trust propagation process can be modelled by the logic operation on two trust opinions.

Let's denote A's opinion on B as

$$\omega_{AB} = \langle \alpha_{AB}, \beta_{AB}, \gamma_{AB} \rangle \,,$$

and B's opinion on C as

$$\omega_{BC} = \langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle \,,$$

where $\{\alpha_{AB}, \beta_{AB}, \gamma_{AB}\} = \mathbf{D}_{AB}$ and $\{\alpha_{BC}, \beta_{BC}, \gamma_{BC}\} = \mathbf{D}_{BC}$ represent the observations made by A and B, about B and C, respectively. In this way, the expected probability that C will behave as what A expects will be

$$\begin{aligned} &\iint (x = 1|\mathbf{p}_{AB}) f(\mathbf{p}_{AB}|\mathbf{D}_{AB}) \times \\ &f(x = 1|\mathbf{p}_{BC}) f(\mathbf{p}_{BC}|\mathbf{D}_{BC}) d(\mathbf{p}_{AB}) d(\mathbf{p}_{BC}). \end{aligned} \tag{3.4}$$

The equation makes sense because A trusts C if and only if A trusts B and B trusts C, which is the assumption A1. In other words, the probability that C will behave as what A expects is equal to the probability that C will behave as what B expects, if A trusts B. In the above equation, $f(x = 1|\mathbf{p}_{AB}) f(\mathbf{p}_{AB}|\mathbf{D}_{AB})$ gives the probability that A trusts B, and $f(x = 1|\mathbf{p}_{BC}) f(\mathbf{p}_{BC}|\mathbf{D}_{BC}) d(\mathbf{p}_{AB})$ is the probability that B trusts C.

Because the probabilities that A trusts B and B trusts C are independent of each other, Eq. 3.4 can be rewritten as

$$\begin{aligned} &\int f(x = 1|\mathbf{p}_{AB}) f(\mathbf{p}_{AB}|\mathbf{D}_{AB}) d(\mathbf{p}_{AB}) \times \\ &\int f(x = 1|\mathbf{p}_{BC}) f(\mathbf{p}_{BC}|\mathbf{D}_{BC}) d(\mathbf{p}_{BC}). \end{aligned} \tag{3.5}$$

The two integrals in the equation are used to compute the expected probabilities that A trusts B and B trusts C, respectively. According to Eq. 3.3, we know

$$\begin{aligned} &\int f(x = 1|\mathbf{p}_{AB}) f(\mathbf{p}_{AB}|\mathbf{D}_{AB}) d(\mathbf{p}_{AB}) \\ = \;& \frac{\alpha_{AB}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \end{aligned}$$

and

$$\begin{aligned} & \int f(x=1|\mathbf{p}_{BC})f(\mathbf{p}_{BC}|\mathbf{D}_{BC})d(\mathbf{p}_{BC}) \\ = & \frac{\alpha_{BC}}{\alpha_{BC}+\beta_{BC}+\gamma_{BC}}. \end{aligned}$$

Inserting these two values into Eq. 3.5, we have the probability that C will behave as what A expects as

$$\frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})}. \tag{3.6}$$

According to assumption A2, the probability that C will not behave as what A expects is

$$\begin{aligned} & \iint f(x=1|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{AB}) \times \\ & f(x=2|\mathbf{p}_{BC})f(\mathbf{p}_{BC}|\mathbf{D}_{BC})d(\mathbf{p}_{AB})d(\mathbf{p}_{BC}). \end{aligned} \tag{3.7}$$

This is because A does not trust C, if and only if A trusts B but B does not trust C. Because these probabilities are independent, we have the expected probability that A does not trust C as

$$\frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})}. \tag{3.8}$$

Finally, based on assumptions A3 and A4, the expected probability that C will behave in an ambiguous way is

$$\begin{aligned} & \iint f(x=1|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{AB})\times \\ & f(x=3|\mathbf{p}_{BC})f(\mathbf{p}_{BC}|\mathbf{D}_{BC})+ \\ & f(x=2|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{AB})+f(x=3|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{AB}) \\ & d(\mathbf{p}_{AB})d(\mathbf{p}_{BC}). \end{aligned}$$

The expected probability can be rewritten as

$$\frac{\alpha_{AB}\gamma_{BC}+(\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})}. \tag{3.9}$$

The summation of Eqs. 3.4, 3.7 and 3.9 equals 1. The three equations actually give the estimated probabilities that C will behave as expected, not as expected, or in an ambiguous way, respectively. Putting these values back into the Categorical distribution

$$f(x|\mathbf{p}_{AC})=\prod_{i=1}^{3}p_i^{[x=i]}, \tag{3.10}$$

where $\mathbf{p}_{AC}=(p_1,p_2,p_3)$ and

$$\begin{aligned} p_1 &= \frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})} \\ p_2 &= \frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})}, \\ p_3 &= \frac{(\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})+\alpha_{AB}\gamma_{BC}}{(\alpha_{AB}+\beta_{AB}+\gamma_{AB})(\alpha_{BC}+\beta_{BC}+\gamma_{BC})} \end{aligned} \tag{3.11}$$

we can compute the probability that X is trustworthy.

From the above description, we know the Categorical distribution is in fact derived from B's opinion on C. Let's assume B made the observations $\mathbf{x}=\{x_1,x_2,\cdots,x_n\}$ on C. Then, we know $\alpha_{BC},\beta_{BC},\gamma_{BC}$ equal to the numbers of observations where $x=1,x=2,x=3$, respectively. Clearly, the observations B made about C do not reflect A's opinion on C. As trust propagates among users, A could derive an indirect opinion from B's observations of C. One may ask the question that if A was provided with the n observations, how many of them will be considered positive, negative, and uncertain, from A's perspective. That implies$A$ needs to reuse B's observations on C to derive its own opinion on C. For each $x_j \in \mathbf{x}$ where $j = 1, 2, \cdots, n$, we know x_j is observed, given the underlying categorical distribution shown in Eq. 3.10. In other words, $\mathbf{x}$ follows the multinomial distribution with parameters $(n, \mathbf{p}_{AC})$. From the multinomial distribution, we can recompute the probability that C is trustworthy, by re-categorizing the observations $\mathbf{x}$. We know the following re-categorization will occur with the highest probability.

$$\begin{aligned}
\alpha_{AC} &= p_1(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) \\
&= \frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\beta_{AC} &= p_2(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) \\
&= \frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\gamma_{AC} &= p_3(\alpha_{BC} + \beta_{BC} + \gamma_{BC}), \\
&= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) + \alpha_{AB}\gamma_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}.
\end{aligned} \tag{3.12}$$

Therefore, we use $\omega_{AC} = \langle \alpha_{AC}, \beta_{AC}, \gamma_{AC} \rangle$ to represent A's opinion about C's trust. Note that the opinion ω_{AC} is generated by distorting positive and negative evidence in ω_{BC} to uncertain evidence. That also means the total amount of evidence does not change in trust propagation.

$$\alpha_{AC} + \beta_{AC} + \gamma_{AC} = \alpha_{BC} + \beta_{BC} + \gamma_{BC}. \tag{3.13}$$

Based on the previous analysis, we formally define the discounting operation in 3VSL as follows.

Definition 1 (Discounting Operation) *Given three users A, B and C, if $\omega_{AB} = \langle \alpha_{AB}, \beta_{AB}, \gamma_{AB} \rangle$ is A's opinion on B's trust, and $\omega_{BC} = \langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle$ is B's opinion on C's trust, the discounting operation $\varDelta(\omega_{AB}, \omega_{BC})$ computes A's opinion on C as*

$$\varDelta(\omega_{AB}, \omega_{BC}) = \langle \alpha_{AC}, \beta_{AC}, \gamma_{AC} \rangle,$$

where

$$\begin{aligned}
\alpha_{AC} &= \frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\beta_{AC} &= \frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\gamma_{AC} &= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) + \alpha_{AB}\gamma_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}.
\end{aligned} \tag{3.14}$$

Opinion ω_{BC} being discounted can be viewed as the certain evidence in ω_{BC} are distorted by opinion ω_{AB}, and then transferred into the uncertainty space of ω_{AC}. Because the total amount of evidence in opinion $\omega_{AC} = \varDelta(\omega_{AB}, \omega_{BC})$ is the same as ω_{BC}'s, we conclude *the resulting opinion of discounting operation shares exactly the same evidence space as the original opinion.*

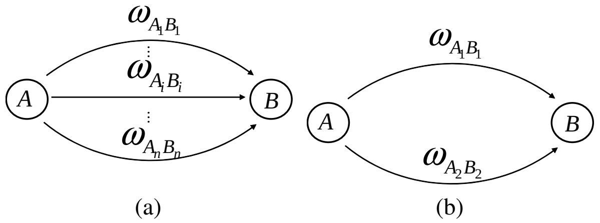
Based on the definition of discounting operation, it offers two interesting properties: decay and associative properties.

Corollary 3.1 *Decay Property: Given two opinions ω_{AB} and ω_{BC}, $\varDelta(\omega_{AB}, \omega_{BC})$ yields a new opinion ω_{AC}, where $\alpha_{AC} \le \alpha_{BC}$, $\beta_{AC} \le \beta_{BC}$ and $\gamma_{AC} > \gamma_{BC}$.*

Proof 1 *Because $\dfrac{\alpha_{AB}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})} \le 1$, according to Eq 3.14, we have $\alpha_{AC} \le \alpha_{BC}$ as well as $\beta_{AC} \le \beta_{BC}$. Hence, $-\alpha_{AC} - \beta_{AC} \ge -\beta_{AC} - \beta_{AC}$. According to Eq. 3.13, we have $\gamma_{AC} \ge \gamma_{BC}$.*

In other words, by applying the discounting operation, the uncertainty in trust (or in the resulting opinion) increases. This property implies that the more trust propagates among users in an OSN, the more uncertain the resulting opinion.

Corollary 3.2 *Associative Property: Given three opinions ω_{AB}, ω_{BC} and ω_{CD}, $\varDelta(\varDelta(\omega_{AB}, \omega_{BC}), \omega_{CD}) \equiv \varDelta(\omega_{AB}, \varDelta(\omega_{BC}, \omega_{CD}))$.*



(a) (b)

Figure 2: Examples of parallel topologies

Proof 2 *Simply based on Eq 3.14.*

However, the discounting operation is not commutative, *i.e.*, $\mathit{\Delta}(\omega_{AB}, \omega_{BC}) \neq \mathit{\Delta}(\omega_{BC}, \omega_{AB})$. Given a series topology where opinions are ordered as $\omega_{A_1A_2}, \omega_{A_2,A_3}, \cdots, \omega_{A_{n-1}A_n}$, the final opinion can be calculated as $\mathit{\Delta}(\mathit{\Delta}(\mathit{\Delta}(\omega_{A_1A_2}, \omega_{A_2A_3}), \cdots), \omega_{A_{n-1}A_n})$. As the discounting operation is associative, it can be simplified as $\mathit{\Delta}(\omega_{A_1A_2}, \omega_{A_2A_3}, \cdots \omega_{A_{n-1}A_n})$.

3.4 Combining Operation

According to previous works [35, 36, 37], trust opinions can be fused into a consensus one by aggregating the evidence from each opinion. We will use the parallel topology shown in Fig. 2(b) to explain how the combining operation works.

Let $\omega_{A_1B_1} = \langle \alpha_{A_1B_1}, \beta_{A_1B_1}, \gamma_{A_1B_1} \rangle$ and $\omega_{A_2B_2} = \langle \alpha_{A_2B_2}, \beta_{A_2B_2}, \gamma_{A_2B_2} \rangle$ be A's two indirect/direct opinions on B. We use $\{\alpha_{A_1B_1}, \beta_{A_1B_1}, \gamma_{A_1B_1}\} = \mathbf{D}_{A_1B_1}$ and $\{\alpha_{A_2B_2}, \beta_{A_2B_2}, \gamma_{A_2B_2}\} = \mathbf{D}_{A_2B_2}$ to represent the two sets of observations A made on B. According to the definition of an opinion, the expected probability that B will behave as what A expects can be computed from the following DC distribution.

$$\int f(x = 1|\mathbf{p}_{AB}) f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}) d(\mathbf{p}_{AB}), \tag{3.15}$$

The intuition of Eq. 3.15 can be explained as follows. A first infers the parameters $\mathbf{p}_{AB}$ by aggregating the observations $\mathbf{D}_{A_1B_1}$ and $\mathbf{D}_{A_2B_2}$, *i.e.*, the posterior pdf of $\mathbf{p}_{AB}$ becomes

$$f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}). \tag{3.16}$$

Then, based on the inferred parameters $\mathbf{p}_{AB}$, the probability that B will behave as what A expects can be computed from $f(x = 1|\mathbf{p}_{AB})$. Considering all possible values of $\mathbf{p}_{AB}$, we can obtain the Eq. 3.15.

Now, we will give the analytic form of Eq. 3.16 as follows. A first forms his opinion on B from $\mathbf{D}_{A_1B_1}$. As such, the pdf of $\mathbf{p}_{AB}$ is obtained. Then, A adjusts the estimate for $\mathbf{p}_{AB}$ based on another set of evidence $\mathbf{D}_{A_2B_2}$. As a matter of fact, Eq. 3.16 can be regarded as the distribution of $\mathbf{p}_{AB}$ based on (1) the posterior evidence in $\mathbf{D}_{A_2B_2}$ and (2) the prior parameters $\mathbf{p}_{A_1B_1}$ estimated from $\mathbf{D}_{A_1B_1}$. According to Bayesian inference, it can be expressed as follows.

$$\begin{aligned}
& f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}) \\
= \; & \frac{f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1})}{f(\mathbf{D}_{A_2B_2})} \\
= \; & \frac{f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1})}{\int f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1}) d\mathbf{p}_{A_1B_1}}.
\end{aligned} \tag{3.17}$$

In the equation, $\mathbf{p}_{A_1B_1}$ is derived from $\mathbf{D}_{A_1B_1}$ that follows Dirichlet distribution, so its pdf can be computed as follows.

$$\begin{aligned}
& f(\mathbf{p}_{A_1B_1}) \\
= \; & \frac{\mathit{\Gamma}(\alpha_{A_1B_1} + \beta_{A_1B_1} + \gamma_{A_1B_1})}{\mathit{\Gamma}(\alpha_{A_1B_1}) \mathit{\Gamma}(\beta_{A_1B_1}) \mathit{\Gamma}(\gamma_{A_1B_1})} \times \\
& (p_1)^{\alpha_{A_1B_1}-1} (p_2)^{\beta_{A_1B_1}-1} (p_3)^{\gamma_{A_1B_1}-1}.
\end{aligned}$$

(3.18)

On the other hand, because $\mathbf{D}_{A_2B_2}$ follows the multinomial distribution, derived from $\mathbf{p}_{A_1B_1}$, its pdf can be expressed as

$$\begin{array}{rl} & f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1}) \\ = & \dfrac{\mathit{\Gamma}(\alpha_{A_2B_2}+\beta_{A_2B_2}+\gamma_{A_2B_2}+1)}{\mathit{\Gamma}(\alpha_{A_2B_2}+1)\mathit{\Gamma}(\beta_{A_2B_2}+1)\mathit{\Gamma}(\gamma_{A_2B_2}+1)} \times \\ & (p_1)^{\alpha_{A_2B_2}}(p_2)^{\beta_{A_2B_2}}(p_3)^{\gamma_{A_2B_2}}. \end{array} \tag{3.19}$$

Substituting Eq. 3.18 and Eq. 3.19 for $f(\mathbf{p}_{A_1B_1})$ and $f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1})$ in Eq. 3.17, the analytic form of Eq. 3.16 will be

$$\begin{array}{rl} & f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1},\mathbf{D}_{A_2B_2}) \\ = & \dfrac{\mathit{\Gamma}(\alpha_{AB}+\beta_{AB}+\gamma_{AB})}{\mathit{\Gamma}(\alpha_{AB})\mathit{\Gamma}(\beta_{AB})\mathit{\Gamma}(\gamma_{AB})} \times \\ & (p_1)^{\alpha_{AB}-1}(p_2)^{\beta_{AB}-1}(p_3)^{\gamma_{AB}-1}, \end{array} \tag{3.20}$$

where

$$\begin{array}{rcl} \alpha_{AB} & = & \alpha_{A_1B_1}+\alpha_{A_2B_2}, \\ \beta_{AB} & = & \beta_{A_1B_1}+\beta_{A_2B_2}, \\ \gamma_{AB} & = & \gamma_{A_1B_1}+\gamma_{A_2B_2}. \end{array}$$

Obviously, Eq. 3.20 can be considered the most likely pdf of the following Dirichlet distribution.

$$Dir(\alpha_{A_1B_1}+\alpha_{A_2B_2},\beta_{A_1B_1}+\beta_{A_2B_2},\gamma_{A_1B_1}+\gamma_{A_2B_2}).$$

Then, the equation

$$\int f(x|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1},\mathbf{D}_{A_2B_2})d(\mathbf{p}_{AB})$$

can be regarded as a DC distribution upon observations $\{\alpha_{A_1B_1}+\alpha_{A_2B_2},\beta_{A_1B_1}+\beta_{A_2B_2},\gamma_{A_1B_1}+\gamma_{A_2B_2}\}$. The above description essentially reflects the important property of DC distribution: two DC distributions can be combined, by adding up the corresponding controlling hyper-parameters, to yield a new DC distribution.

According to the definition of an opinion, the analytic form of Eq. 3.15 can be expressed as

$$\begin{array}{rl} & \displaystyle\int f(x=1|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1},\mathbf{D}_{A_2B_2})d(\mathbf{p}_{AB}) \\ = & \dfrac{\alpha_{A_1B_1}+\alpha_{A_2B_2}}{\alpha_{A_1B_1}+\alpha_{A_2B_2}+\beta_{A_1B_1}+\beta_{A_2B_2}+\gamma_{A_1B_1}+\gamma_{A_2B_2}}. \end{array}$$

The probability that B will not behave as what A expects and the probability that B will behave in an ambiguous way can be expressed as

$$\begin{array}{rl} & \displaystyle\int f(x=2|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1},\mathbf{D}_{A_2B_2})d(\mathbf{p}_{AB}) \\ = & \dfrac{\beta_{A_1B_1}+\beta_{A_2B_2}}{\alpha_{A_1B_1}+\alpha_{A_2B_2}+\beta_{A_1B_1}+\beta_{A_2B_2}+\gamma_{A_1B_1}+\gamma_{A_2B_2}}, \end{array}$$

and

$$\begin{array}{rl} & \displaystyle\int f(x=3|\mathbf{p}_{AB})f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1},\mathbf{D}_{A_2B_2})d(\mathbf{p}_{AB}) \\ = & \dfrac{\gamma_{A_1B_1}+\gamma_{A_2B_2}}{\alpha_{A_1B_1}+\alpha_{A_2B_2}+\beta_{A_1B_1}+\beta_{A_2B_2}+\gamma_{A_1B_1}+\gamma_{A_2B_2}}, \end{array}$$

respectively. As such, we are able to formally define the combining operation as follows.

Definition 2 (Combining Operation) *Let $\omega_{A_1B_1} = \langle\alpha_{A_1B_1}, \beta_{A_1B_1}, \gamma_{A_1B_1}\rangle$ and $\omega_{A_2B_2} = \langle\alpha_{A_2B_2}, \beta_{A_2B_2}, \gamma_{A_2B_2}\rangle$ be the two opinions A has on B, the combining operation $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2})$ is carried out as follows.*

$$\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}) = \langle\alpha_{AB}, \beta_{AB}, \gamma_{AB}\rangle\,, \tag{3.21}$$

where

$$\begin{cases} \alpha_{AB} = \alpha_{A_1B_1} + \alpha_{A_2B_2} \\ \beta_{AB} = \beta_{A_1B_1} + \beta_{A_2B_2} \\ \gamma_{AB} = \gamma_{A_1B_1} + \gamma_{A_2B_2} \end{cases}. \tag{3.22}$$

It is worth mentioning that the combining operation yields two properties: commutative and associative proprieties.

Corollary 3.3 *Commutative Property: Given two independent opinions $\omega_{A_1B_1}$ and $\omega_{A_2B_2}$, $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}) \equiv \Theta(\omega_{A_2B_2}, \omega_{A_1B_1})$.*

Proof 3 *Based on Eq. 3.22.*

Corollary 3.4 *Associative Property: Given three independent opinions $\omega_{A_1B_1}$, $\omega_{A_2B_2}$ and $\omega_{A_3B_3}$, then $\Theta(\omega_{A_1B_1}, \Theta(\omega_{A_2B_2}, \omega_{A_3B_3})) \equiv \Theta(\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}), \omega_{A_3B_3})$.*

Proof 4 *Based on Eq. 3.22.*

If A has more than two opinions on B, e.g., $\omega_{A_1B_1}, \omega_{A_2B_2} \cdots \omega_{A_nB_n}$, these opinion can be combined by $\Theta(\Theta(\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}), \cdots), \omega_{A_nB_n})$. As combining operation is commutative and associative, it can be rewritten as $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}, \cdots \omega_{A_nB_n})$.

3.5 Expected Belief of An Opinion

With the proposed discounting and combining operations, the trust between two users in an OSN can be computed, which will be elaborated in details in Section 4. Note that the computed trust is in the form of an opinion. To transform an opinion into a trust value, i.e., the probability that a user is trustworthy, we need to design a mapping mechanism.

Given an opinion $\omega_{AX} = \langle\alpha_{AX}, \beta_{AX}, \gamma_{AX}\rangle$, it is of interest to know how likely X will perform the desired action(s) requested by A. We call this probability as the expected belief of ω_{AX}. Although α_{AX} denotes the belief of opinion ω_{AX}, components β_{AX}, γ_{AX} also need to be considered in computing the expected belief.

We know that α_{AX} and β_{AX} are the numbers of (negative and positive) certain evidence, so they must be used in computing the expected belief. γ_{AX} only records the uncertain evidence, so it should be omitted in the computation of expected belief. Ignoring uncertain evidence, DC distribution of ω_X^A is collapsed into a Beta-Categorical (BC) distribution.

$$\begin{aligned} & f(p_1, p_2\,|\alpha_{AX}, \beta_{AX}) \\ = \; & \frac{\Gamma(\alpha_{AX} + \beta_{AX})}{\Gamma(\alpha_{AX}) \cdot \Gamma(\beta_{AX})} \cdot (1 - p_1)^{\alpha_{AX}-1} p_2^{\beta_{AX}-1}. \end{aligned}$$

Consequently, the original opinion is collapsed into

$$\omega_{AX} = \langle\alpha_{AX}, \beta_{AX}\rangle\,.$$

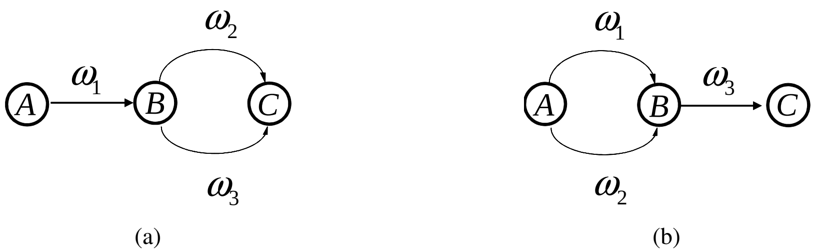
With the collapsed opinion, we apply the approach proposed in [39] to compute the expected belief as follows.

$$\begin{aligned} E_{\omega_{AX}} &= \left(\frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} + \frac{\beta_{AX}}{\alpha_{AX} + \beta_{AX}}\right) a_{AX} \\ &\times \; (1 - c_{AX}) + \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} \cdot c_{AX} \\ &= \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} \cdot c_{AX} + a_{AX} \cdot (1 - c_{AX}), \end{aligned} \tag{3.23}$$

where c_{AX} is the certainty factor [39] of a Beta distribution, and a_{AX} is the base rate. The certainty factor c_{AX}, ranging from 0 to 1, is determined by the total amount of certain evidence and the ratio between positive and negative evidence.

$$c_{AX} = \frac{1}{2} \int_0^1 \left| \frac{1}{B(\alpha_{AX}, \beta_{AX})} x^{\alpha_{AX}} (1 - x^{\beta_{AX}}) - 1 \right| dx. \tag{3.24}$$

Basically, c_{AX} approaches to 1 when the amount of certain evidence or the disparity between positive and negative evidence is large.



(a) (b)

Figure 3: Difference between distorting and original opinions

4 AssessTrust Algorithm

Based on 3VSL and the discounting and combining operations, we design the AssessTrust (AT) algorithm to conduct trust assessment in social networks with arbitrary topologies. Here, we treat a social network as a two-terminal directed graph (TTDG), in which the two terminals represent the trustor and trustee, respectively. Obviously, the trustor and trustee must be different users because a trustor will never evaluate the trust of itself. As a TTDG is not necessarily a directed acyclic graph, there may be cycles in the network.

To ensure AT works in arbitrary topologies, we need to first prove AT can handle non-series-parallel network topologies.This is a challenge because the only operations available for trust computation are the discounting and combining operations. The discounting/combining operation requires that the network topologies must be series/parallel. We address this challenge by differentiating distorting opinions from original opinions in trust propagation. For example, if A trusts B and B trusts C, then A's opinion on B is called the distorting opinion, and B's opinion on C is the original opinion. We discover that, in trust fusion, the original opinions can be used only once but the distorting opinions can be used any number of times. This is because the distorting opinion only depreciates certain evidence into uncertain evidence, *i.e.*, it does not change the total amount of evidence. On the other hand, when two (discounted) original opinions are combined, the total number of evidence in the resulting opinion will be increased.

In addition, we have to further show that AT works in arbitrary TTDGs. This is a challenge because it is impossible to test AT in all possible network topologies. We address this challenge by mathematically proving that AT works in arbitrary networks. After addressing these two challenges, we present the AT algorithm and use an example to illustrate how is works.

4.1 Properties of Different Opinions

For the two opinions involved in a discounting operation, their functionality are different, regarding to trust computation in an OSN.

Definition 3 (Distorting and Original Opinions) *Given a discounting operation $\varDelta(\omega_{AB}, \omega_{BC})$, we define ω_{AB} as the distorting opinion, and ω_{BC} the original opinion.*

To understand the difference between the distorting and original opinions, we study two special cases, as shown in Fig. 3. The detailed study reveals that a distorting opinion can be used several times in trust computation but an original opinion can be used only once.

Theorem 4.1 *Let $\omega_{B_1C_1} = \langle \alpha_{B_1C_1}, \beta_{B_1C_1}, \gamma_{B_1C_1} \rangle$ and $\omega_{B_2C_2} = \langle \alpha_{B_2C_2}, \beta_{B_2C_2}, \gamma_{B_2C_2} \rangle$ be two opinions B has on C. Let $\omega_{AB} = (\alpha_{AB}, \beta_{AB}, \gamma_{AB})$ be A's opinion on B, then we always have*

$$\begin{aligned} & \varTheta(\varDelta(\omega_{AB}, \omega_{B_1C_1}), \varDelta(\omega_{AB}, \omega_{B_2C_2})) \\ \equiv \;& \varDelta(\omega_{AB}, \varTheta(\omega_{B_1C_1}, \omega_{B_2C_2})). \end{aligned} \tag{4.1}$$

Proof 5 *Let's take a look at the left side of Eq. 4.1. According to the definition of discounting operation, the result of $\varDelta(\omega_{AB}, \omega_{B_1C_1})$ can be written as*

$$\begin{aligned} \omega_{AC_1} &= \varDelta(\omega_{AB}, \omega_{B_1C_1}) \\ &= \langle \alpha_{AC_1}, \beta_{AC_1}, \gamma_{AC_1} \rangle , \end{aligned}$$

where

$$\begin{aligned}
\alpha_{AC_1} &= \frac{\alpha_{AB}\alpha_{B_1C_1}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}, \\
\beta_{AC_1} &= \frac{\alpha_{AB}\beta_{B_1C_1}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}, \\
\gamma_{AC_1} &= \frac{(\beta_{AB}+\gamma_{AB})(\alpha_{B_1C_1}+\beta_{B_1C_1}+\gamma_{B_1C_1})}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_1C_1}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}.
\end{aligned} \tag{4.2}$$

The result of $\Delta(\omega_{AB}, \omega_{B_2C_2})$ *can be written as*

$$\begin{aligned}
\omega_{AC_2} &= \Delta(\omega_{AB}, \omega_{B_2C_2}) \\
&= \langle \alpha_{AC_2}, \beta_{AC_2}, \gamma_{AC_2} \rangle,
\end{aligned}$$

where

$$\begin{aligned}
\alpha_{AC_2} &= \frac{\alpha_{AB}\alpha_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}, \\
\beta_{AC_2} &= \frac{\alpha_{AB}\beta_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}, \\
\gamma_{AC_2} &= \frac{(\beta_{AB}+\gamma_{AB})(\alpha_{B_2C_2}+\beta_{B_2C_2}+\gamma_{B_2C_2})}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}.
\end{aligned} \tag{4.3}$$

If these two opinions are combined, we will have

$$\begin{aligned}
\omega_{AC} &= \Theta(\Delta(\omega_{A_1B_1}, \omega_{BC}), \Delta(\omega_{A_2B_2}, \omega_{BC})) \\
&= \langle \alpha_{AC}, \beta_{AC}, \gamma_{AC} \rangle,
\end{aligned}$$

where

$$\begin{aligned}
\alpha_{AC} &= \frac{\alpha_{AB}\alpha_{B_1C_1}+\alpha_{AB}\alpha_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}, \\
\beta_{AC} &= \frac{\alpha_{AB}\beta_{B_1C_1}+\alpha_{AB}\beta_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}},
\end{aligned}$$

$$\begin{aligned}
\gamma_{AC} &= \frac{(\beta_{AB}+\gamma_{AB})(\alpha_{B_1C_1}+\beta_{B_1C_1}+\gamma_{B_1C_1})}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_1C_1}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}} \\
&+ \frac{(\beta_{AB}+\gamma_{AB})(\alpha_{B_2C_2}+\beta_{B_2C_2}+\gamma_{B_2C_2})}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_2C_2}}{\alpha_{AB}+\beta_{AB}+\gamma_{AB}}.
\end{aligned}$$

Now, we look at the right side of Eq. 4.1. The term $\Theta(\omega_{B_1C_1}, \omega_{B_2C_2})$ *can be written as*

$$\begin{aligned}
\omega_{BC} &= \Theta(\omega_{B_1C_1}, \omega_{B_2C_2}) \\
&= \langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle,
\end{aligned} \tag{4.4}$$

where

$$\begin{aligned}
\alpha_{BC} &= \alpha_{B_1C_1}+\alpha_{B_2C_2}, \\
\beta_{BC} &= \beta_{B_1C_1}+\beta_{B_2C_2}, \\
\gamma_{BC} &= \gamma_{B_1C_1}+\gamma_{B_2C_2}.
\end{aligned}$$

Putting Eq. 4.5 back into the equation, we have

$$\begin{aligned} \omega'_{AC} &= \mathit{\Delta}(\omega_{AB}, \mathit{\Theta}(\omega_{B_1C_1}, \omega_{B_2C_2})) \\ &= \langle \alpha'_{AC}, \beta'_{AC}, \gamma'_{AC} \rangle \, , \end{aligned}$$

where

$$\begin{aligned} \alpha'_{AC} &= \frac{\alpha_{AB}(\alpha_{B_1C_1} + \alpha_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\ &= \frac{\alpha_{AB}\alpha_{B_1C_1} + \alpha_{AB}\alpha_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \\ \beta'_{AC} &= \frac{\alpha_{AB}(\beta_{B_1C_1} + \beta_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\ &= \frac{\alpha_{AB}\beta_{B_1C_1} + \alpha_{AB}\beta_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \\ \gamma'_{AC} &= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{B_1C_1} + \beta_{B_1C_1} + \gamma_{B_1C_1})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\ &+ \frac{\alpha_{AB}\gamma_{B_1C_1}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\ &+ \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{B_2C_2} + \beta_{B_2C_2} + \gamma_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\ &+ \frac{\alpha_{AB}\gamma_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}. \end{aligned} \tag{4.5}$$

Clearly, ω'_{AC} is equivalent to ω_{AC}.

Theorem 4.2 *Let $\omega_{A_1B_1} = (\alpha_{A_1B_1}, \beta_{A_1B_1}, \gamma_{A_1B_1})$ and $\omega_{A_2B_2} = (\alpha_{A_2B_2}, \beta_{A_2B_2}, \gamma_{A_2B_2})$ be A's two opinions on B. Let $\omega_{BC} = (\alpha_{BC}, \beta_{BC}, \gamma_{BC})$ be B's opinion on C, then the following equation* ***does not*** *hold.*

$$\begin{aligned} & \mathit{\Theta}(\mathit{\Delta}(\omega_{A_1B_1}, \omega_{BC}), \mathit{\Delta}(\omega_{A_2B_2}, \omega_{BC})) \\ \equiv \; & \mathit{\Delta}(\mathit{\Theta}(\omega_{A_1B_1}, \omega_{A_2B_2}), \omega_{BC}). \end{aligned} \tag{4.6}$$

Proof 6 *In Section 3, we have shown that the combining operation can be applied in $\mathit{\Theta}(\omega_{A_1B_1}, \omega_{A_2B_2})$ only if the evidence in $\omega_{A_1B_1}$ and $\omega_{A_2B_2}$ are independent. In the left side of Eq. 4.6, opinions $\mathit{\Delta}(\omega_{A_1B_1}, \omega_{BC})$ and $\mathit{\Delta}(\omega_{A_2B_2}, \omega_{BC})$ share the same evidence from the opinion ω_{BC}. As a result, the combining operation does not apply here. Therefore, $\mathit{\Delta}(\mathit{\Theta}(\omega_{A_1B_1}, \omega_{A_2B_2}), \omega_{BC})$ is the only correct solution, and it does not equal to $\mathit{\Theta}(\mathit{\Delta}(\omega_{A_1B_1}, \omega_{BC}), \mathit{\Delta}(\omega_{A_2B_2}, \omega_{BC}))$.*

From Theorems 4.1 and 4.2, we note that reusing ω_{AB} in case (a) is allowed but reusing ω_{BC} in case (b) is not.

The difference between ω_{AB} and ω_{BC} is that ω_{AB} is a distorting opinion while ω_{BC} is an original opinion. Therefore, we conclude that in trust computation, an original opinion can be combined only once, while a distorting opinion can be used any number of times, because it does not change the total amount of evidence in the resulting opinion.

4.2 Arbitrary Network Topology

As the distorting and original opinions are distinguished, we will prove that 3VSL is capable of handling non-series-parallel network topologies.

Theorem 4.3 *Given an arbitrary two-terminal directed graph $G = (V, E)$ where A, C are the first and second terminals, or the trustor and trustee. In the graph, a vertex u represents a user, the edge $e(u, v)$ denotes u's opinion about v's trust, denoted as ω_{uv}. By applying the discounting and combining operations, the resulting opinion ω_{AC} is solvable and unique.*

Proof 7 *We prove the theorem in a recursive manner,* i.e., *reducing the original problem into sub-problem(s) and continuing to reduce the sub-problems until the base case is solvable and yields a unique solution.*

As shown in Fig. 4, we assume there are m nodes $(c_1, c_2, \cdots, c_m)$ connecting to C, i.e., *$e(c_i, C) \in E$ where $i = 1, 2, \cdots, m$. There are n nodes $(a_1, a_2, \cdots, a_n)$ being connected from A,* i.e., *$e(A, a_j) \in E$ where $j = 1, 2, \cdots, n$.*

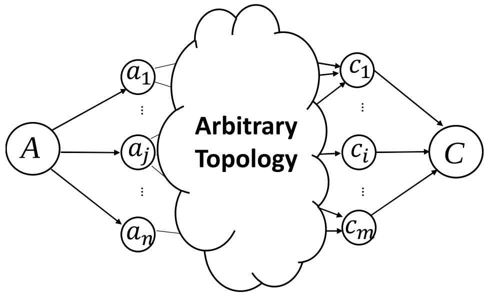


Figure 4: Illustration of an arbitrary network topology

Reduction rules

Case 1: If there is only one node connecting to C, i.e.*,* $m = 1$*, then* $\omega_{AC} = \mathit{\Delta}(\omega_{Ac_1}, \omega_{c_1 C})$*. In this case, we reduce the problem of computing* ω_{AC} *to calculating* ω_{Ac_1}*. We know A and* c_1 *are connected in a smaller sub-graph.*

Case 2: If there is more than one node connected to C, i.e.*,* $m > 1$*,* ω_{AC} *is equal to* $\mathit{\Theta}(\mathit{\Delta}(\omega_{Ac_1}, \omega_{c_1 C}), \mathit{\Delta}(\omega_{Ac_2}, \omega_{c_2 C}), \cdots, \mathit{\Delta}(\omega_{Ac_m}, \omega_{c_m C}))$ *due to Theorem 4.1. Therefore,* ω_{AC} *is solvable and unique if and only if each* ω_{Ac_i} *is solvable and unique, where* ω_{Ac_i} *can be obtained from the sub-graph* G'*, in which edges* $\{e(c_i, C)\}$ *and node C are removed from G. In this case, we reduce the problem of computing* ω_{AC} *to computing* ω_{Ac_i}*.*

In each round of reduction, G is reduced into a smaller graph, with $|E| = |E| - m$ *and* $|V| = |V| - 1$*. After applying the reduction rules on sub-problems recursively, the base case will be eventually reached,* i.e.*,* $|E| = 1$ *and* $|V| = 2$*.*

Base Case

The graph of base case contains only one edge from A to a_j *where* $j = 1, 2, \cdots, n$*. As* ω_{Aa_j} *is known, the base case is solvable and its solution is unique. Applying the equations in Case 1 and 2 repeatedly, we can obtain a unique solution to* ω_{AC}*.*

4.3 Differences between 3VSL and SL

The major difference between SL and 3VSL lies in the definition of uncertainty in the trust models. In 3VSL, the uncertainty in a trust opinion is measured by the number of uncertain evidence. However, the amount of uncertain evidence in a SL opinion is always 2. Because uncertain evidence is obtained if an ambiguous behavior of a trustee is observed, it could not be a constant number.

We take an example to explain the different definitions of uncertainty in SL and 3VSL models. Let's consider a series topology composed of A, B and C, as shown in Fig 1(b). We assume opinions ω_{AB} $\langle 5, 3, 2\rangle$ and $\omega_{BC} = \langle 4, 4, 2\rangle$. Then, A's opinion of C's trust can be computed by applying the discounting operation, defined in SL or 3VSL, on opinions ω_{AB} and ω_{BC}, *i.e.*, $\omega_{AC} = \mathit{\Delta}(\omega_{AB}, \omega_{BC})$. With the SL model, we have $\omega_{AC} = \langle 2/3, 2/3, 2\rangle$. Apparently, $10/3$ positive evidence and $10/3$ negative evidence are removed from the original evidence space. In other words, the amount of certain evidence shrinks for 83%, *i.e.*, 83% of evidence are distorted and disappear. Based on the SL model, we know the belief component b_{AB} in opinion ω_{AB} equals to $5/(5 + 3 + 2) = 0.5$, *i.e.*, with 50% of chance, A could trust B's recommendation. That also implies only 50% of evidence should be distorted from B's opinion of C, which is not the case in the example.

In contrast, 3VSL model introduces an uncertainty state to keep tracking of the uncertain evidence generated when trust propagates within an OSN. In 3VSL, we have $\omega_{AC} = \langle 2, 2, 6\rangle$. The total number of evidence in the resulting opinion ω_{AC} is the same as ω_{BC}, *i.e.*, $\alpha_{AC} + \beta_{AC} + \gamma_{AC} = \alpha_{BC} + \beta_{BC} + \gamma_{BC} = 10$. In fact, only 50% of certain evidence from α_{BC} and β_{BC} are transferred into γ_{AC}. Clearly, 3VSL leverages the uncertainty state to store the "distorted" positive and negative evidence in trust propagation and hence achieves better accuracy. This hypothesis will be validated in Section 5.

Another difference is that 3VSL is capable to handle a social network with arbitrary topologies while SL cannot. It is well-known that SL can only handle series-parallel network topologies. A series-parallel graph can be decomposed into many series (see Fig. 1) or parallel (see Fig. 2) sub-graphs so that every edge in the original graph will appear only once in the sub-graphs [40]. In real-world social networks, however, the connection between two users could be too complicated to be decomposed into series-parallel graphs. To apply the SL model, a complex topology has to be simplified into a series-parallel topology by removing or selecting edges [41, 42, 43]. However, it is not clear which edges need to be removed in a large-scale OSN. As a result, the solutions proposed in [41, 42, 43] cannot be implemented. In 3VSL, the difference between distorting and original opinions is first identified, and then a recursive algorithm is designed accordingly. The algorithm is able to process social networks with complex topologies, even with cycles.

4.4 AssessTrust Algorithm

```
Algorithm 1: AssessTrust($G$, $A$, $C$, $H$)
Require: $G$, $A$, $C$, and $H$.
Ensure: $\Omega_{AC}$.
 1: $n \leftarrow 0$
 2: if $H > 0$ then
 3:     for all incoming edges $e(c_i, C) \in G$ do
 4:         if $c_i = A$ then
 5:             $\Omega_i \leftarrow \omega_{c_iC}$
 6:         else
 7:             $G' \leftarrow G - e(c_i, C)$
 8:             $\Omega_{Ac_i} \leftarrow$ AssessTrust$(G', A, c_i, H-1)$
 9:             $\Omega_i \leftarrow \Delta(\Omega_{Ac_i}, \omega_{c_iC})$
10:         end if
11:         $n \leftarrow n + 1$
12:     end for
13:     if $n > 1$ then
14:         $\Omega_{AC} = \Theta(\Omega_1 \cdots \Omega_n)$
15:     else
16:         $\Omega_{AC} = \Omega_n$
17:     end if
18: else
19:     $\Omega_{AC} = \langle 0, 0, 0 \rangle$
20: end if
```

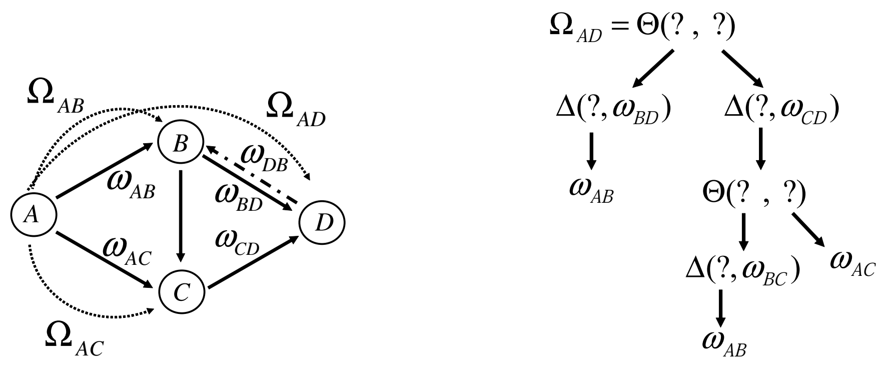
Based on Theorem 4.3, we design the AssessTrust algorithm, as shown in Algorithm 1. The algorithm is based on the 3VSL model and is able to handle any arbitrary network topologies. The inputs of AT algorithm include a social network graph G, a trustor A, a trustee C, and the maximum searching depth H, measured by number of hops. Specifically, H determines the longest distance the algorithm will search between the trustor and trustee. H controls the searching depth of the AT algorithm, which is necessary because G could be potentially very large.

To compute A's individual opinion on C, AT applies a recursive depth first search (DFS) on graph G, with a maximum searching depth of H. AT starts from the trustee C and visits all C's incoming neighbors c_i's, as shown in lines 1 to 12. For each node c_i, we denote A's opinion on C's trust obtained through c_i as Ω_i. At this moment, the opinion Ω_i is unknown unless c_i is the trustor node A. In this case, we have $\Omega_i = \omega_{c_iC} = \Omega_{AC}$. Otherwise, the value of Ω_i needs to be computed recursively by the AT algorithm. To do so, AT recalls itself on the new graph G' that keeps all the edges in the current graph except edge $e(c_i, C)$ and node C, as shown in line 7. The output of the AT algorithm, with G' as the input graph, will be A's opinion on c_i's trust, as shown in line 9. When all the incoming neighbors c_i's are processed, all the edges connecting to C will be removed from the graph as well. After that, if AT visits C again in the future, *i.e.*, C is involved in a cycle in G, the algorithm will stop as there is no incoming neighbor for C. In other words, cycles in graph G will be eliminated when AT searches the graph. A cycle involving a node essentially means the node holds a trust opinion about itself, which does not make sense as a node must absolutely trust itself. Therefore, it is meaningless to let a node to compute its own trust, levering others' opinions upon itself.

When the input graph becomes G', the trustee will be c_i and the maximum searching depth is decreased to $H - 1$, as shown in line 8. If there are more than one c_i, all the resulting opinions Ω_i's will be combined to yield the opinion Ω_{AC}, as shown in line 14. Otherwise, the only obtained opinion Ω_i will be assigned to Ω_{AC}, as shown in line 16. In the end, if the searching depth reaches H, AT return an empty opinion, as shown in line 19.

4.5 Illustration of the AssessTrust Algorithm

In this section, we will use the bridge topology shown in Fig. 5(a) to illustrate how the AT algorithm computes A's indirect opinion on C, denoted as Ω_{AD}. To differentiate from the direct opinion, we use Ω to denote the indirect



(a) Bridge topology

(b) Decomposition parsing parsing tree

Figure 5: An illustration of 3VSL based on the bridge topology

opinion. As shown in Fig. 5(a), to compute Ω_{AD}, discounting and combining operations are applied on opinions $\omega_{AB}, \omega_{AD}, \omega_{BD}, \omega_{CD}$, and ω_{BC}. AT starts from the trustee D and searches the network backwards, and recursively computes the trust of every node. As a result, we obtain a parsing tree, shown in Fig. 5(b), to indicate the correct order that discounting and combining operations are applied in computing A's opinion on D. By traversing the parsing tree in a bottom-up manner, A's indirect opinion about D can be computed as

$$\Theta\left(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\Delta(\omega_{AB}, \omega_{BC}), \omega_{AC}), \omega_{CD})\right). \tag{4.7}$$

To understand how exactly AT searches the bridge network, we use $AT^{(k)}(i, j)$ to denote it is for the kth time that AT is called, to compute the i's opinion on j. At the first time when AT is called, A's opinion on D is computed from

$$\Theta\left(\Delta(\Omega_{AB}, \omega_{BD}), \Delta(\Omega_{AC}, \omega_{CD})\right),$$

where Ω_{AB} and Ω_{AC} are A's indirect opinions on B and C, respectively. These two opinions will then be computed by $AT^{(2)}(A, B)$ and $AT^{(3)}(A, C)$, respectively. In $AT^{(3)}(A, C)$, AT computes A's opinion about C as

$$\Theta\left(\Delta(\Omega_{AB}, \omega_{BC}), \omega_{AC}\right),$$

where Ω_{AB} is computed by $AT^{(4)}(A, B)$. Finally, A's opinion on D can be computed from Eq. 4.7. In the bridge-topology network, the AT algorithm is called four times in total: $AT^{(1)}(A, D)$, $AT^{(2)}(A, B)$, $AT^{(3)}(A, C)$ and $AT^{(4)}(A, B)$. Note that the opinion output from $AT(A, B)$ is used twice, *i.e.*, in sub-graphs $A \to B \to C$ and $A \to B \to D \to C$, which is allowed in 3VSL.

The AT algorithm still works if a cycle is introduced in the graph, e.g., the edge from B to D is reversed. With the reversed edge DB, a loop $D \to B \to C \to D$ is formed. In the following, we will show how AT works on the graph with a cycle $D \to B \to C \to D$. The algorithm starts from D and visits C, and then recalls itself on graph G' in which D and edge CD are removed. The algorithm then reaches A and B. When it processes B, AT cannot visit D as D was already removed, so the algorithm quits. As such, the cycle $D \to B \to C \to D$ is eliminated while computing the indirect trust opinion Ω_{AD}.

4.6 Time Complexity Analysis

In this section, we present the time complexity of the AssessTrust algorithm. Because AT is a recursive algorithm, the recurrence equation of its time complexity is

$$\begin{aligned} T(n) &= (n-1) \cdot (T(n-1) + C_1) + C_2 + O(n-1) \\ &= (n-1) \cdot T(n-1) + O(n-1) + C, \end{aligned}$$

where $(n-1)$ is the maximum number of incoming edges to the trustee (line 3), assuming there are n nodes in the network. $T(n-1)$ is the time complexity of recursively running AT on each branch (line 8), C_1 is the time for lines $4-7$ and $9-11$. $O(n-1)$ is the time for combining operations (line 14). C_2 is the time used outside the "for" loop (line $13-20$). Therefore, the time complexity of AT is

$$O\left(\sum_{i=1}^{H} \frac{(n-1)!}{(n-1-i)!}\right) = O(n^H),$$

where H is the maximum searching depth, and n is the number of nodes in the network.

5 Evaluations

In this section, we evaluate the properties and performances of the 3VSL model and AT algorithm. We conduct comprehensive experiments to evaluate the accuracy of 3VSL model and compare its performance to that of subjective logic, in two real-world datasets: Advogato and PGP.

For the AT algorithm, we evaluate its accuracy and compare its performance to another trust assessment algorithm, called TidalTrust, in Advogato and PGP. We investigate the reasons why AT outperforms TidalTrust by analyzing the results obtained from these experiments.

To understand how accurate various models are in assessing trust within OSNs, we adopt F1 score [44] as the evaluating metric. The F1 score is chosen because it is a comprehensive measure for different models in predicting or inferring trust [44].

After evaluating the accuracy of different trust models, we evaluate the performance of the AT algorithm and compare it to these benchmark solutions: TrustRank and EigenTrust.

5.1 Dataset

The first dataset, Advogato, is obtained from an online software development community where an edge from user A to B represents A's trust on B, regarding B's ability in software development. The trust value between two users is divided into four levels, indicating different trust levels. The second dataset, Pretty Good Privacy (PGP), is collected from a public key certification network where an edge from user A to B indicates that A issues a certificate to B, *i.e.*, A trusts B. Similar to Advogato, the trust value is also divided into four levels.

According to the document provided by Advogato, a user determines the trust level of another user, based on only certain evidence. Therefore, a low-trust edge in Advogato indicates an opinion that contains negative evidence. On the other hand, in PGP, a user tends to give a low trust certification if he is not sure whether the other user is trustworthy or not. A user in PGP will never give a certification to anyone who has malicious behavior. Therefore, a low trust level in PGP indicates an opinion that contains uncertain evidence. We select these two datasets because they are obtained from real world OSNs where trust relations between users are quantified as non-binary values. In addition, the different definitions of trust in these two datasets allow us to evaluate the performance of 3VSL in different trust social networks. Statistics of these datasets are summarized in Table 1.

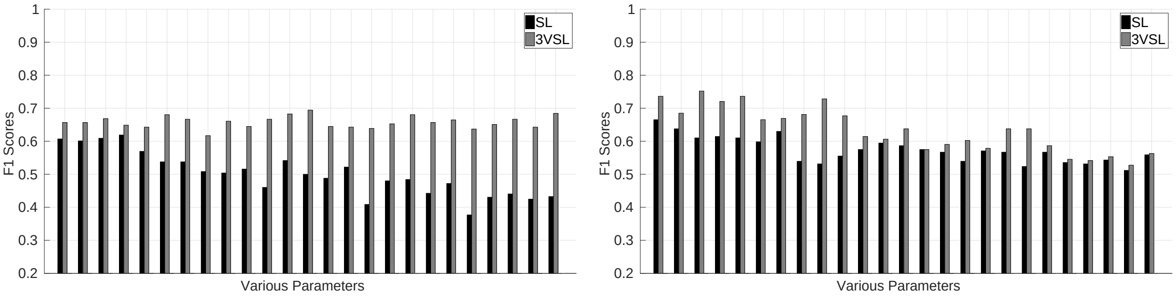
Table 1: Statistics of the Advogato and PGP datasets.

Dataset	# Vertices	# Edges	Avg Deg	Diameter
Advogato	6,541	51,127	19.2	4.82
PGP	38,546	31,7979	16.5	7.7

5.2 Dataset Preparation

In Advogato, trust is classified into four ordinal levels: *observer*, *apprentice*, *journeyer* and *master*. Similarly, in PGP, trust is classified into four levels: *0*, *1*, *2* and *3*. Both Advogato and PGP provide directed graphs where users are nodes and edges are the trust relations among users. Because the trust levels are in ordinal scales, a transformation is needed to convert a trust level into a trust value, ranging from 0 to 1.

In the experiments, we set the total evidence values λ as 10, 20, 30, 40, and 50. Given a certain λ, we can represent an opinion as $\left\langle \frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, \frac{\gamma}{\lambda} \right\rangle$. As aforementioned, the meanings of trust in Advogato and PGP are different, so we use different methods to construct opinions in Advogato and PGP. We assume the opinions in Advogato only contain positive and negative evidence, *i.e.*, $\gamma = 0$. Therefore, an opinion of 3VSL in Advogato can be expressed as $\left\langle \alpha, \lambda\left(1 - \frac{\alpha}{\lambda}\right), 0 \right\rangle$. Given the total number of evidence value λ, an opinion in Advogato is in fact determined by $\frac{\alpha}{\lambda}$, *i.e.*, the proportion of positive evidence. To properly set the value of $\frac{\alpha}{\lambda}$, we use the normal score transformation technique [45] to convert ordinal trust values into real numbers, ranging from 0 to 1. Specifically, trust levels are first converted into z-scores by the normal score transformation method, based on their distributions in the datasets. Then, we map the z-scores to different $\frac{\alpha}{\lambda}$'s, according to the differences among the z-scores. For example, the *master* level trust is converted into $(\frac{\alpha}{\lambda})_3 = 0.9$. For the *observer* level trust, we use different values of $(\frac{\alpha}{\lambda})_0$ as 0.1, 0.2, 0.3, 0.4 and 0.5 to indicate the possible lowest trust levels. With the highest and lowest values of $\frac{\alpha}{\lambda}$, we interpolate the values of $(\frac{\alpha}{\lambda})_1$ and $(\frac{\alpha}{\lambda})_2$ for *apprentice* and *journeyer* level trusts, based on the intervals between the corresponding z-scores. Because there are five different λ's and five different $(\frac{\alpha}{\lambda})_0$'s, we have a total of 25 combinations of parameters.



(a) Advogato

(b) PGP

Figure 6: F1 scores of 3VSL and SL using the A) Advogato and B) PGP dataset. Parameters are the combinations between base trust levels (0.1, 0.2, 0.3, 0.4 and 0.5) and total evidence values (10, 20, 30, 40, and 50)

For the PGP dataset, we assume there is only positive and uncertain evidence, so we set $\beta = 0$. Therefore, an opinion of 3VSL in PGP can be expressed as $\langle \alpha, 0, \lambda(1 - \frac{\alpha}{\lambda}) \rangle$. Similar to Advogato, an opinion in PGP is determined by λ and $\frac{\alpha}{\lambda}$. We use the same transformation method to convert the trust relations in PGP into opinions.

5.3 Accuracy of 3VSL Model

With the above-mentioned two datasets, we evaluate the accuracy of the 3VSL model. We also compare the accuracy of the 3VSL model to the SL model. As we know, SL does not model the trust propagation process correctly and its performance will degrade drastically in real-world OSNs. Due to this issue, SL cannot handle social networks with complex network topologies. Although some approximation solutions are proposed, *e.g.*, removing edges in a social network to reduce it into a simplified graph, there is no existing algorithm that implements any of these solutions. To make a fair comparison, we design an algorithm called SL*, based on the AT algorithm. The structure of the SL* algorithm is exactly the same as AT's, however, the discounting and combining operations used in the AT algorithm are replaced with those defined in SL. As such, SL* implements the SL model and is able to work on OSNs with arbitrary topologies.

The experiments are conducted as follows. First, we randomly select a trustor u from the datasets and find one of its 1-hop neighbors v. We take the opinion from u to v as the ground truth, *i.e.*., how u trusts v. Then, we remove the edge (u, v) from the datasets, if there is a path from u to v. We run the above-mentioned algorithms to compute u's opinion of v's trustworthiness. Finally, we compare the computed results to the ground truth. We select 200 pairs of u and v to get statistically significant results. To compare the computed results to the ground truth, we first use the expected beliefs of computed opinions as the trust values in 3VSL and SL. Then, we round the expected beliefs to the closest trust levels based on the ground truths. Finally, we use F1 score to evaluate the accuracy of different models. Because we do not know the correct parameter settings, we test the above-mentioned 25 combinations of parameters to conduct a comprehensive evaluation.

As shown in Fig. 6(a) and 6(b), 3VSL achieves higher F1 scores than SL, with all different parameter settings, in both datasets. Specifically, 3VSL achieves F1 scores ranging from 0.6 to 0.7 in Advogato, and 0.55 to 0.75 in PGP. On the other hand, the F1 scores of SL range from 0.35 to 0.6 in Advogato and 0.55 to 0.67 in PGP. Considering F1 score is within the range of $[0, 1]$, we conclude that 3VSL significantly outperforms SL.

More importantly, we observe that the F1 scores of 3VSL are relatively stable, with different parameter settings. However, the F1 scores of SL fluctuate, indicating SL is significantly affected by the parameter settings. Overall, we conclude that 3VSL is not only more accurate than SL but also more robust to different parameter settings.

We further investigate the reason why 3VSL outperforms SL by looking at the evidence values in the resulting opinions, computed by 3VSL and SL. We choose the results from experiments with the parameter setting (0.3, 30), wherein 3VSL performs the best. We are only interested in the cases where 3VSL obtains more accurate results than SL. We measure the values of certain evidence ($\alpha + \beta$) in the resulting opinions computed by 3VSL and SL. The CDFs of the values of certain evidence are then plotted in Fig. 7. As shown in Fig. 7, the values of $(\alpha + \beta)$ in the opinions computed by SL are much lower than that of 3VSL. It results in a lack of evidence in computing the expected beliefs of opinions by SL. This observation matches the example introduced in Section 4.3. Because 3VSL employs a third state to store the uncertainty generated in trust propagation, it is more accurate in modeling and computing trust in OSNs.

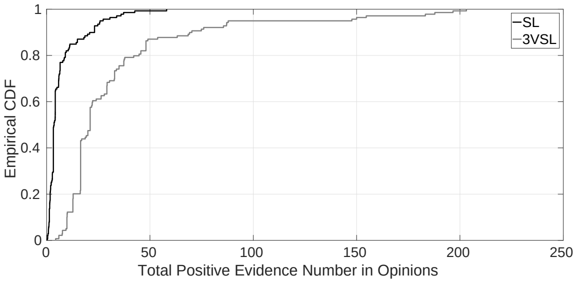


Figure 7: CDFs of $\alpha + \beta$ in opinions computed by 3VSL and subjective logic using the Advogato dataset.

	Advogato	PGP
AT	$(0.3, 30)$	$(0.1, 30)$
SL*	$(0.3, 30)$	$(0.1, 30)$
TT	$(0.2, -)$	$(0.1, -)$

Table 2: Selected parameters (base trust level, total evidence value) for AT, SL* and TT. Note that TT employs a number to represent trust, so its evidence value is empty.

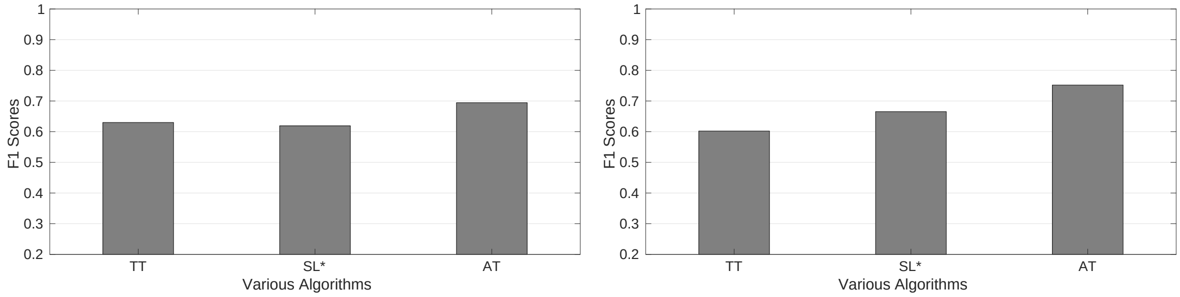
5.4 Performance of the AssessTrust Algorithm

After validating the 3VSL model, we study the performance of the AT algorithm and compare it to other benchmark algorithms, including TidalTrust (TT) [46], TrustRank (TR) [23] and EigenTrust (ET) [22]. TidalTrust is designed to compute the absolute trust of any user in an OSN. However, TR and ET are used to rank users in an OSN based on their relative trustworthiness, *i.e.*, it does not compute the absolute trust.

Because different benchmark algorithms solve the trust assessment problem differently, we conduct two groups of experiments. In the first group of experiments, we compare the performance of AT, SL* and TT in computing the absolute trustworthiness of users in an OSN. In the experiments, we randomly select a trustor u from the datasets and choose one of its 1-hop neighbors v. We take the opinion from u to v as the ground truth. Then, we remove the edge (u, v) from the datasets, if there exist paths from u to v in the network. We run the AT, SL* and TT algorithms to compute the trustworthiness of v, from u's perspective. Finally, we compare the computed trustworthiness to the ground truth.

Different parameters will affect the performances of various algorithms, so we choose different parameters for AT and TT so that they can perform well in the experiments. Because we already validated that 3VSL outperforms SL, regardless of the parameter settings, we choose the same parameter setting used by AT for SL*. The parameter settings for different algorithms in different datasets are shown in Table 2.

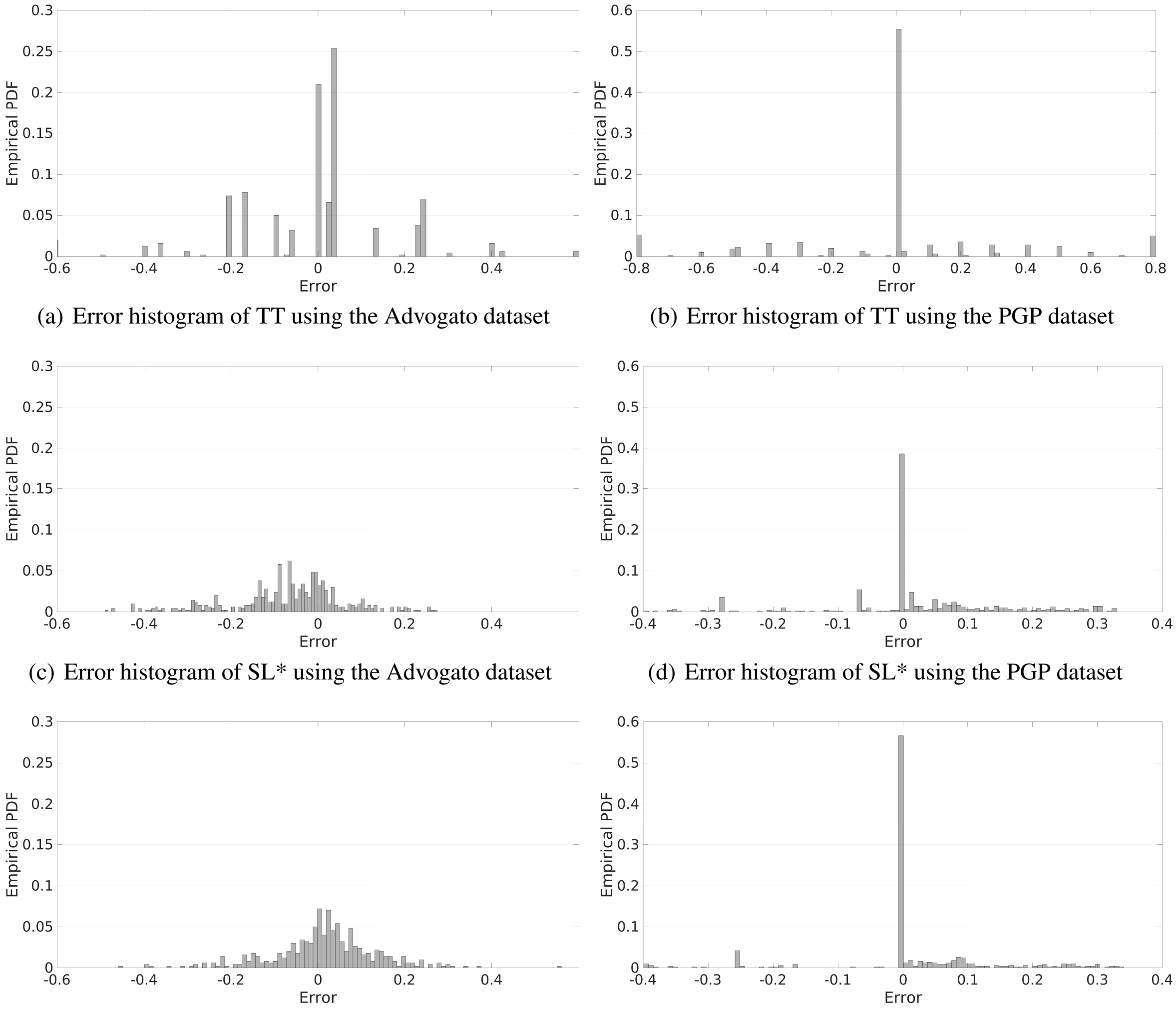
We first look at the F1 scores of the trust assessment results generated by the three algorithms. The F1 scores are plotted in Figs. 8(a) and 8(b). As shown in Figs. 8(a) and 8(b), AT outperforms TT in both datasets, *i.e.*, TT achieves 0.617 and 0.605 F1 scores, and AT offers 0.7 and 0.75 F1 scores in Advogato and PGP. It is worth mentioning that SL* gives the worst F1 scores, indicating that the problem of subjective logic in modeling uncertainty seriously impacts its performance.



(a) Advogato

(b) PGP

Figure 8: F1 scores of the trust assessment results generated by TT, SL* and AT using the a) Advogato and b) PGP datasets.



(a) Error histogram of TT using the Advogato dataset

(b) Error histogram of TT using the PGP dataset

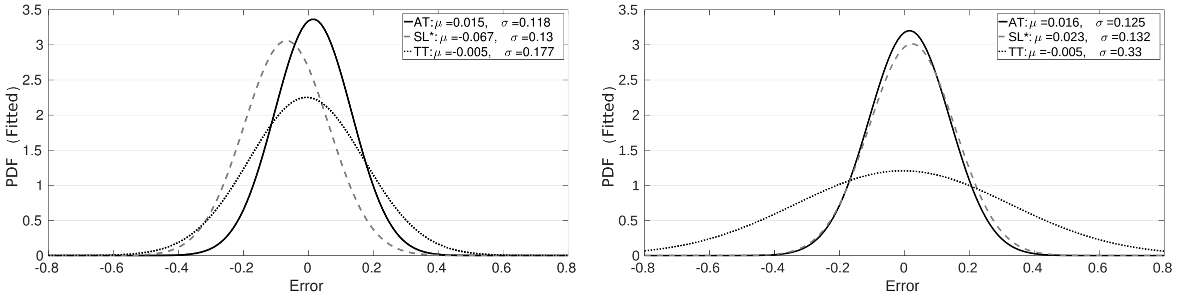
(c) Error histogram of SL* using the Advogato dataset

(d) Error histogram of SL* using the PGP dataset

(e) Error histogram of AT using the Advogato dataset

(f) Error histogram of AT using the PGP dataset

Figure 9: Histogram of the errors generated by TT, SL* and AT using the Advogato and PGP dataset.



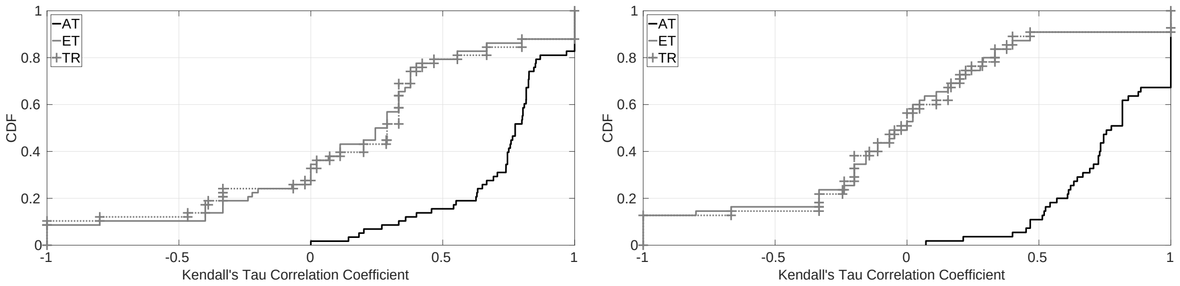
(a) Advogato

(b) PGP

Figure 10: Fitted curves of the error distributions of TT, SL* and AT using the a) Advogato and b) PGP dataset.

Besides F1 scores, we also study the distribution of errors in trust assessment results. The error here is defined as the difference between the computed trust value and the ground truth. The error distributions of different algorithms are shown in Figs. 9.

From Fig. 9(a), we can see that the errors of TT algorithm is either very small or very large when it is used to assess trust using the Advogato dataset. For the SL* and AT algorithms, however, the errors are more concentrated around 0,



(a) Advogato

(b) PGP

Figure 11: The CDFs of Kendall's tau ranking correlation coefficients of different algorithms using the a) Advogato and b) PGP dataset.

as shown in Figs. 9(c) and 9(e). If the PGP dataset is used, we observe the same phenomena, as shown in Figs. 9(b), 9(d) and 9(f).

We further fit this histogram data using the Normal Distribution. As shown in Figs 10(a) and 10(b), the fitted curves of the error distributions of different algorithms clearly indicate that AT gives the best trust assessment results. In these figures, we can see that the error distribution of TT has a close-to-zero mean, *i.e.*, 0.005 for both datasets, but a large variance. On the contrary, the fitted curves of the error distributions of SL* show that SL* has a smaller variance but a large mean, *i.e.*, 0.067 in Advogato and 0.016 in PGP. The fitted curves of the error distributions of AT give the best results, *i.e.*, with a mean of 0.015 in Advogato and 0.016 in PGP, and a smaller variance in both datasets.

In the second group of experiments, we evaluate the performance of AT, ET and TR, in terms of ranking users based on their trustworthiness. We first randomly select a seed node u, and find all its 1-hop neighbors, denoted as V. Then, we rank the nodes in V based on u's direct opinions on these nodes, *i.e.*, nodes with higher trust values are ranked in higher positions than those with lower trust values. We take this ranking as the ground truth.

For each node $v \in V$, we remove edge (u, v) from the datasets if there exist paths from u to v. We run the AT, ET and TR algorithms to compute the trustworthiness of node v, from the perspective of u. Then, we rank the nodes in V based on the expected beliefs of ω_{uv}'s for all possible v's. We compare the ranking results obtained by the three algorithms to the ground truth. Here, ranking errors are measured by Kendall's tau ranking correlation coefficients between the computed ranking results and the ground truth. We repeat each experiment 100 times in Advogato and PGP to get statistically significant results.

In Figs. 11(a) and 11(b), AT gives more accurate ranking results, compared to other algorithms. In Advogato, the Kendall's tau correlation coefficients of AT are always greater than 0. Nearly 20% of the ranking results are exactly the same (with a coefficient of 1) as the ground truth. In PGP, AT generates > 0.1 Kendall's tau ranking correlation coefficients, and about 40% of the ranking results are the same as the ground truth. On the other hand, for ET and TR algorithms, only 20% (Advogato) and 10% (PGP) of their rankings are moderately correct, with coefficients > 0.5. In other words, ET and TR do not work well in ranking users in an OSN, based on their trustworthiness.

6 Related Work

How to model the trust between users in OSNs has attracted much attention in recent years. Existing trust models can be categorized into four groups: topology based, PageRank based, probability based, and subjective logic based models. In this section, we briefly introduce these works.

6.1 Topology based Models

Trust between users in OSNs was first studied by analyzing the characteristics of the network topology between the users. Topology based trust models treat a social network as a graph, where the edge between users represents the trust relation between them. The advantage of these models is that they leverage random walk to evaluate users' trust, and thus can easily be applied in large-scale social networks. By analyzing network topologies, the works in [18, 19, 9, 8] are able to identify untrustworthy nodes in an OSN. The basic idea is to distinguish untrustworthy regions from trustworthy regions in a social network. Specifically, the random walk algorithm is applied, starting from a trustor and searching the network to compute the probability that a trustee will be visited. A low probability indicates the trustee is in the untrustworthy region, and thus untrustworthy. Later on, the trust relation between two users is treated as a probabilistic value in [47]. Then, indirect trust inference becomes a network reachability problem. For example, in [48], a social network is considered a resistor network where the resistance of each edge is derived from the trust of the two users being connected by the edge. In [49, 50], a depth first search algorithm is employed to compute the trust between two users.

6.2 PageRank based Models

Instead of analyzing the entire structure of a social network, PageRank based solutions are inspired by the assumption that trustworthy users are likely to have more connections from other users. PageRank based trust models employ the PageRank algorithm [51] to compute the relative trust of users. For example, the EigenTrust algorithm[22] searches a social network based on the following rule: it moves from a user to another with probability proportional to how the user trusts the other, *i.e.*, higher the trust, higher the probability. In this way, EigenTrust algorithm is more likely to reach more trustworthy users. Similarly, the TrustRank algorithm [23] also employs the PageRank algorithm to rank users, based on their relative trust values. Both EigenTrust and TrustRank extend the PageRank algorithm that was commonly used to determine the importance scores of web pages. The assumption that PageRank algorithm depends on, however, may not be realistic in a social network environment, leading to inaccurate trust assessment results.

6.3 Probability based Models

Unlike the aforementioned models that treat trust as either binary or real numbers, the probability based model considers trust as a probability, *i.e.*, the likelihood that a trustee is trustworthy. Probability based trust models usually represent trust as a probability distribution. In these models, a trustor uses its interactions with a trustee to construct a probabilistic distribution to estimate the trustee's trust. The benefit of these models is that trust is accurately modeled by a rich set of statistical and probability techniques, including Hidden Markov chain and maximum likelihood estimation. In this category, trust can be represented by several different probability distributions [52, 53, 25, 26, 27]. The trust assessment becomes the problem of likelihood estimation, regarding to the corresponding distribution's parameters, based on observed evidence. For example, trust was first modeled as a binomial distribution in [28], and the likelihood estimation was carried out, based on Beta distribution. Then, trust is considered a continuous random variable [52, 53], and Gaussian distribution is used to model trust. Binomial distribution is further extended to a multinomial distribution, to handle the cases where trust is a discrete random variable [54]. Based on multinomial distribution, Bayesian inference [53, 52] and Hidden Markov Model (HMM) [25, 26, 27] can be applied to realize trust assessments.

While the former integrates evidence from various sources, *e.g.*, reputation and preference similarity of users, the latter focuses on trust modeling in a dynamic environment.

6.4 Subjective Logic based Models

The 3VSL model extends the subjective logic (SL) trust models [28, 55, 56] by redefining the uncertainty in trust, and thus achieves more accurate trust assessments. Therefore, it is worth studying the fundamental principles of SL, and the limitations when SL is applied in trust assessment in OSNs. Considering trust as a binary event, *i.e.*, a trustee is trustworthy or untrustworthy, SL assumes the probability of a trustee being trustworthy follows Beta distribution. The Beta distribution can be formed based on the amounts of positive and negative evidence, collected by a trustor about the trustee. The advantage of SL based models is that trust is more accurately modeled and the uncertainty in trust is considered. In [57, 58, 59, 60, 41, 42], the SL model is further refined and better trust assessment performance is achieved.

Subjective logic based models treat trust as an opinion and introduce a set of opinion operations, *e.g.*, discounting and consensus operations to account for trust propagation and trust fusion, respectively. The consensus operation provides a mechanism to combine possibly-conflicting opinions to generate a consensus opinion [61]. On the other hand, the discounting operation is used to help a trustor to derive indirect trust of the trustee, based on other users' recommendations [62]. For instance, if Alice trusts Bob, and Bob trusts Claire, then Alice will have an indirect opinion on Claire's trust. With the discounting and consensus operations, the trust between any two uses in an OSN can be computed. In addition to the basic discounting and consensus operations, multiplication, co-multiplication, division, and co-division of opinions are also defined in the SL models [63].

Later on, the SL model is extended to support conditional inference [64]. A conditional inference is represented in the form of "IF x THEN y" where x denotes the antecedent and y the consequent proposition. The antecedent x is modeled by the SL model, so it is not a binary value; instead, it is a vector, representing the probability that this antecedent is true. Overall, SL was proven to be compatible with binary logic, probability calculus, and classical probabilistic logic [65]. The properties are also inherited in the proposed 3VSL model, which makes 3VSL computationally effective for trust assessments in OSNs.

7 Conclusions

In this paper, the three-valued subjective logic is proposed to model and compute trust between any two users connected within OSNs. 3VSL introduces the uncertainty space to store evidence distorted from certain spaces as trust propagates through a social network, and keeps track of evidence as multiple trusts combine. We discover that there are differences between distorting and original opinions, *i.e.*, distorting opinions are so unique that they can be reused in trust computation while original opinions are not. This property enables 3VSL to handle complex topologies, which is not feasible in the subjective logic model.

Based on 3VSL, we design the AT algorithm to compute the trust between any pair of users in a given OSN. By recursively decomposing an arbitrary topology into a parsing tree, we prove AT is able to compute the tree and get the correct results.

We validate 3VSL both in experimental evaluations. The evaluation results indicate that 3VSL is accurate in modeling computing trust within complex OSNs. We further compare the AT algorithm to other benchmark trust assessment algorithms. Experiments in two real-world OSNs show that AT is a better algorithm in both absolute trust computation and relative trust ranking.

References

[1] Anirban Basu, Jaideep Vaidya, Juan Camilo Corena, Shinsaku Kiyomoto, Stephen Marsh, Guibing Guo, Jie Zhang, and Yutaka Miyake. Opinions of people: Factoring in privacy and trust. *SIGAPP Appl. Comput. Rev.*, 14(3):7–21, September 2014.

[2] Dapeng Wu, Shushan Si, Shaoen Wu, and Ruyan Wang. Dynamic trust relationships aware data privacy protection in mobile crowd-sensing. *IEEE Internet of Things Journal*, 2017.

[3] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[4] De-Nian Yang, Hui-Ju Hung, Wang-Chien Lee, and Wei Chen. Maximizing acceptance probability for active friending in online social networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 713–721, New York, NY, USA, 2013. ACM.

[5] Dapeng Wu, Junjie Yan, Honggang Wang, Dalei Wu, and Ruyan Wang. Social attribute aware incentive mechanism for device-to-device video distribution. *IEEE Transactions on Multimedia*, 19(8):1908–1920, 2017.

[6] Tiffany Hyun-Jin Kim, Payas Gupta, Jun Han, Emmanuel Owusu, Jason Hong, Adrian Perrig, and Debin Gao. Oto: Online trust oracle for user-centric trust establishment. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 391–403, New York, NY, USA, 2012. ACM.

[7] Lu Shi, Shucheng Yu, Wenjing Lou, and Y.T. Hou. SybilShield: An agent-aided social network-based sybil defense among multiple communities. In *INFOCOM, 2013 Proceedings IEEE*, pages 1034–1042, 2013.

[8] Haifeng Yu, P.B. Gibbons, M. Kaminsky, and Feng Xiao. SybilLimit: A near-optimal social network defense against sybil attacks. *Networking, IEEE/ACM Transactions on*, 18(3):885–898, June 2010.

[9] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham D. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *IEEE/ACM Trans. Netw.*, 16(3):576–589, June 2008.

[10] Denise M Rousseau, Sim B Sitkin, Ronald S Burt, and Colin Camerer. Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3):393–404, 1998.

[11] David Gefen, Elena Karahanna, and Detmar W. Straub. Trust and tam in online shopping: An integrated model. *MIS Q.*, 27(1):51–90, March 2003.

[12] Diego Gambetta. *Trust: Making and Breaking Cooperative Relations*, volume 52. Blackwell, 1988.

[13] D Harrison McKnight, Vivek Choudhury, and Charles Kacmar. Developing and validating trust measures for e-commerce: An integrative typology. *Information systems research*, 13(3):334–359, 2002.

[14] Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In *Trust and deception in virtual societies*, pages 55–90. Springer, 2001.

[15] TK Ahn and Justin Esarey. A dynamic model of generalized social trust. *Journal of Theoretical Politics*, 20(2):151–180, 2008.

[16] Larue Tone Hosmer. Trust: The connecting link between organizational theory and philosophical ethics. *Academy of management Review*, 20(2):379–403, 1995.

[17] Jomi F Hübner, Emiliano Lorini, Andreas Herzig, and Laurent Vercouter. From cognitive trust theories to computational trust. In *Proceedings of the 12th International Workshop on Trust in Agent Societies, Budapest, Hungary*, volume 10, pages 2009–11. Citeseer, 2009.

[18] Wei Wei, Fengyuan Xu, C.C. Tan, and Qun Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1951–1959, March 2012.

[19] George Danezis and Prateek Mittal. SybilInfer: Detecting sybil nodes using social networks. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA, 8th February - 11th February 2009*, 2009.

[20] Jennifer Golbeck and James Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96. Citeseer, 2006.

[21] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *Proceedings of the National Conference on artificial Intelligence*, volume 20, page 121, 2005.

[22] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[23] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

[24] Reid Andersen, Fan Chung, and Kevin Lang. Local partitioning for directed graphs using pagerank. In Anthony Bonato and FanR.K. Chung, editors, *Algorithms and Models for the Web-Graph*, volume 4863 of *Lecture Notes in Computer Science*, pages 166–178. Springer Berlin Heidelberg, 2007.

[25] Ehab ElSalamouny, Vladimiro Sassone, and Mogens Nielsen. HMM-based trust model. In *Formal Aspects in Security and Trust*, pages 21–35. Springer, 2010.

[26] Xin Liu and Anwitaman Datta. Modeling context aware dynamic trust using hidden markov model. In *AAAI*, 2012.

[27] George Vogiatzis, Ian MacGillivray, and Maria Chli. A probabilistic model for trust and reputation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 225–232. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[28] AUDUN JØSANG. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 09(03):279–311, 2001.

[29] Guangchi Liu, Qing Yang, Honggang Wang, Shaoen Wu, and M. P. Wittie. Uncovering the mystery of trust in an online social network. In *2015 IEEE Conference on Communications and Network Security (CNS)*, pages 488–496, Sept 2015.

[30] Wenjun Jiang, Jie Wu, Guojun Wang, and Huanyang Zheng. Fluidrating: A time-evolving rating scheme in trust-based recommendation systems using fluid dynamics. In *INFOCOM, 2014 Proceedings IEEE*, pages 1707–1715, April 2014.

[31] Ugur Kuter and Jennifer Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1377–1382. AAAI Press, 2007.

[32] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[33] Audun Jøsang, Ross Hayward, and Simon Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 85–94, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[34] Stephen Tu. The dirichlet-multinomial and dirichlet-categorical models for bayesian inference. *Computer Science Division, UC Berkeley, Tech. Rep.[Online]. Available: http://www. cs. berkeley. edu/ stephentu/writeups/dirichlet-conjugate-prior. pdf*, 2014.

[35] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM.

[36] Christian Borgs, Jennifer Chayes, Adam Tauman Kalai, Azarakhsh Malekian, and Moshe Tennenholtz. *A Novel Approach to Propagating Distrust*, pages 87–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[37] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4):337–358, 2005.

[38] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM.

[39] Yonghong Wang and Munindar P. Singh. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1551–1556, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[40] Andreas Jakoby, Maciej Liskiewicz, and Rüdiger Reischuk. Space efficient algorithms for series-parallel graphs. In *STACS 2001*, pages 339–352. Springer, 2001.

[41] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1025–1032, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[42] Chung-Wei Hang and Munindar P Singh. Trust-based recommendation based on graph similarity. In *Proceedings of the 13th International Workshop on Trust in Agent Societies (TRUST). Toronto, Canada*, 2010.

[43] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1025–1032, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[44] f1 score. `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score`.

[45] Daniel A Powers and Yu Xie. *Statistical methods for categorical data analysis*. Emerald Group Publishing, 2008.

[46] Jennifer Ann Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, USA, 2005. AAI3178583.

[47] T. DuBois, J. Golbeck, and A. Srinivasan. Rigorous probabilistic trust-inference with applications to clustering. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 655–658, Sept 2009.

[48] Yanjun Zuo, Wen-chen Hu, and Timothy O'Keefe. Trust computing for social networking. In *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 1534–1539. IEEE, 2009.

[49] Jennifer Ann Golbeck. Computing and applying trust in web-based social networks. 2005.

[50] Yu Zhang, Huajun Chen, and Zhaohui Wu. A social network-based trust model for the semantic web. In LaurenceT. Yang, Hai Jin, Jianhua Ma, and Theo Ungerer, editors, *Autonomic and Trusted Computing*, volume 4158 of *Lecture Notes in Computer Science*, pages 183–192. Springer Berlin Heidelberg, 2006.

[51] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[52] Zoran Despotovic and Karl Aberer. Probabilistic prediction of peers' performance in {P2P} networks. *Engineering Applications of Artificial Intelligence*, 18(7):771 – 780, 2005.

[53] WT Teacy, Michael Luck, Alex Rogers, and Nicholas R Jennings. An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artificial Intelligence*, 193(0):149 – 185, 2012.

[54] Carol J Fung, Jie Zhang, Issam Aib, and Raouf Boutaba. Dirichlet-based trust management for effective collaborative intrusion detection networks. *Network and Service Management, IEEE Transactions on*, 8(2):79–91, 2011.

[55] Audun Jøsang and Simon Pope. Semantic constraints for trust transitivity. In *Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43*, APCCM '05, pages 59–68, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[56] A. Josang and T. Bhuiyan. Optimal trust network analysis with subjective logic. In *Emerging Security Information, Systems and Technologies, 2008. SECURWARE '08. Second International Conference on*, pages 179–184, Aug 2008.

[57] Yonghong Wang and Munindar P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1425–1430. AAAI Press, 2006.

[58] Yonghong Wang, Chung-Wei Hang, and Munindar P. Singh. A probabilistic approach for maintaining trust based on evidence. *J. Artif. Int. Res.*, 40(1):221–267, January 2011.

[59] Yonghong Wang and Munindar P. Singh. Evidence-based trust: A mathematical model geared for multiagent systems. *ACM Trans. Auton. Adapt. Syst.*, 5(4):14:1–14:28, November 2010.

[60] Ugur Kuter and Jennifer Golbeck. Using probabilistic confidence models for trust inference in web-based social networks. *ACM Trans. Internet Technol.*, 10(2):8:1–8:23, June 2010.

[61] Audun Jøsang. The consensus operator for combining beliefs. *Artificial Intelligence*, 141(1):157–170, 2002.

[62] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer, 2006.

[63] Audun Jøsang and David McAnally. Multiplication and comultiplication of beliefs. *International Journal of Approximate Reasoning*, 38(1):19–51, 2005.

[64] Audun Josang. Conditional reasoning with subjective logic. *Journal of Multiple-Valued Logic and Soft Computing*, 15(1):5–38, 2008.

[65] Audun Jøsang. Probabilistic logic under uncertainty. In *Proceedings of the thirteenth Australasian symposium on Theory of computing-Volume 65*, pages 101–110. Australian Computer Society, Inc., 2007.