

TRUST ASSESSMENT IN ONLINE SOCIAL NETWORKS

by

Guangchi Liu

A dissertation submitted in partial fulfillment  
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY  
Bozeman, Montana

April, 2017

©COPYRIGHT

by

Guangchi Liu

2017

All Rights Reserved

## TABLE OF CONTENTS

1. INTRODUCTION .....	1
Problem Statements.....	2
Limitations of Prior Art .....	3
Proposed Approaches.....	5
Key Contributions .....	7
2. RELATED WORK .....	9
Definitions of Trust.....	9
Trust Models in OSNs.....	10
Topology based Trust Model .....	10
PageRank based Trust Model.....	11
Probability based Trust Model .....	11
Subjective Logic based Trust Model .....	12
Applications of Trust in Online Systems.....	13
Trust in Cloud Computing .....	14
Trust in P2P Network and Semantic Web .....	14
Trust in Cyber-Physical Systems .....	14
Trust in Spam Detection and Sybil Defense.....	15
Trust in Recommendation and Crowdsourcing Systems.....	15
3. THREE-VALUED SUBJECTIVE LOGIC .....	17
Preliminaries .....	17
A Probabilistic Interpretation of Trust .....	18
Opinion.....	21
Discounting Operation .....	22
Combining Operation.....	30
Expected Belief of An Opinion .....	35
4. THE ASSESSTRUST ALGORITHM .....	39
Properties of Different Opinions.....	40
Arbitrary Network Topology .....	45
Differences between 3VSL and SL.....	47
AssessTrust Algorithm .....	50
Illustration of the AssessTrust Algorithm .....	50
Time Complexity Analysis .....	53

## TABLE OF CONTENTS — CONTINUED

5. MASSIVE TRUST ASSESSMENT IN OSNS.....	54
Design of OpinionWalk .....	55
Opinion Matrix .....	56
Individual Opinion Vector.....	57
Opinion Walk Operation.....	57
OpinionWalk Algorithm.....	59
Illustration of the OpinionWalk Algorithm .....	61
Correctness of OpinionWalk .....	66
Series Network Topology .....	67
Parallel Network Topology .....	68
Arbitrary Topology .....	70
Time Complexity Analysis .....	73
Differences between AssessTrust and OpinionWalk Algorithms .....	74
Distributed OpinionWalk Algorithm .....	75
6. EVALUATIONS .....	79
Numerical Analysis .....	79
Discounting Operation.....	79
Combining Operation .....	81
Bridge Topology .....	82
Survey Experiments .....	84
Setup of the Survey Experiments.....	84
Errors in Discounting and Combining Operations .....	86
Experimental Evaluations.....	87
Dataset.....	88
Dataset Preparation .....	89
Accuracy of 3VSL Model .....	91
Performance of the AssessTrust Algorithm .....	94
Performance of the OpinionWalk Algorithm .....	101
7. CONCLUSION .....	104
REFERENCES CITED.....	105

## LIST OF TABLES

Table	Page
6.1 Statistics of the Advogato and PGP datasets.....	89
6.2 Selected parameters (base trust level, total evidence value) for AT, SL* and TT.....	94

## LIST OF FIGURES

Figure	Page
3.1 Examples of series topologies .....	23
3.2 Examples of parallel topologies .....	30
3.3 Combining opinions with high and low uncertainties .....	36
4.1 Difference between distorting and original opinions .....	40
4.2 Illustration of an arbitrary network topology .....	46
4.3 Difference between 3VSL and SL on the discounting operation.....	48
4.4 Difference between 3VSL and SL on the combining operation .....	49
4.5 An illustration of 3VSL based on the bridge topology.....	51
5.1 A detailed illustration of OpinionWalk .....	58
5.2 A general view of the “opinion walk” operation.....	60
5.3 Illustration of how OpinionWalk processes the bridge topology .....	63
5.4 Illustration of two fundamental topologies in an OSN .....	67
5.5 Illustration of a network with an arbitrary topology .....	71
5.6 A general view of the D-OpinionWalk algorithm .....	77
6.1 Influence of belief on discounting operation.....	80
6.2 Influence of belief and uncertainty on the discounting operation.....	81
6.3 Influence of total evidence value on combining operation .....	82
6.4 Influence of positive evidence value on combining operation .....	82
6.5 Influence of bridge opinion’s positive/total evidence ratio .....	83
6.6 Influence of bridge opinion’s total evidence value.....	83
6.7 Absolute errors in expected belief of the discounting operation.....	86
6.8 Absolute errors in expected belief of the combining operation .....	87
6.9 CDFs of errors in expected belief of the discounting operation .....	87
6.10 CDFs of errors in expected belief of the combining operation .....	88

## LIST OF FIGURES — CONTINUED

Figure	Page
6.11 F1 scores of 3VSL and SL using the Advogato dataset .....	91
6.12 F1 scores of 3VSL and SL using the PGP dataset .....	92
6.13 CDFs of $\alpha + \beta$ in opinions computed by 3VSL and subjective logic using the Advogato dataset.....	93
6.14 F1 scores of the trust assessment results generated by TT, SL* and AT using the Advogato dataset.....	94
6.15 F1 scores of the trust assessment results generated by TT, SL* and AT using the PGP dataset .....	95
6.16 Histogram of the errors generated by TT, SL* and AT using the Advogato dataset .....	96
6.17 Histogram of the errors generated by TT, SL* and AT using the PGP dataset .....	97
6.18 Fitted curves of the error distributions of TT, SL* and AT using the Advogato dataset .....	98
6.19 Fitted curves of the error distributions of TT, SL* and AT using the PGP dataset .....	98
6.20 The CDFs of Kendall's tau ranking correlation coefficients of different algorithms using the Advogato dataset.....	99
6.21 The CDFs of Kendall's tau ranking correlation coefficients of different algorithms using the PGP dataset.....	100
6.22 Execution times of different algorithms (OW, MT, ET, AT, and TT) using the Advogato dataset.....	101
6.23 Execution times of different algorithms (OW, MT, ET, AT, and TT) using the PGP dataset.....	102

## LIST OF ALGORITHMS

Algorithm	Page
4.1 AssessTrust( $G, A, C, H$ ) .....	51
5.2 OpinionWalk( $G, i, H$ ) .....	60
5.3 D-OpinionWalk Algorithm that is executed on user $j$ .....	76

## 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.

AT is able to accurately conduct one-to-one trustworthiness, however, it is inefficient in addressing the massive trust assessment (MTA) problem, i.e., computing one-to-many trustworthiness in OSNs. MTA plays a vital role in OSNs, e.g., identifying trustworthy opinions in a crowdsourcing system. If the AssessTrust algorithm is applied directly to solve the MTA problem, its time complexity is exponential. To efficiently address MTA, we propose the OpinionWalk algorithm that yields an polynomial-time complexity. OpinionWalk uses a matrix to represent a social network's topology and a vector to store the trustworthiness of all users in the network. The vector is iteratively updated when the algorithm "walks" through the entire network.

To validate the 3VSL model, we first conduct a numerical analysis. An online survey system is then implemented to validate the correctness and accuracy of 3VSL in the real world. Finally, 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. To evaluate the performance of the AssessTrust and OpinionWalk algorithms, we use the same datasets. Compared to the state-of-art solutions, e.g., EigenTrust and MoleTrust, OpinionWalk yields the same order of time complexity and a higher accuracy in trust assessment.

## 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 [9, 108]. In Twitter, spam undermines the trust among users by distributing false links [101], and thus seriously impacts the user experience. In online marketing applications [81], trust is used to identify trustworthy sellers. In a proactive friendship construction system [98], trust enables the discovery of potential friendships. In the networking security domain, trust is considered an important metric to detect malicious users [60, 85, 99, 100]. In social influence analysis, trust is a key factor in evaluating the impacts of influential users [65, 105]. Given the above-mentioned applications, one confounding issue is to what degree can a user trust another user in an OSN. This dissertation studies the fundamental issue of trust assessment in OSNs: *given an OSN, how to model and compute trust among users?*

Trust is traditionally defined as either a rating-based reputation or the probability that a user is benign. In an online marketing system, e.g. Ebay, users rate each other based on their previous interactions, so the trust of a given user is derived from aggregated ratings. In the network security domain, however, 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 [23, 26, 73, 84], we define trust as *the probability*

*that a trustee will 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 the trustworthiness of the trustee. 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 [4, 21, 41, 43] formed in the absence of interaction experiences, is not considered in the dissertation.

### Problem Statements

This dissertation aims at addressing the fundamental issue of accurately modeling and computing trust in OSNs, which requires us to solve the following three technical problems.

- **P1:** How to model direct trust in online social networks?
- **P2:** How to compute indirect trust in online social networks?
- **P3:** How to conduct massive trust assessment in online social networks?

In the first problem, because the trustor and trustee have direct interactions between each other, we call the trust relation between them *direct trust*. Based on the assumption that trust is determined by objective evidences, this problem can be formulated as follows.

**P1:** *Given the interactions between a trustor and trustee, how does one model the trustworthiness of the trustee, from the perspective of the trustor?*

Solving the second problem will provide a method to calculate the trust between two users who have no previous interactions. As the two users did not interact with each other previously, their trust relation is called *indirect trust*. Here, we model a trust 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 the trust relation from  $u$  to  $v$ . The weight of an edge  $w(u, v)$  denotes how much  $u$  trusts  $v$ , which is usually referred to as the direct trust from  $u$  to  $v$ . As such, the second problem is formulated as follows.

**P2:** *Given a trust 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 trustworthiness on  $v$ , i.e., how should  $u$  trust a stranger  $v$ ?*

Massive trust assessment (MTA) allows a user to compute the direct/indirect trustworthiness of all other users in an OSN. MTA is important in many applications. For example, the LendingClub, Inc. [2] leverages the trust relations among users in Facebook.com [3] to improve its online peer-to-peer lending service. It offers a mechanism to evaluate the trustworthiness of all potential borrowers, from the perspective of a lender. Clearly, the hinge of this application is to efficiently and accurately compute the trustworthiness of all trustees, from the point of view of the trustor. Therefore, the third problem can be formulated as follows.

**P3:** *Given a trust social network  $G = (V, E)$ ,  $\forall i$  and  $j$ , s.t.  $i, j \in V$ ,  $\exists$  at least one path from  $i$  to  $j$ , how does one efficiently compute the trustworthiness of users  $\{j \in V, j \neq i\}$ , from the perspective of user  $i$ ?*

### Limitations of Prior Art

Existing trust models can be categorized as topology (or graph) based models [13, 29, 69, 95, 99, 100], PageRank based models [7, 36, 58], probability based models [20, 68, 90], and subjective logic based models [57]. None of them, however, are able to accurately model and compute trust in OSNs.

Topology based models [13, 95, 99, 100] 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 as trustworthy to each other. The key

limitation of these models is that the trustworthiness of users within a community is indistinguishable [67], which limits the application of the models. Graph based models [29, 47, 62, 69] assign a real number, ranging from 0 to 1, on each edge in the trust social network, and employ graph searching algorithms to evaluate the trustworthiness between any two users. The major limitation of these models is that trust is represented as a single value, which omits the uncertainty existing in trust and thus is inaccurate in assessing trust.

In addition to traditional graph searching algorithms, PageRank based models, e.g., TrustRank and EigenTrust [36, 58, 59], apply the idea of PageRank to rank users based on their trustworthiness. The trustworthiness of users is obtained by calculating how likely a user can be reached from the trustor within the network. In these models, the probability of reaching a user (from the trustor) is determined by the trust value of the edge connecting to the user. The key limitation of these models is that they mistakenly treat trust propagation in a social network as a random walk process, which is not correct.

Probability based models [20, 68, 90] model trust as a probability distribution, i.e., a trustor uses previous interactions with a trustee to construct a probabilistic model to approximate the trustee’s future behavior. The major limitation of these models is that they only focus on direct trust and cannot be applied to compute indirect trust. Although the subjective logic [53, 57] model takes advantage of both graph and probability based models, it can only handle simple network topology. Its performance degrades drastically in a complex network topology that is common in real-world online social networks.

### Proposed Approaches

To address problem **P1**, we propose the three-valued subjective logic (3VSL) model that is able to accurately model trust based on users' interactions within an OSN. 3VSL is based on the subjective logic (SL) model [57]. However, it is significantly different from SL. Instead of defining trust as a binary value in SL, 3VSL treats it as a ternary value (i.e., belief, distrust, and uncertainty). In other words, a user in an OSN could be trustworthy, not trustworthy, or uncertain. Therefore, the probability of a user being trustworthy can be modeled by the *Dirichlet-Categorical* (DC) distribution that is characterized by three parameters  $\alpha$ ,  $\beta$  and  $\gamma$ . Here,  $\alpha$  represents the number of positive interactions/evidence that supports the user is trustworthy. For example, we observed that the user behaved as expected  $\alpha$  times in the past.  $\beta$  denotes the amount of negative evidence indicating the user is not trustworthy.  $\gamma$  is the amount of neutral evidence that neither supports nor opposes the user is trustworthy. The reason of introducing the uncertain state in 3VSL is that it can accurately model the trust propagation process in an OSN. During trust propagation, certain evidence measured by  $\alpha+\beta$  is “distorted” and becomes uncertain evidence, measured by  $\gamma$ . Distorted evidence is usual in trust assessment, however, they are totally omitted in SL.

To address problem **P2**, we propose a trust computation algorithm, called AssessTrust (AT), based on 3VSL model. AT decomposes the sub-graph between the trustor and trustee as a parsing tree, which provides the correct order of applying trust propagation and fusion to compute the indirect trust between the trustor and trustee. Here, trust propagation and fusion are modeled by two basic operations: discounting and combining operations. Leveraging the properties of 3VSL, AT is proven to be able to accurately compute the trustworthiness between any two users

connected within an OSN. Because 3VSL uses a probability distribution to describe whether a user is trustworthy, AT offers more accurate trust assessment, compared to the topology and graph based solutions. On the other hand, while AT makes use of the social connections between the trustor and trustee to compute their trust, it outperforms the probability based models that are only applicable for direct trust. Experiment results indicate that AT achieves the best accuracy of trust assessment in OSNs. Specifically, AT achieves the F1 scores of 0.7 and 0.75, in trust assessment, using the Advogato and Pretty Good Privacy (PGP) datasets, respectively. AT can also be used to rank users, based on their trustworthiness. We measure the accuracy of the ranking results using the Kendall’s tau coefficients, compared to the ground truth ranking. Experiment results show that AT offers 0.73 and 0.77 kendall’s tau coefficients on average in Advogato and PGP, respectively.

Although AT is able to conduct accurate trust assessment between any two users in an arbitrary social network, it is too slow to solve the problem **P3**. If AT is applied to solve the MTA problem in OSNs, it needs to be executed  $O(n)$  times, if the network contains  $n$  users. That will yield an  $O(n^{k+1})$  time complexity where  $k$  is the network’s diameter that is usually a function of  $n$ . Therefore, it is critical to design an algorithm to efficiently compute the trustworthiness of all users in the network, for any given user. Based on the 3VSL and AT algorithm, we propose a polynomial-time algorithm, called OpinionWalk, to efficiently address the MTA problem. In OpinionWalk, we use an *opinion matrix* to represent a social network’s topology. Elements in the *opinion matrix* are opinions that indicate the direct trust between users in an OSN. We design a set of matrix operations, called *opinion walk*, to capture the trust propagation and fusion with the network. Traditional multiplication and summation operations are replaced by the discounting and combining operations defined in 3VSL [66]. We prove the correctness of OpinionWalk and analyze its time complexity. We find

that OpinionWalk perfectly implements the 3VSL model and offers a better time complexity,  $O(n^3)$ , in addressing the MTA problem. Experiment results using the Advogato and PGP datasets validate the correctness of OpinionWalk.

### Key Contributions

In this dissertation, we make the following key contributions. First, we propose 3VSL to model the direct and/or indirect trust between two users connected within an OSN. 3VSL differs from prior trust models in that it considers both trust relations and network topologies, and thus is applicable in large-scale OSNs. Second, 3VSL extends SL by introducing a neutral state, distinguishing distorting opinions from original opinions, and redesigning the discounting and combining operations. Third, based on 3VSL, we propose a trust assessment algorithm AT to accurately compute the trust between any two users in an OSN. Fourth, we propose another algorithm, called OpinionWalk, to address the massive trust assessment problem. Fifth, the correctness of OpinionWalk is proven and its time complexity is analyzed. Sixth, to validate the 3VSL model and associated algorithms, we conduct intensive experiments including numerical analysis, online surveys and validation against two real-world datasets, Advogato and PGP [72].

The rest of this dissertation is organized as follows. In chapter 2, the related work is introduced. In chapter 3, we introduce the 3VSL model and define the trust propagation and fusion operations. In chapter 4, we differentiate discounting opinions from original opinions and prove that 3VSL can handle arbitrary network topologies. Based on the model, we further propose the AssessTrust algorithm. In chapter 5, we introduce the OpinionWalk algorithm and prove its correctness and analyze its time complexity. In chapter 6, we validate 3VSL through numerical and experimental evaluations. Furthermore, we evaluate the performance of AT and OpinionWalk using

two real-world datasets. In chapter 7, we conclude the dissertation and present a plan for future work.

## RELATED WORK

### Definitions of Trust

Trust has been widely studied in psychology, sociology and management domains. A widely accepted definition of trust was summarized by Rousseau in [84], based on a cross-disciplinary literature review: “Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviors of another.” Despite the various definitions of trust [23, 26, 73], they are similar to Rousseau’s, i.e., it can be concluded that trust is composed of two parts: expectation and vulnerability. While the former indicates the probability that the trustee will behave as expected, the latter shows the trustor’s willingness of relying on the trustee. Specifically, the word vulnerability emphasizes the trustor’s concerns about the uncertainty [17, 76] of the trustee’s future behaviors. The definition of trust in this dissertation is inspired by the above studies, and we define trust as the probability that the trustee will behave as expected, from the perspective of the trustor.

Although trust is commonly confused with reputation, they are two different concepts. Previous works [17, 24, 46] have identified the positive correlations between reputation and trust. However, reputation is not equivalent to trust. According to the definition from Merriam-Webster dictionary and Wikipedia, reputation is the common opinion that people have about someone or something, i.e., the overall quality or character as seen or judged by people in general. In essence, reputation comes from the public and general opinion. However, trust comes from individual opinions, i.e., from a trustor to a trustee with emphasis on personal interactions. On the other hand, reputation is a summary of past events while trust is the intention and expectation of the future.

### Trust Models in OSNs

Trust is built on the social ties between users and how to model trust in online social networks has attracted more attention in recent OSN studies. Several works exist regarding to modeling trust in social networks. In this section, we briefly introduce these works.

#### Topology based Trust Model

Topology based trust models treat a trust social network as a graph, where an edge represents the trust relationship between two neighboring nodes. The advantage of these methods is that they leverage random walk to evaluate trust, and thus can be easily applied in large-scale OSNs.

By analyzing network topologies, the works in [13, 95, 99, 100] are able to identify untrustworthy nodes in an OSN. Their fundamental idea is to identify untrustworthy nodes by distinguishing untrustworthy regions from trustworthy regions in the network. Specifically, they play random walk from a trustor and evaluate the probability of reaching a trustee. A low probability indicates that the trustee is not in the trustworthy region, and vice versa. Later on, people began to model indirect trust by considering the trust values between users. In [19], a trust relation between two users is treated as a probabilistic value. All users and their associated trust relations compose a graph. Then, the indirect trust inference problem becomes a network reachability problem. In [109], a trust network is considered a resistor network where the resistance of each edge is derived from the trustworthiness of the edge. In [31, 103], given a trust network, a depth-first search algorithm is employed to compute the trust between any two users.

### PageRank based Trust Model

PageRank based trust models employ the PageRank algorithm [78] to compute the relative trustworthiness of interested users. For example, the EigenTrust algorithm, proposed in a peer-to-peer system [58], starts from a peer and searches for trustworthy peers based on several rules. It moves from peer to peer with a probability that is proportional to the other peer's trust score, i.e., higher the trust score, higher the moving probability. In this way, EigenTrust will more likely reach trustworthy peers. Later on, the relative trust of web pages is investigated in [36] to identify spam pages. The TrustRank algorithm proposed in [36] again employs the PageRank algorithm on the network to rank the trustworthiness of web pages. Both EigenTrust and TrustRank can be viewed as a variant of the PageRank algorithm that is a well known solution to assigning importance scores to pages on the Internet. These algorithms, however, only generate trust rankings, instead of absolute trust values of peers/pages.

### Probability based Trust Model

Probability based trust models treat direct trust as probability distributions, where a trustor uses past interactions and observations of a trustee to construct a probabilistic model approximating the trustee's future behavior. The advantage of these models is that trust can be accurately computed based on a wide variety of statistical and probability techniques, including Hidden Markov Chain, Maximum Likelihood Estimation, etc.

Many previous efforts were devoted to the study of modeling direct trust between OSN users in a computational way [14,20,68,88,90]. For example, direct trust is modeled as a discrete multinomial distribution in [22]. Therefore, trust assessment becomes a problem of likelihood estimation, regarding to the distribution parameters

based on given evidence. If trust is modeled as a discrete binomial distribution (i.e., a user is either trustworthy or not), the likelihood estimation can be performed on the Beta distribution [57]. If trust is modeled as a continuous random variable, Gaussian distribution can be used [14, 88] to model non-discrete cases where a possible outcome is a continuous value.

The binomial distribution can be further extended to a multinomial distribution to handle the case of multiple discrete random variables [22]. Based on the multinomial distribution (including the binomial distribution), Bayesian analysis [14, 88] and Hidden Markov Model (HMM) [20, 68, 90] can be applied in trust assessment. While the former integrates evidence from various sources, e.g., reputation scores and preference similarity, the latter handles the dynamic in trust.

### Subjective Logic based Trust Model

To understand trust in online social networks, Jøsang proposed the subjective logic model in [52, 56, 57]. Considering a binary trust value, subjective logic assumes the probability of a user being trustworthy follows the Beta distribution. The Beta distribution here can be computed from the numbers of positive and negative evidence, respectively. The advantage of using subjective logic is that trust can be more realistically modeled by considering the uncertainty in a person's judgment about trust. Such uncertainty exists because it is difficult for a person to determine with absolute certainty whether a person is trustworthy or not. In [37, 38, 63, 91, 92, 94], the subjective logic model is further refined to improve its accuracy in trust assessment.

Subjective logic treats trust as opinions and introduces an algebra for opinion operations, e.g., discounting and consensus operations for trust propagation and fusion, respectively. The consensus operation provides a method for combining

possibly conflicting beliefs/opinions to generate a consensus opinion [49]. The consensus opinion reflects all opinions being combined in a fair and equal way. The discounting operation is the operation by which a new trust relationship can be derived from pre-existing trust relationships [54]. For example, if Alice trusts Bob, and Bob trusts Claire, then by trust propagation, Alice will also trust Claire. With the discounting and consensus operations, it is possible to compute the indirect trust between two connected users in OSNs.

Besides the two basic discounting and consensus operations, Jøsang further defined the multiplication, co-multiplication, division, and co-division of opinions [55]. Although these operations are irrelevant in modeling trust propagation and fusion, they allow an opinion to be multiplied or divided by another opinion. Later on, subjective logic is extended to support conditional inference [51]. A conditional inference is usually in the form of “IF  $x$  THEN  $y$ ” where  $x$  denotes the antecedent and  $y$  the consequent proposition. Here, the antecedent  $x$  is modeled by subjective logic so that it is not a binary value, true or false. Instead, it is a vector representing the probability that this antecedent is true. Overall, subjective logic was proven to be compatible with binary logic, probability calculus, and classical probabilistic logic [50].

### Applications of Trust in Online Systems

Along with the rapid development of the Internet and online services, trust has been used in many applications for either improving users’ quality of experience (QoE) or preventing the disturbance of malicious users. In this section, we briefly introduce these applications.

### Trust in Cloud Computing

Recently, trust was introduced in the concept of social cloud. In [75], Mohaisen *et al.* employ trust as a metric to identify good workers for an outsourcer through her social network. In [77], Moyano *et al.* proposed a framework to employ trust and reputation for cloud provider selection. In [79], Pietro *et al.* proposed a multi-round approach, called AntiCheetah, to dynamically assign tasks to cloud nodes, accounting for their trustworthiness.

### Trust in P2P Network and Semantic Web

Trust analysis was first implemented in peer-to-peer (P2P) networks [58,96,104]. In P2P networks, trust is used to evaluate the trustworthiness of a particular resource owner, and thereby identify malicious sources. Trust analysis was also applied to semantic webs [8, 30, 82]. The purpose of analyzing trust in semantic webs is to study the trustworthiness of data with efficient knowledge processing mechanisms. For example, the trustworthiness of web hyperlinks are studied in [36, 61, 71]. Trust analysis is then applied to filter untrustworthy contents in [10,12,15,16,18,28]. Finally, trust was used to evaluate the quality of contents on semantic webs in [27, 32, 71, 83, 86, 106].

### Trust in Cyber-Physical Systems

Trust analysis is also introduced in cyber-physical systems (CPS), e.g., wireless sensor networks and vehicular networks [40]. For example, a trust based framework is proposed to secure data aggregation in wireless sensor networks [102], which evaluates the trustworthiness of each sensor node by the Kullback-Leibler (KL) distance to identify the compromised nodes through an unsupervised learning technique. In [64], trust analysis is employed to identify malicious and selfish nodes in a mobile ad

hoc network. In addition, Xiaoyan *et al.* propose a new trust architecture, called situation-aware trust (SAT), to address several important trust issues in vehicular networks, which are essential to overcome the weaknesses of current vehicular network security and trust models [40].

### Trust in Spam Detection and Sybil Defense

Another important domain in which trust analysis is widely applied is Sybil defense and spam detection [5, 25, 42, 74, 87, 95]. The goal of these works is to identify forged multiple identities and spam information in OSNs. The basic idea of [5, 95] is to employ random walk to rank the neighbors in a given OSN from a seed node, and extract a trust community composed of high ranking nodes. Then, the users outside the trust community will be considered as not trustworthy, i.e., potential Sybil nodes. In [87], Tan *et al.* integrated traditional Sybil defense techniques with the analysis of user-link graphs. In [74], Mohaisen *et al.* proposed a derivation of the random walk algorithm, which employs biased random mechanism, to account for trust and other social ties. In [97], besides graph based features, Yang *et al.* introduced some other features to identify spammers. In addition, in [25, 42], spam detection approaches based on user similarity and content analysis are studied.

### Trust in Recommendation and Crowdsourcing Systems

In addition to Sybil defense in OSNs, trust analysis is also useful in recommendation systems [6, 9, 37, 45, 70, 108]. For example, in [108], Zou *et al.* proposed a belief propagation algorithm to identify untrustworthy recommendations generated by spam users. In [9], Basu *et al.* proposed a privacy preserving trusted social feedback scheme to help users obtain opinions from friends and experts whom they trust. In [6], Andersen *et al.* proposed a trust-based recommendation system that generates

personalized recommendations by aggregating the opinions from other users. In addition, five axioms about trust in a recommendation system are studied in [6].

## THREE-VALUED SUBJECTIVE LOGIC

In this chapter, we propose the three-valued subjective logic (3VSL) model to model the trust between users in OSNs. Designing this model is a challenging task because trust propagation in OSNs is not well understood, although it is widely used in many applications. We address this challenge by modeling trust as a probabilistic distribution over three different states, i.e., belief, distrust, and uncertainty. By looking at how the states of trust change during trust propagation, we redesign the trust discounting operation in subjective logic [57]. In 3VSL, the parameters controlling the probabilistic distribution are determined by the amount of evidence that support each state. The evidence is collected from the interactions between the trustor and trustee. To model trust fusion, we further design the combining operation. Together with the discounting operation, we are able to model and compute the trust between two users that are directly or indirectly connected to each other.

Preliminaries

To better understand 3VSL, we first briefly introduce the subjective logic [57]. Considering two users  $A$  and  $X$ ,  $A$ 's opinion about the trustworthiness of  $X$  can be described by an *opinion vector*:

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

where  $\alpha_{AX}$ ,  $\beta_{AX}$ , 2 denotes the amount of evidence that supports user  $X$  is trustworthy, not trustworthy, and uncertain, respectively. Note that the amount of uncertain evidence in an opinion in SL is always 2.  $a_{AX}$  is called base rate and formed from an existing impression without solid evidence, e.g. prejudice, preference,

or general opinion obtained from hearsay. For example, if  $A$  always distrusts/trusts the persons from a certain group where  $X$  belongs to, then  $a_{AX}$  will be smaller/greater than 0.5.

Based on the Beta distribution, two opinions  $\omega_1 = \langle \alpha_1, \beta_1, 2 \rangle | a_1$  and  $\omega_2 = \langle \alpha_2, \beta_2, 2 \rangle | a_2$  can be combined to form a new opinion  $\omega_{12} = \langle \alpha_{12}, \beta_{12}, 2 \rangle | a_{12}$ , where  $\alpha_{12}$ ,  $\beta_{12}$  and  $a_{12}$  are calculated as follows.

$$\begin{cases} \alpha_{12} = \alpha_1 + \alpha_2 \\ \beta_{12} = \beta_1 + \beta_2 \\ a_{12} = \frac{a_1 + a_2}{2} \end{cases} .$$

Let  $A$  and  $B$  denote two persons where  $\omega_1 = \langle \alpha_1, \beta_1, 2 \rangle | a_1$  is  $A$ 's opinion about  $B$ 's trustworthiness. Assume  $C$  is another person where  $\omega_2 = \langle \alpha_2, \beta_2, 2 \rangle | a_2$  is  $B$ 's opinion about  $C$ . Then, subjective logic applies the discounting operation to compute  $A$ 's opinion about  $C$ 's trustworthiness  $\omega_{AC} = \langle \alpha_{12}, \beta_{12}, 2 \rangle | a_{12}$ , where  $\alpha_{12}$ ,  $\beta_{12}$  and  $a_{12}$  is calculated as follows.

$$\begin{cases} \alpha_{12} = \frac{\alpha_1 \alpha_2}{\beta_2 + \alpha_2 + 2} \cdot \frac{2}{\kappa} \\ \beta_{12} = \frac{\alpha_1 \beta_2}{\beta_2 + \alpha_2 + 2} \cdot \frac{2}{\kappa} \\ a_{12} = a_2 \end{cases} ,$$

where

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

### A Probabilistic Interpretation of Trust

Trust in 3VSL is defined as the probability that a user will behave as expected in the future. 3VSL models a user's future behavior as a random variable  $x$  that takes

on one of three possible outcomes  $\{1, 2, 3\}$ , i.e.,  $x = 1$ ,  $x = 2$  and  $x = 3$  indicate the user will behave as expected, not as expected, or in an uncertain way, respectively. The third state, uncertain state, is introduced in 3VSL to capture the uncertainty that exists in trust assessment. 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 seeing event  $i$ . The Iverson bracket  $[x = i]$  evaluates to 1 if  $x = i$ , and 0 otherwise.

If the value tuple  $\mathbf{p}$  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 a group of random variables that follows the Dirichlet distribution:

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

where  $\alpha, \beta, \gamma$  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 can have various pdfs of  $f(\mathbf{p})$  by changing the values of  $\alpha, \beta, \gamma$ :

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

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

With the mathematical model in place, parameter  $\mathbf{p}$  can be estimated based on the observations of  $x$ , according to the Bayesian inference. Given a set of independent observations of  $x$ , denoted by  $\mathbf{D} = \{x_1, x_2, \dots, x_n\}$  where  $x_j \in \{1, 2, 3\}$  and  $j = 1, 2, \dots, n$ , 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  $x = i$ , we know  $\sum c_i = n$ . 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  $P(\mathbf{D}|\mathbf{p})$  is the likelihood function  $p_1^{c_1} p_2^{c_2} p_3^{c_3}$ , and  $f(\mathbf{p})$  the prior pdf of  $\mathbf{p}$ .  $P(\mathbf{D})$  is the probability that  $\mathbf{D}$  occurs, which is independent of  $\mathbf{p}$ . So 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 predictive model for  $x$ :

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

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 [89].

### Opinion

In the previous section, we introduce how to model trust as a DC distribution. Because the shape of a DC distribution is determined by three parameters, we can instead use these parameters to form a vector to represent trust. This vector is called an *opinion* that expresses a trustor's opinion about a trustee's trustworthiness.

For a given DC distribution, the only undetermined parameters are  $\alpha, \beta, \gamma$ . We set  $\alpha = \beta = \gamma = 1$  in default, if there is no prior knowledge about  $\mathbf{D}$ . In this case, the Dirichlet distribution becomes an uniform distribution, i.e.,  $p_1 = p_2 = p_3 = 1/3$ . Assuming  $\mathbf{p}$  initially follows the uniform distribution is reasonable because we make no observation of  $x$ , and the best choice is to believe that  $x$  could be 1, 2, 3 with equal probability. As more observations of  $x$  are made, the pdf of  $\mathbf{p}$  approaches to the true one.

From Eq. 3.2, we can predict the probability of  $x = i$  where  $i = 1, 2, 3$ , i.e., whether a user will behave as expected, not as expected, or in an uncertain way. In other words, we can use Eq. 3.2 to compute the trustworthiness of a user. From Eq. 3.2, we can obtain the expectation of the probability that a user will behave as expected:

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

where  $\Gamma(n) = (n - 1)!$  is the Gamma function. Similarly, the probabilities that the user will behave not as expected or uncertain are

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

and

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

If the hyper-parameters  $\alpha, \beta, \gamma$  equal to 1, the trustworthiness of a user is only determined by  $c_1, c_2, c_3$ , i.e., the numbers of observations that support the user will behave as expected, not as expected, and uncertain. We call these observations positive, negative, and uncertain evidence. In other words, a trustee  $X$ 's trustworthiness to trustor  $A$  can be modeled by the interaction evidence between them:

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

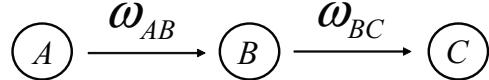
Here,  $\omega_{AX}$  denotes  $A$ 's opinion on  $X$ 's trustworthiness, and  $\alpha_{AX}, \beta_{AX}, \gamma_{AX}$  refers to the amount of observed positive, negative and uncertain evidence, based on  $A$ 's interactions with  $X$ . We further name them *belief*, *distrust* and *uncertainty* parameters in the rest of this dissertation. The subscripts of  $\alpha_{AX}, \beta_{AX}, \gamma_{AX}$  differentiate them from the prior  $\alpha, \beta, \gamma$ , i.e., the former represents observed evidence while the latter is set as 1.

### Discounting Operation

Because trust is modeled by DC distribution, in this section, we will model the trust propagation by defining the operation between two DC distributions (or opinions). Trust propagation in OSNs was intensively studied in the past decade. It



(a) A general illustration of series topology.



(b) A simple example of series topology.

Figure 3.1: Examples of series topologies

can be illustrated in a series topology, e.g., Fig. 3.1(a), where two edges are connected in *series* if they are incident to a vertex of degree 2. In Fig. 3.1(a), the nodes are users in a trust social network. The directed edges indicate the opinions between them. Trust propagation means that if user  $A_{i-1}$  trusts  $A_i$  and  $A_i$  trusts  $A_{i+1}$ , then  $A_{i-1}$  will trust  $A_{i+1}$ , even if  $A_{i-1}$  did not interact with  $A_{i+1}$  before.

Let's take the example shown in Fig. 3.1(b) to define the discounting operation in 3VSL. Based on existing research works about trust propagation [11, 34, 35, 107], 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 trustworthiness of  $C$ , then  $A$  is uncertain about  $C$ 's trustworthiness.
- A4: If  $A$  does not trust  $B$ , or  $A$  is uncertain about the trustworthiness of  $B$ ,  $A$  is uncertain about the trustworthiness of  $C$ .

It is worth mentioning that trust propagation refers to an opinion being transferred from a trustful user to another user. In other words, if  $A$  trusts  $B$ , then  $B$ 's opinion

of  $C$  will be transferred and becomes  $A$ 's opinion of  $C$ . Otherwise, if  $A$  does not trust or is uncertain about  $B$ , then  $A$  is uncertain about  $C$  as  $B$ 's opinion on  $C$  cannot be trusted. According to the 3VSL model, we model  $A$ 's opinion of  $B$  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  $B$  (on  $B$  and  $C$ ), respectively. In this way, the expected probability that  $C$  will behave as  $A$ 's expectation can be computed from

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

The intuition behind Eq. 3.4 can be explained as follows.  $A$  trusts  $C$  if and only if  $A$  trusts  $B$  and  $B$  trusts  $C$ , which is the assumption A1 we made based on the findings from [11, 34, 107]. In other words, the probability that  $C$  will behave as  $A$  expects is equal to the probability that  $C$  will behave as  $B$ 's 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  $f(x = 1 | \mathbf{p}_{BC}) f(\mathbf{p}_{BC} | \mathbf{D}_{BC}) d(\mathbf{p}_{AB})$  denotes the probability that  $B$  trusts  $C$ .

Because the two events, i.e.,  $A$  trusts  $B$  and  $B$  trusts  $C$ , are independent with each other, Eq. 3.4 can be rewritten as

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

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

$$\begin{aligned} & \int f(x = 1 | \mathbf{p}_{AB}) f(\mathbf{p}_{AB} | \mathbf{D}_{AB}) d(\mathbf{p}_{AB}) \\ = & \frac{\alpha_{AB}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \\ & \int f(x = 1 | \mathbf{p}_{BC}) f(\mathbf{p}_{BC} | \mathbf{D}_{BC}) d(\mathbf{p}_{BC}) \\ = & \frac{\alpha_{BC}}{\alpha_{BC} + \beta_{BC} + \gamma_{BC}}. \end{aligned} \quad (3.6)$$

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

$$\frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC})}. \quad (3.7)$$

According to assumption A2, the probability that  $C$  will not behave as  $A$  expects can be computed from

$$\begin{aligned} & \iint f(x = 1 | \mathbf{p}_{AB}) f(\mathbf{p}_{AB} | \mathbf{D}_{AB}) \times \\ & f(x = 2 | \mathbf{p}_{BC}) f(\mathbf{p}_{BC} | \mathbf{D}_{BC}) d(\mathbf{p}_{AB}) d(\mathbf{p}_{BC}). \end{aligned} \quad (3.8)$$

This equation makes sense because  $A$  does not trust  $C$  if and only if he trusts  $B$  and  $B$  does not trust  $C$ . Because the events that  $A$  trusts  $B$  and  $B$  does not trust  $C$  are independent, we have the expected probability that  $A$  does not trust  $C$  as

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

Finally, the expected probability that  $A$  is uncertain about  $C$ 's trustworthiness can be derived from assumptions A3 and A4:

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

The expected probability can then be computed as

$$\frac{\alpha_{AB}\gamma_{BC} + (\beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC})}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC})}. \quad (3.10)$$

Note that the summation of Eqs. 3.4, 3.8 and 3.10 equals 1.

Because Eqs. 3.7, 3.9 and 3.10 give the current estimates of probabilities that  $C$  will behave as expected, not as expected, or in a uncertain way, respectively, we could use the following categorical distribution to model  $C$ 's future behavior (from  $A$ 's perspective).

$$f(x|\mathbf{p}_{AC}) = \prod_{i=1}^3 p_i^{[x=i]}, \quad (3.11)$$

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

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

Based on our calculation, we know the categorical distribution is derived from  $B$ 's opinion on  $C$ . Let's assume that  $B$  makes a set of observations  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  on  $C$ 's behavior. According to our definitions, we know  $\alpha_{BC}, \beta_{BC}, \gamma_{BC}$  equal to the number of observations where  $x = 1, x = 2, x = 3$ , respectively. Clearly, the observations  $B$  made about  $C$  do not reflect  $A$ 's opinion on  $C$  because  $\langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle$  represents only  $B$ 's opinion on  $C$ . Here, the question is if  $A$  were asked to make the  $n$  observations, how many of them will be positive, negative, and uncertain. In other words,  $A$  needs to re-categorize  $B$ 's observations on  $C$  such that the updated evidence supports  $A$ 's current opinion on  $C$ .

For each  $x_j \in \mathbf{x}$  where  $j = 1, 2, \dots, n$ , we know  $x_j$  is observed given the underlying categorical distribution in Eq. 3.11. Therefore, we know  $\mathbf{x}$  follows the multinomial distribution with parameters  $(n, \mathbf{p}_{AC})$ . From the multinomial distribution, we can compute the probability of any means of re-categorizing the observation set  $\mathbf{x}$ . The maximum probability corresponds to the most-likely way of re-categorizing  $\mathbf{x}$ . Therefore, we know the following re-categorization occurs with the

highest probability.

$$\begin{aligned}
\alpha_{AC} &= p_1(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) \\
&= \frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\beta_{AC} &= p_2(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) \\
&= \frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
\gamma_{AC} &= p_3(\alpha_{BC} + \beta_{BC} + \gamma_{BC}), \\
&= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) + \alpha_{AB}\gamma_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}.
\end{aligned} \tag{3.13}$$

Therefore, we use  $\omega_{AC} = \langle \alpha_{AC}, \beta_{AC}, \gamma_{AC} \rangle$  to represent  $A$ 's opinion about  $C$ 's trustworthiness. It is worth mentioning that opinion  $\omega_{AC}$  is generated from distorting the positive/negative evidence in  $\omega_{BC}$  and saving them as uncertain evidence, i.e.,

$$\alpha_{AC} + \beta_{AC} + \gamma_{AC} = \alpha_{BC} + \beta_{BC} + \gamma_{BC}. \tag{3.14}$$

In other words, the total amount of evidence observed does not change during the discounting process. 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 trustworthiness, and  $\omega_{BC} = \langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle$  is  $B$ 's opinion on  $C$ 's trustworthiness; the discounting operation  $\Delta(\omega_{AB}, \omega_{BC})$  computes  $A$ 's opinion on  $C$  as*

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

where

$$\begin{aligned}
 \alpha_{AC} &= \frac{\alpha_{AB}\alpha_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
 \beta_{AC} &= \frac{\alpha_{AB}\beta_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}, \\
 \gamma_{AC} &= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{BC} + \beta_{BC} + \gamma_{BC}) + \alpha_{AB}\gamma_{BC}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}.
 \end{aligned} \tag{3.15}$$

Intuitively, the discounting operation can be understood as certain evidence from  $\omega_{BC}$  is distorted by  $\omega_{AB}$  and transferred into the uncertainty space of  $\omega_{BC}$ . Recall that the total amount of evidence of opinion  $\Delta(\omega_{AB}, \omega_{BC})$  is the same as  $\omega_{BC}$ 's, we conclude *the resulting opinion of a discounting operation shares exactly the same evidence space as the original opinion*. It is worth mentioning that the discounting operation yields two properties. The first one is called the decay property:

**Corollary 3.0.1.** *Decay Property: Given two opinions  $\omega_{AB}$  and  $\omega_{BC}$ ,  $\Delta(\omega_{AB}, \omega_{BC})$  operation yields  $\alpha_{AC} \leq \alpha_{BC}$ ,  $\beta_{AC} \leq \beta_{BC}$  and  $\gamma_{AC} > \gamma_{BC}$ .*

*Proof.* Since  $\frac{\alpha_{AB}}{(\alpha_{AB} + \beta_{AB} + \gamma_{AB})} \leq 1$ , according to Eq 3.15, we have  $\alpha_{AC} \leq \alpha_{BC}$  as well as  $\beta_{AC} \leq \beta_{BC}$ . Hence,  $-\alpha_{AC} - \beta_{AC} \geq -\beta_{AC} - \beta_{AC}$ . According to Eq. 3.14, so we have  $\gamma_{AC} \geq \gamma_{BC}$ .  $\square$

In other words, by applying a discounting operation, the uncertainty parameter of the resulting opinion increases while the belief and distrust parameters decrease. This property implies that the more trust propagates, the more uncertain the resulting opinion. The second one is called associative property:

**Corollary 3.0.2.** *Associative Property: Given three opinions  $\omega_{AB}$ ,  $\omega_{BC}$  and  $\omega_{CD}$ ,  $\Delta(\Delta(\omega_{AB}, \omega_{BC}), \omega_{CD}) \equiv \Delta(\omega_{AB}, \Delta(\omega_{BC}, \omega_{CD}))$ .*

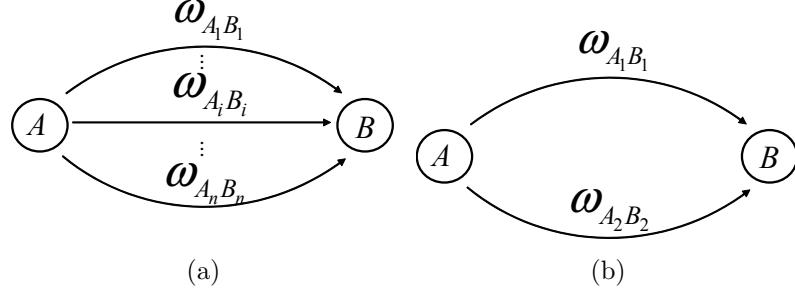


Figure 3.2: Examples of parallel topologies

*Proof.* Simply based on Eq 3.15.  $\square$

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

### Combining Operation

In this section, we will introduce the combining operation in 3VSL. According to previous works [11, 34, 107], several trust opinions can be fused into a consensus one by aggregating the opinions from different sources. Trust fusion can be illustrated by a parallel topology, e.g., Fig. 3.2(a), where two edges are connected in *parallel* if they join the same pair of distinct vertices. In Fig. 3.2(a), nodes  $A$ ,  $B$  are users in a trust social network. The edges from  $A$  to  $B$  denote  $A$ 's opinions about  $B$ 's trustworthiness that are formed from different sources.

Let's use the simplest parallel topology shown in Fig. 3.2(b) to explain the 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  $A$ 's opinions of  $B$  from two different sources. Here, 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 observations made by  $A$  on  $B$  from two different sources.

According to the definition of an opinion, the expected probability that  $B$  will behave as  $A$  expects is computed from the following DC distribution.

$$\int f(x = 1 | \mathbf{p}_{AB}) f(\mathbf{p}_{AB} | \mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}) d(\mathbf{p}_{AB}), \quad (3.16)$$

where  $\mathbf{D}_{A_1B_1}$  and  $\mathbf{D}_{A_2B_2}$  denote the aggregated observations from two sources.

The intuition of Eq. 3.16 can be explained as follows.  $A$  first infers the parameters  $\mathbf{p}_{AB}$  by aggregating the observations  $\mathbf{D}_{A_1B_1}$  and  $\mathbf{D}_{A_2B_2}$ . Therefore, the posterior pdf of  $\mathbf{p}_{AB}$  becomes

$$f(\mathbf{p}_{AB} | \mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}). \quad (3.17)$$

Based on the inferred parameters  $\mathbf{p}_{AB}$ , the probability that  $B$  will behave as  $A$  expects can be computed from  $f(x = 1 | \mathbf{p}_{AB})$ . By considering all possible values of  $\mathbf{p}_{AB}$ , we can obtain Eq. 3.16.

We now derive the analytic form of Eq. 3.16. The intuition of trust fusion can be explained as follows.  $A$  first forms his opinion on  $B$ 's trustworthiness from observation  $\mathbf{D}_{A_1B_1}$ . As such, the pdf of parameters  $\mathbf{p}_{AB}$  can be computed. Then,  $A$  adjusts its estimate about  $\mathbf{p}_{AB}$  based on a new set of evidence  $\mathbf{D}_{A_2B_2}$ . Therefore, Eq. 3.17 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 Bayes' rule,

it can be expressed as follows.

$$\begin{aligned}
& f(\mathbf{p}_{AB} | \mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}) \\
&= \frac{f(\mathbf{D}_{A_2B_2} | \mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1})}{f(\mathbf{D}_{A_2B_2})} \\
&= \frac{f(\mathbf{D}_{A_2B_2} | \mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1})}{\int f(\mathbf{D}_{A_2B_2} | \mathbf{p}_{A_1B_1}) f(\mathbf{p}_{A_1B_1}) d\mathbf{p}_{A_1B_1}}. \tag{3.18}
\end{aligned}$$

In the above equation, parameters  $\mathbf{p}_{A_1B_1}$  are derived from  $\mathbf{D}_{A_1B_1}$  that follow the Dirichlet distribution, so its pdf  $f(\mathbf{p}_{A_1B_1})$  can be computed as follows.

$$\begin{aligned}
& f(\mathbf{p}_{A_1B_1}) \\
&= \frac{\Gamma(\alpha_{A_1B_1} + \beta_{A_1B_1} + \gamma_{A_1B_1})}{\Gamma(\alpha_{A_1B_1})\Gamma(\beta_{A_1B_1})\Gamma(\gamma_{A_1B_1})} \times \\
&\quad (p_1)^{\alpha_{A_1B_1}-1} (p_2)^{\beta_{A_1B_1}-1} (p_3)^{\gamma_{A_1B_1}-1}. \tag{3.19}
\end{aligned}$$

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

$$\begin{aligned}
& f(\mathbf{D}_{A_2B_2} | \mathbf{p}_{A_1B_1}) \\
&= \frac{\Gamma(\alpha_{A_2B_2} + \beta_{A_2B_2} + \gamma_{A_2B_2} + 1)}{\Gamma(\alpha_{A_2B_2} + 1)\Gamma(\beta_{A_2B_2} + 1)\Gamma(\gamma_{A_2B_2} + 1)} \times \\
&\quad (p_1)^{\alpha_{A_2B_2}} (p_2)^{\beta_{A_2B_2}} (p_3)^{\gamma_{A_2B_2}}. \tag{3.20}
\end{aligned}$$

Substituting  $f(\mathbf{p}_{A_1B_1})$  and  $f(\mathbf{D}_{A_2B_2}|\mathbf{p}_{A_1B_1})$  in Eq. 3.18 by Eq. 3.19 and Eq. 3.20, we obtain the analytic form of Eq. 3.18.

$$\begin{aligned}
 & f(\mathbf{p}_{AB}|\mathbf{D}_{A_1B_1}, \mathbf{D}_{A_2B_2}) \\
 &= \frac{\Gamma(\alpha_{AB} + \beta_{AB} + \gamma_{AB})}{\Gamma(\alpha_{AB})\Gamma(\beta_{AB})\Gamma(\gamma_{AB})} \times \\
 & \quad (p_1)^{\alpha_{AB}-1}(p_2)^{\beta_{AB}-1}(p_3)^{\gamma_{AB}-1},
 \end{aligned} \tag{3.21}$$

where

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

Obviously, Eq. 3.21 can be considered the pdf of the following Dirichlet distribution.

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

Therefore, the following 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  $\{\alpha_{A_1B_1} + \alpha_{A_2B_2}, \beta_{A_1B_1} + \beta_{A_2B_2}, \gamma_{A_1B_1} + \gamma_{A_2B_2}\}$ . According to the definition of an opinion, the analytic form of

Eq. 3.16 can be expressed as follows.

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

The probability that  $B$  will not behave as  $A$  expects and the probability that  $B$  will behave in an uncertain way can be expressed as follows.

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

and

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

respectively. Now, we 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 opinions on two parallel paths from users  $A$  to  $B$ , the combining operation  $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2})$  is carried out as follows.*

$$\omega_{AB} = \Theta(\omega_{A_1B_1}, \omega_{A_2B_2}) = \langle \alpha_{AB}, \beta_{AB}, \gamma_{AB} \rangle, \quad (3.22)$$

where

$$\begin{cases} \alpha_{AB} = \alpha_{A_1B_1} + \alpha_{A_2B_2} \\ \beta_{AB} = \beta_{A_1B_1} + \beta_{A_2B_2} \\ \gamma_{AB} = \gamma_{A_1B_1} + \gamma_{A_2B_2} \end{cases} \quad (3.23)$$

It is worth mentioning that the combining operation yields two properties.

**Corollary 3.0.3.** *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.* Based on Eq. 3.23. □

**Corollary 3.0.4.** *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.* Based on Eq. 3.23. □

If there exist multiple parallel opinions  $\omega_{A_1B_1}, \omega_{A_2B_2} \dots \omega_{A_nB_n}$  from  $A$  to  $B$ , the overall opinion can be calculated as  $\Theta(\Theta(\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}), \dots), \omega_{A_nB_n})$ . As combining operation is commutative and associative, it is simplified to  $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}, \dots \omega_{A_nB_n})$ .

### 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 introduced in chapter 4. Many times, it is desired to represent the trust by a single number, rather than a vector composed of three numbers. Therefore, we introduce how to compute the expected belief of an opinion.

Given an opinion  $\omega_{AX} = \langle \alpha_{AX}, \beta_{AX}, \gamma_{AX} \rangle$ , it is interesting to know how likely  $X$  will perform the desired actions requested by  $A$ . We call this probability as the expected belief of  $\omega_{AX}$ . Although  $\alpha_{AX}$  denotes the belief of the opinion  $\omega_{AX}$ , other components like  $\beta_{AX}, \gamma_{AX}$  also need to be considered to compute the expected belief.

The expected belief of an opinion in the subjective logic is defined as

$$\begin{aligned} E_{\omega_{AX}} &= \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX} + 2} + \frac{1 \times a_{AX}}{\alpha_{AX} + \beta_{AX} + 2} \\ &= \frac{\alpha_{AX} + a_{AX}}{\alpha_{AX} + \beta_{AX} + 2}. \end{aligned}$$

According to this definition, the expected belief in 3VSL would become

$$E_{\omega_{AX}} = \frac{\alpha_{AX} + a_{AX}\gamma_{AX}}{\alpha_{AX} + \beta_{AX} + \gamma_{AX}}.$$

The above definition, however, is incorrect and we will illustrate the problem using an example shown in Fig. 3.3. In this figure, there exist two opinions  $\omega_1$  and  $\omega_2$ .

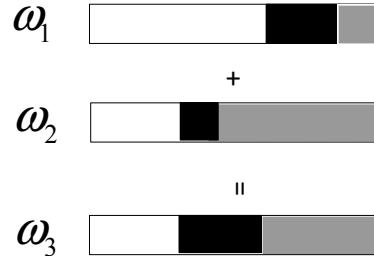


Figure 3.3: Combining opinions with high and low uncertainties

We assume the total evidence values of  $\omega_1$  and  $\omega_2$  are equal, i.e.,  $\lambda_1 = \lambda_2$  where  $\lambda_1 = \alpha_1 + \beta_1 + \gamma_1$  and  $\lambda_2 = \alpha_2 + \beta_2 + \gamma_2$ . If these two opinions are combined, the resulting opinion  $\omega_3$  can be seen as a mixture of  $\omega_1$  and  $\omega_2$ . According to the combining operation (Eq. 3.23) in 3VSL, the evidence value for the neutral state  $\gamma_3$  of  $\omega_3$  becomes  $\gamma_1 + \gamma_2$ . Assuming  $\gamma_2 \gg \gamma_1$ , we have  $\gamma_3 \gg \gamma_1$ . Combining more opinions actually increases the evidence values, so the uncertainty of the resulting opinion should decrease. However, based on Eq. 3.24, the resulting opinion  $\omega_3$  is more uncertain than  $\omega_1$  (as  $\gamma_3 \gg \gamma_1$ ), which is contradictory to the common sense. In

other words,  $\omega_2$  *polluted* the certainty of  $\omega_3$  if uncertainty is considered in computing expected belief.

We know that  $\alpha_{AX}$  and  $\beta_{AX}$  are the numbers of (negative and positive) certain evidence, so they must be used in computing the expected belief.  $\gamma_{AX}$  only records the amount of neutral evidence, so it should be omitted in the expected belief computation. Ignoring the uncertain evidence recorded as  $\gamma_{AX}$ , the DC distribution of  $\omega_X^A$  is collapsed into a Beta-Categorical (BC) distribution:

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

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 [93] to compute the expected belief as follows.

$$\begin{aligned} E_{\omega_{AX}} &= \left( \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} + \frac{\beta_{AX}}{\alpha_{AX} + \beta_{AX}} \right) a_{AX} \cdot (1 - c_{AX}) \\ &+ \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} \cdot c_{AX} \\ &= \frac{\alpha_{AX}}{\alpha_{AX} + \beta_{AX}} \cdot c_{AX} + a_{AX} \cdot (1 - c_{AX}), \end{aligned} \quad (3.24)$$

where  $c_{AX}$  is the certainty factor [93] 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. It is computed from

$$c_{AX} = \frac{1}{2} \int_0^1 \left| \frac{1}{B(\alpha_{AX}, \beta_{AX})} x^{\alpha_{AX}} (1 - x^{\beta_{AX}}) - 1 \right| dx. \quad (3.25)$$

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

## THE ASSESSTRUST ALGORITHM

In this chapter, we introduce the AssessTrust (AT) algorithm that implements the 3VSL model and is able to conduct trust assessment in social networks with arbitrary topologies. Here, we assume that social network graph does not contain cycles, i.e., we are interested in the trust assessment in a directed acyclic graph (DAG).

To ensure AT works in arbitrary DAGs topologies, we need to prove that AT can handle non-series-parallel network topologies, e.g., the bridge topology in Fig. 4.5(a). This is a challenge because the only operations available for trust computation are the discounting and combining operations. The issue is that discounting/combining operation requires the network topologies to be series/parallel. We address this challenge by differentiating the distorting and 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 values into uncertain ones, it does not change the total amount of evidence. That also implies the distorting opinion from  $A$  to  $B$ , shown in the bridge topology in Fig. 4.5(a), can appear twice in both sub-graphs  $A \rightarrow B \rightarrow C$  and  $A \rightarrow B \rightarrow D \rightarrow C$ .

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

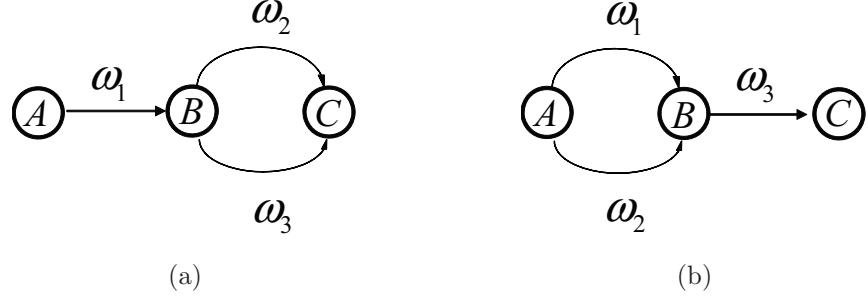


Figure 4.1: Difference between distorting and original opinions.

### Properties of Different Opinions

Before introducing the AT algorithm, we need to understand some important features of the discounting operation defined in 3VSL. For a discounting operation, there must be two opinions involved. However, the functionality of the two opinions are different.

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

To understand the difference between the distorting and original opinions, we study two special cases shown in Fig. 4.1. By analyzing them, we discover a distorting opinion can be used several times in trust computation but an original opinion can be used only once.

**Theorem 4.0.1.** *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 the two opinions on two parallel paths from users  $B$  to  $C$ . Let  $\omega_{AB} = (\alpha_{AB}, \beta_{AB}, \gamma_{AB})$  be the opinion from  $A$  to  $B$ , then the following equation will always hold:

$$\begin{aligned} & \Theta(\Delta(\omega_{AB}, \omega_{B_1C_1}), \Delta(\omega_{AB}, \omega_{B_2C_2})) \\ & \equiv \Delta(\omega_{AB}, \Theta(\omega_{B_1C_1}, \omega_{B_2C_2})). \end{aligned} \quad (4.1)$$

*Proof.* Let's take a look at the left side of Eq. 4.1:

$$\Theta(\Delta(\omega_{AB}, \omega_{B_1C_1}), \Delta(\omega_{AB}, \omega_{B_2C_2})).$$

According to the definition of the discounting operation, the result of  $\Delta(\omega_{AB}, \omega_{B_1C_1})$  can be written as

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

where

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

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:

$$\Delta(\omega_{AB}, \Theta(\omega_{B_1C_1}, \omega_{B_2C_2})). \quad (4.4)$$

The term  $\Theta(\omega_{B_1C_1}, \omega_{B_2C_2})$  in the above formula can be written as

$$\begin{aligned}
\omega_{BC} &= \Theta(\omega_{B_1C_1}, \omega_{B_2C_2}) \\
&= \langle \alpha_{BC}, \beta_{BC}, \gamma_{BC} \rangle,
\end{aligned} \quad (4.5)$$

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. 4.6 back into Eq. 4.4, we will have

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

where

$$\begin{aligned}
\alpha'_{AC} &= \frac{\alpha_{AB}(\alpha_{B_1C_1} + \alpha_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\
&= \frac{\alpha_{AB}\alpha_{B_1C_1} + \alpha_{AB}\alpha_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \\
\beta'_{AC} &= \frac{\alpha_{AB}(\beta_{B_1C_1} + \beta_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\
&= \frac{\alpha_{AB}\beta_{B_1C_1} + \alpha_{AB}\beta_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}, \\
\gamma'_{AC} &= \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{B_1C_1} + \beta_{B_1C_1} + \gamma_{B_1C_1})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_1C_1}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\
&+ \frac{(\beta_{AB} + \gamma_{AB})(\alpha_{B_2C_2} + \beta_{B_2C_2} + \gamma_{B_2C_2})}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}} \\
&+ \frac{\alpha_{AB}\gamma_{B_2C_2}}{\alpha_{AB} + \beta_{AB} + \gamma_{AB}}. \tag{4.6}
\end{aligned}$$

Clearly,  $\omega'_{AC}$  is equivalent to  $\omega_{AC}$ . □

**Theorem 4.0.2.** *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 the opinions on two parallel paths between two users  $A$  and  $B$ ; Let  $\omega_{BC} = (\alpha_{BC}, \beta_{BC}, \gamma_{BC})$  be an opinion from users  $B$  to  $C$ , then the following equation **does not hold**:

$$\begin{aligned}
&\Theta(\Delta(\omega_{A_1B_1}, \omega_{BC}), \Delta(\omega_{A_2B_2}, \omega_{BC})) \\
&\equiv \Delta(\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}), \omega_{BC}). \tag{4.7}
\end{aligned}$$

*Proof.* In Chapter 3, we have shown that the combining operation applies on  $\Theta(\omega_{A_1B_1}, \omega_{A_2B_2})$  when the evidence of  $\omega_{A_1B_1}$  and  $\omega_{A_2B_2}$  are from different sources, i.e., they are independent. In the left side of Eq. 4.7, opinions  $\Delta(\omega_{A_1B_1}, \omega_{BC})$  and  $\Delta(\omega_{A_2B_2}, \omega_{BC})$  share the same evidence from the opinion  $\omega_{BC}$ . As a result, the combining operation does not apply to them. Therefore,  $\Delta(\Theta(\omega_{A_1B_1}, \omega_{A_2B_2}), \omega_{BC})$  is the only correct solution, and is not equal to  $\Theta(\Delta(\omega_{A_1B_1}, \omega_{BC}), \Delta(\omega_{A_2B_2}, \omega_{BC}))$ .  $\square$

From Theorem 4.0.1 and 4.0.2, we note that reusing  $\omega_{AB}$  in case (a) is allowed but reusing  $\omega_{BC}$  in case (b) is not. The difference between  $\omega_{AB}$  and  $\omega_{BC}$  is that  $\omega_{AB}$  is a distorting opinion while  $\omega_{BC}$  is an original opinion. Therefore, we conclude that in trust computation, original opinions can be combined only once, while distorting opinions can be used any number of times because they do not change the total amount of evidence of final opinions.

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

*Proof.* 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.

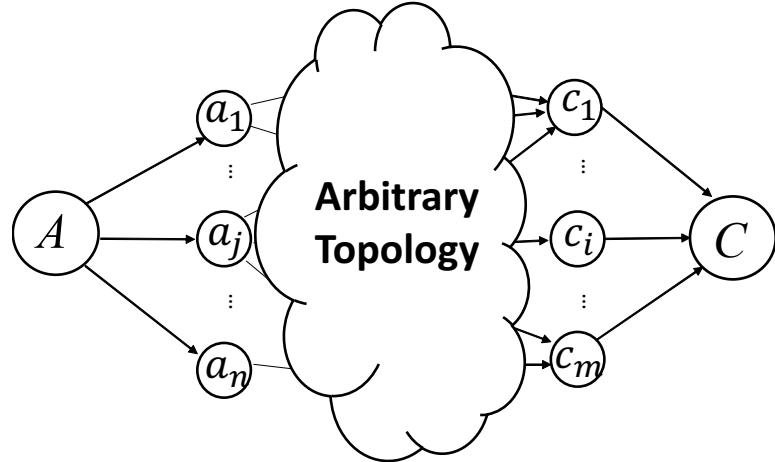


Figure 4.2: Illustration of an arbitrary network topology.

As shown in Fig. 4.2, we assume there are  $m$  nodes  $(c_1, c_2, \dots, c_m)$  connecting to  $C$ , i.e.,  $e(c_i, C) \in E$  where  $i \in [1, m]$ . There are  $n$  nodes  $(a_1, a_2, \dots, a_n)$  being connected from  $A$ , i.e.,  $e(A, a_j) \in E$  where  $j \in [1, n]$ .

#### Reduction rules

*Case 1:* If there is only one node connecting to  $C$ , i.e.,  $m = 1$ , then  $\omega_{AC} = \Delta(\omega_{Ac_1}, \omega_{c_1C})$ . In this case, we reduce the problem of computing  $\omega_{AC}$  to  $\omega_{Ac_1}$ , and  $A$  and  $c_1$  are connected by a smaller sub-graph.

*Case 2:* If there is more than one node connected to  $C$ , i.e.,  $m > 1$ ,  $\omega_{AC}$  is equal to  $\Theta(\Delta(\omega_{Ac_1}, \omega_{c_1C}), \Delta(\omega_{Ac_2}, \omega_{c_2C}), \dots, \Delta(\omega_{Ac_m}, \omega_{c_mC}))$  due to Theorem 4.0.1. Therefore,  $\omega_{AC}$  is solvable and unique if and only if each  $\omega_{Ac_i}$  is solvable and unique, where  $\omega_{Ac_i}$  corresponds to the sub-graph  $G' = G - \Sigma e(c_i, C) - C$ . In this case, we reduce the problem of computing  $\omega_{AC}$  to  $\omega_{Ac_i}$ .

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

### Base Case

The sub-graph of base case contains only one edge from  $A$  to  $a_j$  where  $j \in [1, n]$ . As  $\omega_{Aa_j}$  is known from the original graph  $G$ , the base case is solvable and its solution is unique. Applying the equations in Case 1 and 2 repeatedly, we can obtain an unique  $\omega_{AC}$ .  $\square$

### Differences between 3VSL and SL

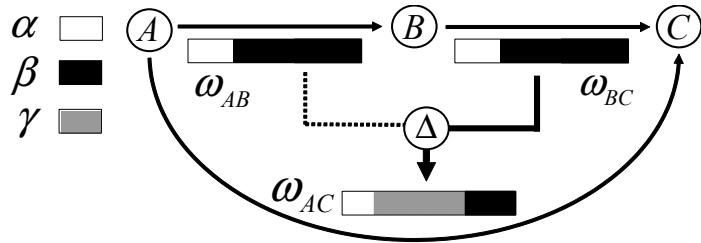
In this section, we present the differences between 3VSL and SL by introducing several examples. Compared to SL, 3VSL introduces the uncertainty state to keep track of the uncertainty generated when trust propagates within an OSN. Particularly, the uncertainty state is used to store the “distorted” positive and negative evidence in trust propagation and fusion.

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. 3.1) or parallel (see Fig. 3.2) sub-graphs so that every edge in the original graph will appear only once in the sub-graphs [44]. 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 [37–39]. The simplifications will result in information loss and inaccurate trust assessment. This problem is also observed in our numerical experiments, which will be presented in Chapter 6. Furthermore, it is not clear which edges need to be removed in a large-scale OSN, i.e., there is no algorithm for the solutions proposed in [37–39].

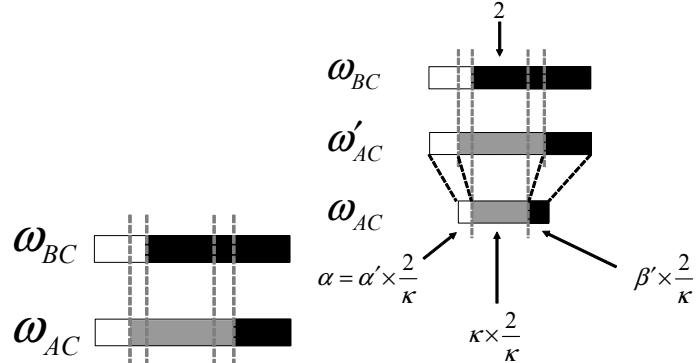
Due to the lack of the uncertainty state, SL results in inaccurate trust assessments even if it processes a social network similarly to 3VSL. We take two

examples to explain why the inaccuracy will occur.

### Example 1



(a) Illustration of the discounting operation



(b) Result of 3VSL      (c) Result of Subjective Logic

Figure 4.3: Difference between 3VSL and SL on the discounting operation.

Let's consider a series topology composed of  $A$ ,  $B$  and  $C$ , as shown in Fig 4.3(a). We assume the evidence values for  $\alpha$ ,  $\beta$  and  $\gamma$  are non-zero in both  $\omega_{AB}$  and  $\omega_{BC}$ .  $A$ 's opinion of  $C$ 's trustworthiness can be computed by applying the discounting operation defined in 3VSL (or SL) on  $\omega_{AB}$  and  $\omega_{BC}$ , i.e.,  $\omega_{AC} = \Delta(\omega_{AB}, \omega_{BC})$ .

With the 3VSL model, the total number of evidence in the resulting opinion  $\omega_{AC}$  is the same as  $\omega_{BC}$ , as shown in Fig 4.3(b). Part of  $\alpha_{BC}$  and  $\beta_{BC}$  will be transferred into  $\gamma_{AC}$ , indicating a “distortion” from positive and negative evidence values to uncertain evidence values. On the other hand, with the SL model, the distorted evidence values are merged into the prior uncertainty state, which is a fixed

number and always equals to 2, as shown in Fig 4.3(c). As a result, the positive and negative evidence values in  $\omega_{AC}$  shrink, leading to the missing of evidence.

### Example 2

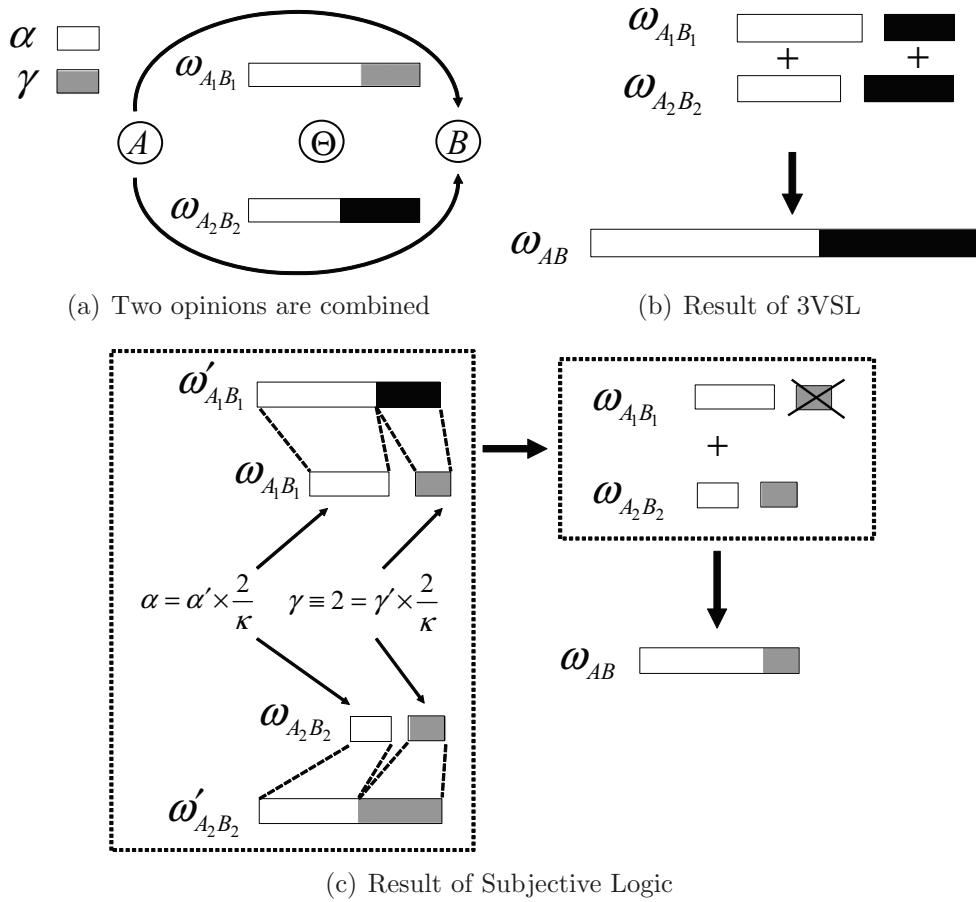


Figure 4.4: Difference between 3VSL and SL on the combining operation.

Let's consider a parallel topology shown in Fig 4.4(a).  $A$  has two parallel opinions  $\omega_{A1B1}$  and  $\omega_{A2B2}$  on  $B$ . We assume the evidence values  $\alpha$  and  $\gamma$  are non-zero and  $\beta$  is zero in both  $\omega_{A1B1}$  and  $\omega_{A2B2}$ .  $A$ 's opinion of  $B$  can be computed by applying the combining operation defined in 3VSL (or SL) on  $\omega_{A1B1}$  and  $\omega_{A2B2}$ , i.e.,  $\omega_{AB} = \Theta(\omega_{A1B1}, \omega_{A2B2})$ .

As shown in Fig 4.4(b), according to 3VSL, the numbers of positive and uncertain evidence in the resulting opinion  $\omega_{AB}$  are the sums of the positive and uncertain evidence numbers in  $\omega_{A_1B_1}$  and  $\omega_{A_2B_2}$ . As shown in Fig 4.4(c), using SL, the uncertain evidence value in the resulting opinion is always 2. According to the combining operation defined in SL, either the uncertain evidence values in  $\omega_{A_1B_1}$  or those in  $\omega_{A_2B_2}$  are ignored. As a result, the number of positive evidence values will be more than the actual.

The problems identified in the above examples will impact the accuracy of SL. On the other hand, 3VSL avoids these problems by treating uncertainty as a third state. This conclusion will be validated in Chapter 6, by comparing 3VSL and SL using two real OSN datasets.

### AssessTrust Algorithm

Based on Theorem 4.0.3, we design the AssessTrust algorithm. The algorithm is based on the 3VSL model and is able to work with arbitrary network topologies. The inputs of this algorithm include the corresponding graph  $G$ , the trustor  $A$ , the trustee  $C$ , and the maximum searching depth  $H$ , measured by number of hops. Specifically,  $H$  determines the longest distance between the trustor and trustee.  $H$  controls the searching depth on graph  $G$ , which is necessary because  $G$  could be potentially very large.  $H$  is helpful in reducing the running time of AssessTrust without sacrificing much trust assessment accuracy.

### Illustration of the AssessTrust Algorithm

In this section, we will use the bridge topology shown in Fig. 4.5(a) to illustrate how the AT algorithm computes  $A$ 's indirect opinion upon  $C$ , denoted as  $\Omega_{AD}$ .

Algorithm 4.1: AssessTrust( $G, A, C, H$ ).

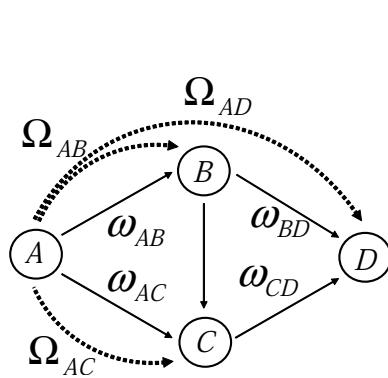
**Require:**  $G, A, C$ , and  $H$ .

**Ensure:**  $\omega_{AC}$ .

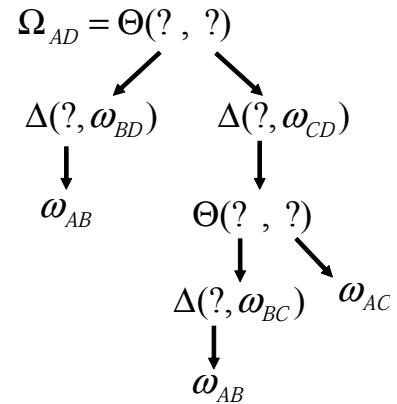
```

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:        $\omega_i \leftarrow \omega_{Ac_i}$ 
6:     else
7:        $G' \leftarrow G - e(c_i, C)$ 
8:        $\omega_{Ac_i} \leftarrow \text{AssessTrust}(G', A, c_i, H - 1)$ 
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:    $\omega_{AC} = \Theta(\omega_1 \dots \omega_n)$ 
15: else
16:    $\omega_{AC} = \omega_n$ 
17: end if
18: else
19:    $\omega_{AC} = \langle 0, 0, 0 \rangle$ 
20: end if

```



(a) Bridge topology



(b) Decomposition parsing parsing tree

Figure 4.5: An illustration of 3VSL based on the bridge topology.

To differentiate from the direct opinion, we use  $\Omega$  to denote the indirect opinion. As shown in Fig. 4.5(a), to compute  $\Omega_{AD}$ , discounting and combining operations are applied on opinions  $\omega_{AB}, \omega_{AD}, \omega_{BD}, \omega_{CD}$ , and  $\omega_{BC}$ . AT starts from the trustee  $D$  in Fig. 4.5(a), searches the network backwards and recursively computes the trustworthiness of every user during the search. As a result, we get a parsing tree, shown in Fig. 4.5(b), to describe how discounting and combining operations are applied in computing  $A$ 's opinion of  $D$ . Traversing the parsing tree in a bottom-up manner,  $A$ 's indirect opinion of  $D$ ,  $\Omega_{AD}$ , can be computed as

$$\Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\Delta(\omega_{AB}, \omega_{BC}), \omega_{AC}), \omega_{CD})). \quad (4.8)$$

To understand the time complexity of AT when it is applied on the bridge topology, we use  $AT^{(k)}(i, j)$  to denote that it is the  $k$ th time that AT is called to compute the user  $i$ 's opinion on  $j$ . At the time when AT is first called,  $A$ 's opinion on  $D$  is computed from

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

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

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

where  $\Omega_{AB}$  is computed by  $AT^{(4)}(A, B)$ . Finally,  $A$ 's opinion of  $D$  can be computed from Eq. 4.8. With the bridge topology, AT 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  $AT(A, B)$  is called twice

in this example, i.e., in sub-graphs  $A \rightarrow B \rightarrow C$  and  $A \rightarrow B \rightarrow D \rightarrow C$ , which is allowed in 3VSL.

### Time Complexity Analysis

Finally, we present the time complexity of AssessTrust in this section. Since 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 branches from the trustee node (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, 5, 6, 7, 9, 10 and 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

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

where  $k$  is the searching depth, and  $n$  is the number of nodes in the network. Note that the time complexity is for one-to-one trust assessment in OSNs. To solve the MTA problem, AT needs to go through every trustee one by one, so its time complexity is  $O(n \cdot n^k) = O(n^{k+1})$  in total. Clearly, this is unacceptable for most MTA applications. This problem motivates us to propose a more efficient solution to MTA problem, which will be shown in the next chapter.

## MASSIVE TRUST ASSESSMENT IN OSNS

One major limitation of AssessTrust is that it is inefficient in conducting massive trust assessments (MTA). To efficiently address the MTA problem, we propose the OpinionWalk algorithm that is based on AT and offers a better time complexity. To design the OpinionWalk algorithm, we need to address the following three challenges.

The first challenge is how to address the MTA problem while keeping low time complexity. This is a challenge because AT is designed for one-to-one trust assessment but MTA focuses on one-to-many situations. To address this challenge, we use an *opinion matrix* to represent the network’s topology and an *individual opinion vector* to store the trustworthiness of all nodes. In this way, similar to matrix operations, the *individual opinion vector* can be updated in a parallel manner. Based on this novel design, the time complexity is reduced from  $O(n^{k+1})$  to  $O(n^3)$ , where  $k$  is the longest distance in hops from the trustor to the trustee node.

The second challenge is to eliminate the recursive operations in the AT algorithm. This is a challenge because AT needs to first transform a trust social network into a recursion tree, and then processes its sub-trees before getting into the upper-level of the tree. To address this challenge, we design the OpinionWalk algorithm to implement this recursive procedure in an iterative way. This is non-trivial because OpinionWalk has to use the operations defined in 3VSL. As recursive operations are slower and take up more memory/stack, OpinionWalk offers a faster running time, especially in large-scale networks.

The third challenge is to show OpinionWalk is equivalent to AT, in addressing MTA. This is a challenge because these two algorithms are different and we need to prove they output the same results in arbitrary network topologies. To address this challenge, we first prove the *opinion walk* operations equivalently implement

the discounting and combination operations in 3VSL. Then, we extend the proof into arbitrary network topologies and recursively show that each case encountered in AT can be equivalently solved by OpinionWalk. In other words, OpinionWalk is an equivalent implementation of AT. Additionally, we analyze OpinionWalk's time complexity and show that it offers a better time complexity. At the end of this chapter, we also use an example to illustrate how OpinionWalk works.

### Design of OpinionWalk

OpinionWalk is essentially a matrix-based algorithm that implements 3VSL in a more efficient way to address the MTA problem. Given a trust social network  $G = (V, E, w)$ , OpinionWalk represents this graph by an *opinion matrix*  $M$ . The elements in  $M$  are edges/opinions between nodes in the graph  $G = (V, E, w)$ . The trustworthiness of all nodes are stored in the *individual opinion vector*  $Y$ . The procedure of OpinionWalk can be expressed as an iteration equation:

$$Y^{(k)} = M^T \odot Y^{(k-1)},$$

where  $k$  is the current searching depth in the graph from the trustor to the trustee. The operation rules of  $\odot$  will be introduced later. The trustworthiness of a given trustee can be obtained from  $Y$ . OpinionWalk is more efficient than AT because it uses an iterative method rather than a recursive one to address the MTA problem.

Based on 3VSL, we define two special opinions that will be used to initialize the OpinionWalk algorithm.

**Definition 4** (Uncertain Opinion). *An uncertain opinion  $\mathbb{O}$  is defined as*

$$\mathbb{O} \triangleq \langle 0, 0, 0 \rangle,$$

that indicates the trustor is totally uncertain about the trustee's trustworthiness.

**Definition 5** (Absolute Opinion). *An absolute opinion  $\mathbb{I}$  is defined as*

$$\mathbb{I} \triangleq \langle \infty, 0, 0 \rangle,$$

that indicates the trustor has infinite positive evidence, hence absolutely trusts the trustee.

Based on the uncertain opinion and 3VSL, we can have the following corollaries.

**Corollary 5.0.1.** *Applying the discounting operation on  $\mathbb{O}$  and an opinion  $\omega$ , we have  $\Delta(\omega, \mathbb{O}) = \mathbb{O}$  and  $\Delta(\mathbb{O}, \omega) = \mathbb{O}$ .*

**Corollary 5.0.2.** *Applying the combining operation on  $\mathbb{O}$  and an opinion  $\omega$ , we have  $\Theta(\omega, \mathbb{O}) = \Theta(\omega, \mathbb{O}) = \omega$ .*

Based on the absolute opinion and 3VSL, we can have the following corollaries.

**Corollary 5.0.3.** *Applying the discounting operation on  $\mathbb{I}$  and an opinion  $\omega$ , we have  $\Delta(\omega, \mathbb{I}) = \omega$  and  $\Delta(\mathbb{I}, \omega) = \omega$ .*

**Corollary 5.0.4.** *Applying the combining operation on  $\mathbb{I}$  and an opinion  $\omega$ , we have  $\Theta(\omega, \mathbb{I}) = \Theta(\omega, \mathbb{I}) = \mathbb{I}$ .*

### Opinion Matrix

**Definition 6.** *Given a trust social network containing  $n$  nodes, an opinion matrix  $M$  is an  $n \times n$  matrix:*

$$M_{n \times n} \triangleq \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \dots & \dots & \omega_{nn} \end{bmatrix},$$

where each element  $\omega_{ij}$  ( $i, j \leq n$ ) denotes the direct opinion from node  $i$  to  $j$ .

Unlike the traditional representation of a graph, e.g., adjacency or Laplace matrix, the entries in  $M$  are the direct opinions between nodes in  $G$ . If  $i$  does not have a direct opinion on  $j$ , we use the uncertain opinion to represent the corresponding entry, i.e.,  $\omega_{ij} \stackrel{\Delta}{=} \emptyset$  ( $i, j \in n, i \neq j$ ) if  $e_{ij} \notin E$ .

### Individual Opinion Vector

**Definition 7.** An individual opinion vector  $Y_i^{(k)}$  is an  $n \times 1$  column vector composed of  $n$  opinions:

$$Y_i^{(k)} \stackrel{\Delta}{=} \left[ \Omega_{i1}^{(k)}, \Omega_{i2}^{(k)}, \dots, \Omega_{ij}^{(k)}, \dots, \Omega_{in}^{(k)} \right]^T,$$

where  $\Omega_{ij}^{(k)}$  denotes user  $i$ 's individual opinion on  $j$ . The head note  $k$  indicates the current iteration step in the OpinionWalk algorithm.

### Opinion Walk Operation

**Definition 8.** An opinion walk operation  $\odot$  “multiplies” matrix  $M$  and vector  $Y_i^{(k-1)}$  to yield a new vector  $Y_i^{(k)}$  as follows.

$$\begin{aligned} Y_i^{(k)} &= M^T \odot Y_i^{(k-1)} \\ &\stackrel{\Delta}{=} \left[ \begin{array}{c} \Theta(\Delta(\Omega_{i1}^{(k-1)}, \omega_{11}), \dots, \Delta(\Omega_{in}^{(k-1)}, \omega_{n1})), \\ \Theta(\Delta(\Omega_{i1}^{(k-1)}, \omega_{12}), \dots, \Delta(\Omega_{in}^{(k-1)}, \omega_{n2})), \\ \dots \\ \Theta(\Delta(\Omega_{i1}^{(k-1)}, \omega_{1n}), \dots, \Delta(\Omega_{in}^{(k-1)}, \omega_{nn})) \end{array} \right] \\ &\stackrel{\Delta}{=} \left[ \Omega_{i1}^{(k)}, \Omega_{i2}^{(k)}, \dots, \Omega_{ij}^{(k)}, \dots, \Omega_{in}^{(k)} \right]^T, \end{aligned}$$

where  $k$  denotes the iteration step. The  $\Theta$  and  $\Delta$  implement the combining and discounting operations in 3VSL.

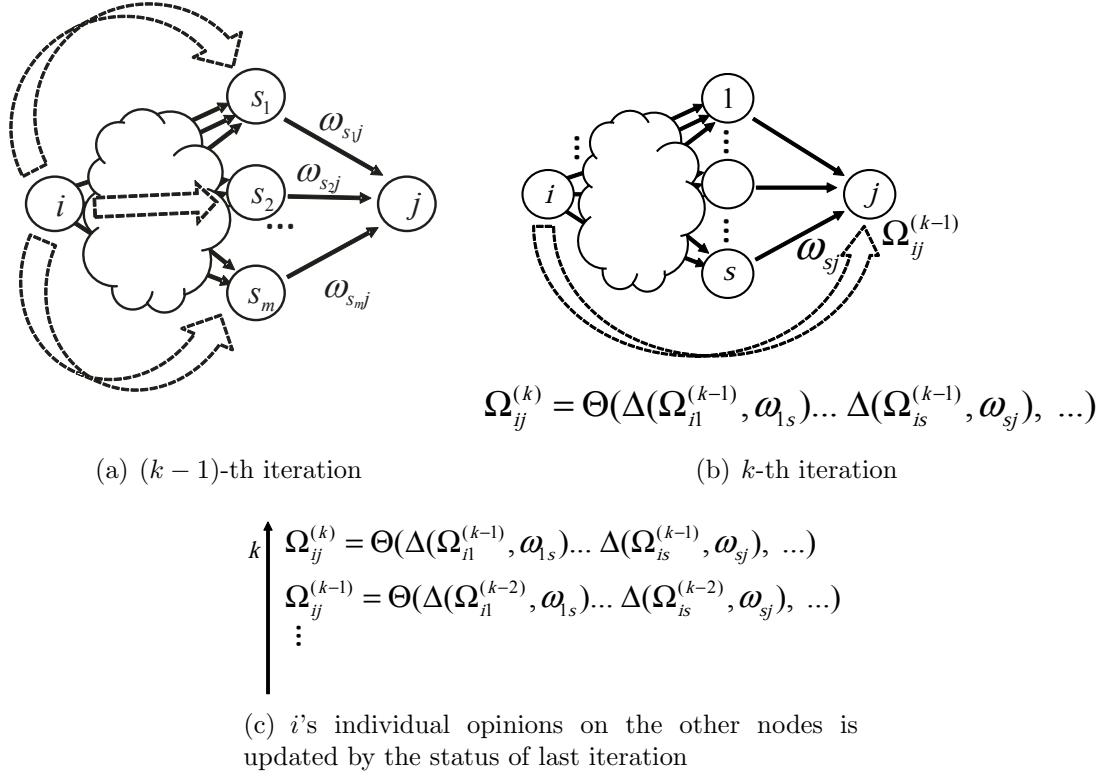


Figure 5.1: A detailed illustration of OpinionWalk.

When the OpinionWalk algorithm is initialized,  $Y^{(0)}$  is set as

$$Y_i^{(0)} = \left[ \Omega_{i1}^{(0)}, \Omega_{i2}^{(0)}, \dots, \Omega_{ii}^{(0)}, \dots, \Omega_{in}^{(0)} \right]^T$$

$$= [\mathbb{O}, \mathbb{O}, \mathbb{O}, \dots, \mathbb{I}, \dots, \mathbb{O}]^T,$$

This vector indicates node  $i$  does not trust other nodes except for itself. In the following steps, the OpinionWalk either updates  $\Omega_{ij}^{(k-1)}$  or keeps it unchanged.

A detailed explanation of the opinion walk operation can be seen in Fig. 5.1. As shown in Fig. 5.1(a), at the  $(k-1)$ -th iteration, node  $i$ 's individual opinions on all other nodes ( $\forall j \in V \setminus i$ ) are stored in the individual opinion vector, which is denoted as  $Y^{(k-1)}$ . Then, at the  $k$ -th iteration, as shown in Fig. 5.1(b),  $i$ 's individual opinion

on  $j$  ( $\forall j \in V \setminus i$ ) is updated by applying the discounting and combining operations, respectively. The discounting operation is applied to  $\Omega_{is}^{(k-1)}$  and  $\omega_{sj}$ .  $\Omega_{is}^{(k-1)}$  is  $i$ 's individual opinion on  $s \in S$  for the  $(k-1)$ -th iteration, where  $S$  is the set of  $j$ 's in-neighbors.  $\omega_{sj}$  is  $s$ 's direct opinion on  $j$ , which does not change in all iterations. As shown in Fig. 5.1(c), the logic behind the discounting operation is using  $i$ 's individual opinion on  $s$  (in the  $(k-1)$ -th iteration) and  $s$ 's direct opinion on  $j$  to form  $i$ 's “partial” opinion  $\Omega_{is|sj}$  on  $j$  (in the  $k$ -th iteration). In other words, the “partial” opinion is made through  $i$ 's individual opinion on  $s$  ( $\Omega_{is}$ ) and  $s$ 's direct opinion on  $j$  ( $\omega_{sj}$ ). The combining operation is then applied on the results of the discounting operations from above. The logic behind the combining operation is aggregating all of the “partial” opinions  $\Omega_{is|sj}$  (from  $j$ 's in-neighbor nodes) together to form  $i$ 's overall individual opinion on  $j$ , which is shown in Fig. 5.1(c).

As shown in Fig. 5.2, the OpinionWalk operation is similar to the multiplication between a matrix and a vector. The difference is that the summation and production operations are replaced by the combining and discounting operations. Notice that mathematically  $k$  denotes the iteration number of the opinion walk operation, as shown in Fig. 5.2. On the other hand, as shown in Fig. 5.1(a) and Fig. 5.1(b), the physical meaning of  $k$  is the searching depth of the OpinionWalk algorithm that originates from the trustor. Most importantly, the trustor's individual opinion on any trustee within  $k$  hops can be found from  $Y^{(k)}$ .

### OpinionWalk Algorithm

The pseudo-code of the OpinionWalk algorithm is shown in Algorithm 5.2. In the algorithm, line 3 controls how many levels OpinionWalk will search on the network. Lines 5-14 update the indirect opinion  $\Omega_{ij}$  iteratively. Line 5 considers all users, other than  $i$ , as the trustees. Lines 7-12 combines all opinions derived from

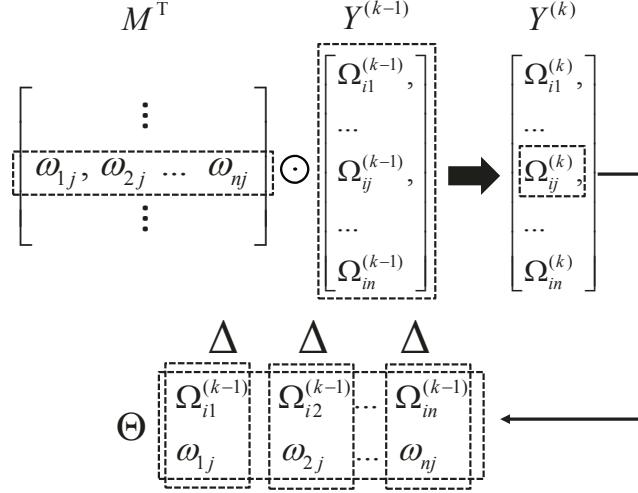


Figure 5.2: A general view of the “opinion walk” operation.

Algorithm 5.2: OpinionWalk( $G, i, H$ ).**Require:** A directed graph  $G$  with a trustor  $i$  and the maximum searching level  $H$ .**Ensure:**  $i$ ’s opinion  $j$  where  $j \neq i$ .

```

1: Initialize  $M$  and  $Y_i^{(1)}$  based on  $G$ 
2:  $k \leftarrow 1$ 
3: while  $k < H$  do
4:    $k \leftarrow k + 1$ 
5:   for all columns  $c_j \in M$  s.t.  $j \neq i$  do
6:      $\Omega_{ij}^{(k)} \leftarrow \emptyset$ 
7:     for all direct opinions  $\omega_{sj} \in c_j$  s.t.  $\omega_{sj} \neq \emptyset$  do
8:        $\Omega_{is}^{(k-1)} \leftarrow Y_i^{(k-1)}[s]$ 
9:       if  $\Omega_{is}^{(k-1)} \neq \emptyset$  then
10:         $\Omega_{ij}^{(k)} \leftarrow \Theta(\Omega_{ij}^{(k)}, \Delta(\Omega_{is}^{(k-1)}, \omega_{sj}))$ 
11:       end if
12:     end for
13:      $Y_i^{(k)}[j] \leftarrow \Omega_{ij}^{(k)}$ 
14:   end for
15: end while
16: return  $Y_i^{(k)}$ 

```

$\omega_{sj} \neq \emptyset$ . Line 8 obtains  $i$ ’s indirect opinion on one of the predecessors of  $j$ , e.g.,  $s$ . If this opinion already exists,  $i$  discounts  $s$ ’s opinion on  $j$  to update  $\Omega_{ij}^{(k)}$  at line

9. Otherwise, it checks another predecessor. Line 10 combines all opinions that are currently computed from  $\omega_{sj} \neq \emptyset$ . Note that line 10 essentially combines opinions one by one, so  $\Omega_{ij}^{(k)}$  equals to

$$\Theta(\Delta(\Omega_{i1}^{(k-1)}, \omega_{1j}), \dots, \Theta(\Delta(\Omega_{in-1}^{(k-1)}, \omega_{n-1j}), \Delta(\Omega_{in}^{(k-1)}, \omega_{nj}))).$$

Because the combining operation is associative (see corollary 5.0.2), the above equation is the same as the following form:

$$\Theta \left( \Delta(\Omega_{i1}^{(k-1)}, \omega_{1j}), \dots, \Delta(\Omega_{in}^{(k-1)}, \omega_{nj}) \right).$$

After processing all users connecting to  $j$ , at line 13, the newly computed  $\Omega_{ij}$  is used to update the corresponding element in the individual opinion vector. When  $i$ 's opinions on all possible  $j$ 's are updated, at line 14, OpinionWalk searches the next level. Finally, the vector  $Y_i^{(k)}$  will contain  $i$ 's opinions about the trustworthiness of all other users.

#### Illustration of the OpinionWalk Algorithm

In this section, we use the example shown in Fig. 4.5(a) to illustrate how OpinionWalk is used to compute the trustworthiness of all users ( $B$ ,  $C$  and  $D$ ), from the perspective of  $A$ .

The opinion matrix of the corresponding graph in Fig. 4.5(a) can be expressed as

$$M^T = \begin{bmatrix} \mathbb{O} & \mathbb{O} & \mathbb{O} & \mathbb{O} \\ \omega_{AB} & \mathbb{O} & \mathbb{O} & \mathbb{O} \\ \omega_{AC} & \omega_{BC} & \mathbb{O} & \mathbb{O} \\ \mathbb{O} & \omega_{BD} & \omega_{CD} & \mathbb{O} \end{bmatrix}.$$

Because we want to evaluate  $A$ 's opinions on other users, the algorithm starts from  $A$  and ends at  $D$ . Hence, we set the initial individual opinion vector as

$$Y_A^{(0)} = [\Omega_{AA}^{(0)}, \Omega_{AB}^{(0)}, \Omega_{AC}^{(0)}, \Omega_{AD}^{(0)}]^T = [\mathbb{I}, \mathbb{O}, \mathbb{O}, \mathbb{O}]^T.$$

After the initialization, the algorithm will go through several iterations that can be expressed as

$$\begin{aligned} Y_A^{(k)} &= M^T \odot Y_A^{(k-1)} \\ &= \begin{bmatrix} \Theta(\Delta(\Omega_{AA}^{(k-1)}, \omega_{AA})), \\ \Theta(\Delta(\Omega_{AA}^{(k-1)}, \omega_{AB})), \\ \Theta(\Delta(\Omega_{AA}^{(k-1)}, \omega_{AC}), \Delta(\Omega_{AB}^{(k-1)}, \omega_{BC})), \\ \Theta(\Delta(\Omega_{AB}^{(k-1)}, \omega_{BD}), \Delta(\Omega_{AC}^{(k-1)}, \omega_{CD})) \end{bmatrix}, \end{aligned}$$

where  $k$  is the number of iterations.

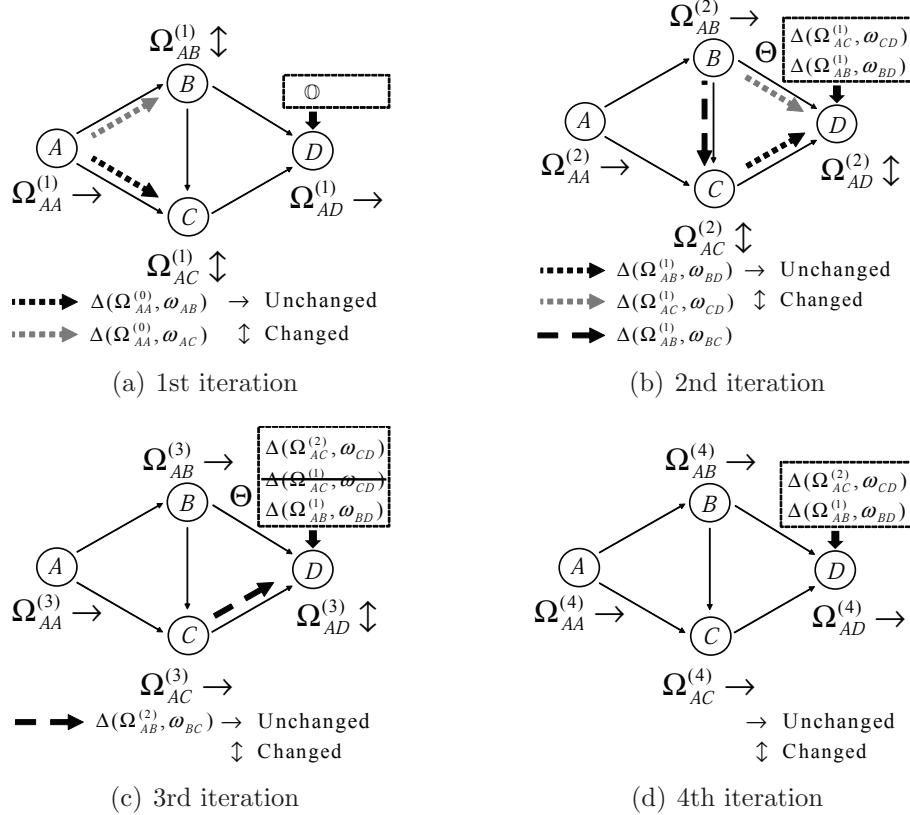


Figure 5.3: Illustration of how OpinionWalk processes the bridge topology. The dashed box shows how the combining operation in OpinionWalk works.

The first iteration is shown Fig. 5.3(a) and can be expressed as

$$\begin{aligned}
 Y_A^{(1)} &= M^T \odot Y_A^{(0)} \\
 &= \left[ \begin{array}{l} \Theta(\Delta(\mathbb{I}, \mathbb{I})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AB})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AC}), \Delta(\mathbb{O}, \omega_{BC})), \\ \Theta(\Delta(\mathbb{O}, \omega_{BD}), \Delta(\mathbb{O}, \omega_{CD})) \end{array} \right] \\
 &= [\mathbb{I}, \Theta(\mathbb{O}, \Delta(\mathbb{I}, \omega_{AB})), \Theta(\mathbb{O}, \Delta(\mathbb{I}, \omega_{AC})), \mathbb{O}]^T \\
 &= [\mathbb{I}, \omega_{AB}, \omega_{AC}, \mathbb{O}]^T.
 \end{aligned}$$

In this iteration, because both  $\Omega_{AB}^{(0)}$  and  $\Omega_{AC}^{(0)}$  are still  $\mathbb{O}$ , we have

$$\begin{aligned}\Omega_{AD}^{(1)} &= \Theta(\Delta(\Omega_{AB}^{(0)}, \omega_{BD}), \Delta(\Omega_{AC}^{(0)}, \omega_{CD})) \\ &= \Omega_{AD}^{(0)}.\end{aligned}$$

The result of the second iteration is shown in Fig. 5.3(b) and can be expressed as

$$\begin{aligned}Y_A^{(2)} &= M^T \odot Y_A^{(1)} = \left[ \Omega_{AA}^{(2)}, \Omega_{AB}^{(2)}, \Omega_{AC}^{(2)}, \Omega_{AD}^{(2)} \right]^T \\ &= \left[ \begin{array}{l} \Theta(\Delta(\mathbb{I}, \mathbb{I})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AB})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AC}), \Delta(\omega_{AB}, \omega_{BC})), \\ \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\omega_{AC}, \omega_{CD})) \end{array} \right]^T \\ &= [\mathbb{I}, \omega_{AB}, \Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \\ &\quad \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\omega_{AC}, \omega_{CD}))]^T.\end{aligned}$$

In this iteration, as both  $\Omega_{AB}^{(1)}$  and  $\Omega_{AC}^{(1)}$  have changed (compared to  $\Omega_{AB}^{(0)}$  and  $\Omega_{AC}^{(0)}$ ), we have

$$\begin{aligned}\Omega_{AD}^{(2)} &= \Theta(\Delta(\Omega_{AB}^{(1)}, \omega_{BD}), \Delta(\Omega_{AC}^{(1)}, \omega_{CD})) \\ &= \Theta(\Delta(\Omega_{AB}^{(1)}, \omega_{BD}), \Delta(\Omega_{AC}^{(1)}, \omega_{CD})).\end{aligned}$$

The result of the third iteration shown in Fig. 5.3(c) reflects the following computation.

$$\begin{aligned}
Y_A^{(3)} &= M^T \odot Y_A^{(2)} = \left[ \Omega_{AA}^{(3)}, \Omega_{AB}^{(3)}, \Omega_{AC}^{(3)}, \Omega_{AD}^{(3)} \right]^T \\
&= \left[ \begin{array}{l} \Theta(\Delta(\mathbb{I}, \mathbb{O})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AB})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AC}), \Delta(\omega_{AB}, \omega_{BC})), \\ \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \omega_{CD})) \end{array} \right] \\
&= [\mathbb{I}, \omega_{AB}, \Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \\
&\quad \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \omega_{CD}))]^T.
\end{aligned}$$

It is worth mentioning that  $\Omega_{AB}^{(2)}$  did not change, but  $\Omega_{AC}^{(2)}$  has changed, so we update  $\Omega_{AD}$  by substituting  $\Omega_{AC}^{(1)}$  with  $\Omega_{AC}^{(2)}$  as follows.

$$\begin{aligned}
\Omega_{AD}^{(3)} &= \Theta(\Delta(\Omega_{AB}^{(2)}, \omega_{BD}), \Delta(\Omega_{AC}^{(2)}, \omega_{CD})) \\
&= \Theta(\Delta(\Omega_{AB}^{(1)}, \omega_{BD}), \Delta(\Omega_{AC}^{(2)}, \omega_{CD})).
\end{aligned}$$

In the end, the fourth iteration (see Fig. 5.3(d)) can be expressed as

$$\begin{aligned}
Y_A^{(4)} &= M^T \odot Y_A^{(3)} = \left[ \Omega_{AA}^{(4)}, \Omega_{AB}^{(4)}, \Omega_{AC}^{(4)}, \Omega_{AD}^{(4)} \right]^T \\
&= \left[ \begin{array}{l} \Theta(\Delta(\mathbb{I}, \mathbb{O})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AB})), \\ \Theta(\Delta(\mathbb{I}, \omega_{AC}), \Delta(\omega_{AB}, \omega_{BC})), \\ \Theta(\Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \Delta(\omega_{AB}, \omega_{BC})) \end{array} \right] \\
&= [\mathbb{I}, \omega_{AB}, \Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \\
&\quad \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \omega_{CD}))]^T.
\end{aligned}$$

In this iteration, neither  $\Omega_{AB}^{(3)}$  nor  $\Omega_{AC}^{(3)}$  changed, so we have

$$\begin{aligned}\Omega_{AD}^{(4)} &= \Theta(\Delta(\Omega_{AB}^{(3)}, \omega_{BD}), \Delta(\Omega_{AC}^{(3)}, \omega_{CD})) \\ &= \Omega_{AD}^{(3)}.\end{aligned}$$

The components in the final individual opinion vector are

$$\begin{aligned}\Omega_{AA}^{(4)} &= \mathbb{I}, \\ \Omega_{AB}^{(4)} &= \omega_{AB}, \\ \Omega_{AC}^{(4)} &= \Theta(\Delta(\omega_{AB}, \omega_{BC}), \omega_{AC}), \\ \Omega_{AD}^{(4)} &= \Theta(\Delta(\omega_{AB}, \omega_{BD}), \Delta(\Theta(\omega_{AC}, \Delta(\omega_{AB}, \omega_{BC})), \omega_{CD})).\end{aligned}$$

which are exactly the same as those obtained by the AT algorithm.

### Correctness of OpinionWalk

To prove OpinionWalk equivalently implements the AT algorithm, we first show both AT and OpinionWalk generate the same result if network topology is either series or parallel. Then, we show this is true for arbitrary network topologies.

If we zoom into a trust social network, two edges can be connected in series if they are incident to a vertex of degree 2, or in parallel if they join the same pair of distinct vertices. Therefore, two users can be connected in a series topology shown in Fig. 5.4(a), or a parallel topology shown in Fig. 5.4(b). Note that the paths from  $i$  to  $s_1, s_2, \dots, s_m$  in Fig. 5.4(b) are disjoint, i.e., no sharing edges along the paths.

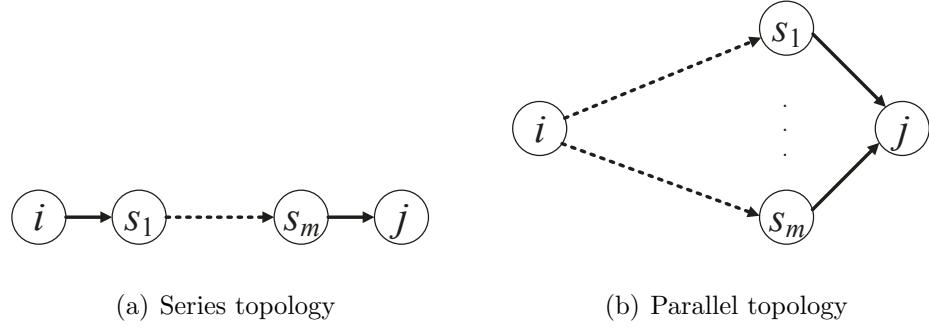


Figure 5.4: Illustration of two fundamental topologies in an OSN.

### Series Network Topology

**Lemma 5.0.1.** *Given two users  $i$  and  $j$  who are connected by  $m$  users in a series topology, the opinion  $\Omega_{ij}$  computed from AT and OpinionWalk will be the same.*

*Proof.* We use  $s_1, s_2, \dots, s_m$  to denote the users that connects  $i$  to  $j$ , as shown in Fig. 5.4(a). According to the AT algorithm,  $i$ 's trustworthiness of  $j$  is

$$\Omega_{ij} = \Delta(\omega_{is_1}, \Delta(\omega_{s_1s_2}, \dots, \Delta(\omega_{s_{m-1}s_m}, \omega_{s_mj}))). \quad (5.1)$$

According to the OpinionWalk algorithm, in the opinion matrix, except for opinions  $\omega_{is_1}, \omega_{s_1s_2}, \dots, \omega_{s_mj}$ , all other opinions are uncertain opinions  $\mathbb{O}$ . The initial individual opinion vector is

$$Y_i^{(1)} = [\mathbb{O}, \dots, \omega_{is_1}, \dots, \mathbb{O}]^T.$$

If OpinionWalk searches the 2nd level of the network, the individual opinion vector is updated to

$$Y_i^{(2)} = [\mathbb{O}, \dots, \omega_{is_1}, \dots, \Delta(\omega_{is_1}, \omega_{s_1s_2}), \dots, \mathbb{O}]^T.$$

where  $\Omega_{is_1}^{(2)} = \omega_{is_1}$  and  $\Omega_{is_2}^{(2)} = \Delta(\omega_{is_1}, \omega_{s_1s_2})$  are  $i$ 's current opinions on users  $s_1$  and  $s_2$ , respectively.

When OpinionWalk searches the  $(m+1)$ -th level, it will reach user  $j$ , and  $Y_i^{(m+1)}$  becomes

$$\begin{aligned} Y_i^{(m+1)} &= M^T \odot Y_i^{(m)} \\ &= \left[ \mathbb{O}, \dots, \Omega_{is_m}^{(m)}, \dots, \Delta(\Omega_{is_m}^{(m)}, \omega_{s_mj}), \dots, \mathbb{O} \right]^T \\ &= \left[ \mathbb{O}, \dots, \Omega_{is_m}^{(m+1)}, \dots, \Omega_{ij}^{(m+1)}, \dots, \mathbb{O} \right]^T. \end{aligned}$$

If we expand  $\Omega_{ij}^{(m+1)}$  in the above equation, we will get

$$\begin{aligned} \Omega_{ij}^{(m+1)} &= \Delta(\Omega_{is_m}^{(m)}, \omega_{s_mj}) \\ &= \Delta(\Delta(\Omega_{is_{m-1}}^{(m-1)}, \omega_{s_{m-1}s_m}), \omega_{s_mj}) \\ &= \Delta(\Delta(\Delta(\omega_{is_1}, \omega_{s_1s_2}), \dots, \omega_{s_{m-1}s_m}), \omega_{s_mj}) \end{aligned} \tag{5.2}$$

Because the discount operation is associative [66], Eqs. 5.1 and 5.2 give the same result. Therefore, given a series topology, OpinionWalk equivalently implements AT. □

### Parallel Network Topology

**Lemma 5.0.2.** *Given two users  $i$  and  $j$  who are connected by  $m$  users in a parallel topology, the opinion  $\Omega_{ij}$  computed from AT and OpinionWalk will be the same.*

*Proof.* We use  $s_1, s_2, \dots, s_m$  to denote the  $m$  users based on their distances to  $i$ , i.e.,  $s_m$  is the farthest away from  $i$ . As shown in Fig. 5.4(b), because the paths from  $i$  to

$s_1, s_2, \dots, s_m$  are disjointed, the AT algorithm computes  $i$ 's opinion on  $j$  as

$$\Theta(\Delta(\Omega_{is_1}, \omega_{s_1j}), \Delta(\Omega_{is_2}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}, \omega_{s_mj})). \quad (5.3)$$

According to the OpinionWalk algorithm, the opinion matrix will be

$$M = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \omega_{s_1j} & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \omega_{s_2j} & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \omega_{s_mj} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}.$$

Let's assume after  $k_1, k_2, \dots, k_m$  searches, the OpinionWalk algorithm reaches users  $s_1, s_2, \dots, s_m$ , respectively. After  $k_1$  searches, OpinionWalk obtains  $i$ 's opinion on  $s_1$  as  $\Omega_{is_1}^{(k_1)}$ . Here,  $\Omega_{is_1}^{(k_1)}$  is equal to  $\Omega_{is_1}$  computed from AT in Eq. 5.3, due to Lemma 5.0.1.

During the  $(k_1 + 1)$ -th search, OpinionWalk reaches user  $j$ , and it updates  $i$ 's opinion on  $j$  to

$$\Omega_{ij}^{(k_1+1)} = \Delta(\Omega_{is_1}, \omega_{s_1j}).$$

After  $k_2$  searches, OpinionWalk gets  $i$ 's opinion on  $s_2$  as  $\Omega_{is_2}$ . At the  $(k_2 + 1)$ -th search, OpinionWalk hits user  $j$  and updates  $i$ 's opinion on  $j$  to

$$\Omega_{ij}^{(k_2+1)} = \Theta\left(\Omega_{ij}^{(k_2)}, \Delta(\Omega_{is_2}, \omega_{s_2j})\right).$$

From  $(k_1 + 1)$ -th to  $k_2$ -th searches, opinion  $\Omega_{ij}$  is not updated, so we have  $\Omega_{ij}^{(k_2)} = \Omega_{ij}^{(k_1+1)}$ . Therefore, we have

$$\Omega_{ij}^{(k_2+1)} = \Theta(\Delta(\Omega_{is_1}, \omega_{s_1j}), \Delta(\Omega_{is_2}, \omega_{s_2j})).$$

Similarly, after  $k_m + 1$  searches,  $i$ 's opinion on  $j$  is updated to

$$\Theta\left(\Delta(\Omega_{is_1}^{(k_m)}, \omega_{s_1j}), \Delta(\Omega_{is_2}^{(k_m)}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}^{(k_m)}, \omega_{s_mj})\right).$$

For every  $l = 1, 2, \dots, m$ , we have  $\Omega_{is_l}^{(k_l)} = \Omega_{is_l}^{(k_m)}$  as it does not change after the  $k_l$ -th search. Therefore, the above equation becomes

$$\Theta\left(\Delta(\Omega_{is_1}^{(k_1)}, \omega_{s_1j}), \Delta(\Omega_{is_2}^{(k_2)}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}^{(k_m)}, \omega_{s_mj})\right).$$

For any  $l = 1, 2, \dots, m$ , we know that  $\Omega_{is_l}^{(k_l)}$  is equal to  $\Omega_{is_l}$  computed by AT, due to Lemma 5.0.1. Therefore, OpinionWalk computes  $i$ 's opinion on  $j$  as

$$\Theta(\Delta(\Omega_{is_1}, \omega_{s_1j}), \Delta(\Omega_{is_2}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}, \omega_{s_mj})). \quad (5.4)$$

Because Eqs. 5.3 and 5.4 give the same result, we conclude that OpinionWalk equivalently implements AT on a parallel topology.  $\square$

### Arbitrary Topology

Given an arbitrary network topology shown in Fig. 5.5, we assume  $m > 1$  nodes  $\{s_1, s_2, \dots, s_m\}$  directly connect to  $j$ . We use  $k_1, k_2, \dots, k_m$  to denote the last times when users  $s_1, s_2, \dots, s_m$  are visited by the OpinionWalk algorithm, respectively. Let  $\{t_1, t_2, \dots, t_r\}$  denote the users who are connected from  $i$ , via either series or parallel topologies.

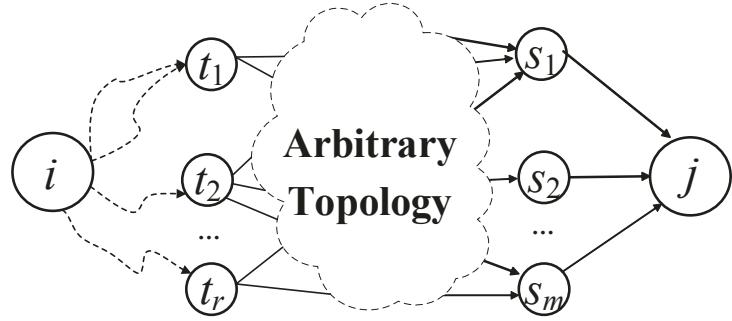


Figure 5.5: Illustration of a network with an arbitrary topology.

**Theorem 5.0.1.** *OpinionWalk equivalently implements the AT algorithm in an arbitrary network topology.*

*Proof.* We prove the theorem in a recursive manner, i.e., reducing the original network into sub-network(s) and keep reducing the sub-network(s) until the base case is reached, i.e., two users are connected via either a series or parallel topology.

#### Reduction rules

*Case 1:* There is only one user connecting to  $j$ , i.e.,  $m = 1$ . In this case, according to the AT algorithm,  $\Omega_{ij}$  is computed as  $\Delta(\Omega_{is_1}, \omega_{s_1j})$ , where  $\Omega_{is_1}$  denotes  $i$ 's opinion on  $s_1$ , and it can be computed by recursively calling the AT algorithm. For OpinionWalk, at the  $(k_1 + 1)$ -th search, it reaches  $j$  and updates  $i$ 's opinion on  $j$  as

$$\Delta(\Omega_{is_1}^{(k_1)}, \omega_{s_1j}).$$

*Case 2:* There are more than one user connecting to  $j$ , i.e.,  $m > 1$ . In this case, AT computes  $\Omega_{ij}$  as

$$\Theta(\Delta(\Omega_{is_1}, \omega_{s_1j}), \Delta(\Omega_{is_2}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}, \omega_{s_mj})).$$

After  $\hat{k} + 1$  searches where  $\hat{k} = \max(k_1, k_2, \dots, k_m)$ , OpinionWalk updates  $i$ 's opinion on  $j$  to

$$\Theta(\Delta(\Omega_{is_1}^{(\hat{k})}, \omega_{s_1j}), \Delta(\Omega_{is_2}^{(\hat{k})}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}^{(\hat{k})}, \omega_{s_mj})).$$

For any user  $s_l$ , where  $l = 1, 2, \dots, m$ , because  $k_l$  is the last time that  $s_l$  was visited by OpinionWalk,  $\Omega_{is_l}$  was not updated after the  $k_l$ -th search. Therefore, the above equation can be rewritten as

$$\Theta(\Delta(\Omega_{is_1}^{(k_1)}, \omega_{s_1j}), \Delta(\Omega_{is_2}^{(k_2)}, \omega_{s_2j}), \dots, \Delta(\Omega_{is_m}^{(k_m)}, \omega_{s_mj})).$$

Summarizing the above two cases, we know that if  $\Omega_{is_l} = \Omega_{is_l}^{(k_l)}$  for every  $l = 1, 2, \dots, m$ , then OpinionWalk and AT yield the same result of  $\Omega_{ij}$ . For any user  $s_l$ ,  $\Omega_{is_l}$  and  $\Omega_{is_l}^{(k_l)}$  actually denote  $i$ 's opinion on  $s_l$  computed by AT and OpinionWalk, respectively. Both AT and OpinionWalk will work on the same sub-network  $G' = G - e(s_l, j)$  that connects  $i$  to  $s_l$ . The sub-network  $G'$  can be further reduced by removing the edge connecting to  $s_l$ . Continuing this process, the original network  $G$  will eventually be reduced to the following base case.

### Base Case

In the base case, user  $i$  connects to users  $t_1, t_2, \dots, t_r$  via either series or parallel topologies. Based on Lemmas 5.0.1 and 5.0.2, AT and OpinionWalk give the same

values of  $\Omega_{it_1}, \Omega_{it_2}, \dots, \Omega_{it_r}$ . Overall, we prove that AT and OpinionWalk give the same result.  $\square$

### Time Complexity Analysis

Let's further investigate the time complexity of OpinionWalk in DAG. For OpinionWalk shown in Algorithm 5.2, the time complexity of the “for” loop from line 7 to line 12 is  $n \cdot C_1$ . Lines 5 – 14 take  $n \cdot (n \cdot C_1 + C_2)$ , and lines 3 – 15 take  $k \cdot n \cdot (n \cdot C_1 + C_2)$ . For the first iteration, there is only one node (the starting node) to process. After the first iteration, the trustworthiness of all trustees within  $k$  hops from a trustor can be obtained from  $Y^{(k)}$  after OpinionWalk runs  $k$  iterations. Therefore, the time complexity of OpinionWalk is

$$\begin{cases} O(k(n + n)) = O(n) \ (k = 1) \\ O(k(n^2 + n)) = O(n^3) \ (k > 1) \end{cases}, \quad (5.5)$$

where  $k$  is the searching depth or number of iterations.

In addition, the time complexity of the “for” loop from line 7 to line 12 is actually determined by the in-degree of node  $j$ . The in-degree of a node ranges from 0 to  $n$ , where  $n$  is the total node number of  $G$ . For example, in a series topology where the in-degree of each intermediate node is 1, the time complexity of OpinionWalk will be  $k \cdot n \cdot (C_1 + C_2) = O(n)$ . Since the average degree of a real social network graph is often far less than  $n$ , the execution time of OpinionWalk in real social networks increases slowly as the trustee set size  $n$  becomes larger. This finding will be validated in Chapter 6. Note that OpinionWalk is a polynomial-time solution in DAG. If the given graph contains loop(s), then the convergence of OpinionWalk is unknown. As a result, the time complexity of OpinionWalk in graphs containing loop is an open

issue, which is considered our future work. Since there are always loops in real social networks, the searching depth of OpinionWalk is empirically set. Our experiments in Chapter 6 show that even though OpinionWalk cannot get exact results in social networks containing loops, the trust assessment results are still promising.

Lower time complexity is not the only advantage of OpinionWalk. Since an iterative procedure is used in OpinionWalk, it offers faster running time compared to AT that involves large numbers of stack operations and consumes much more system memory. Faster execution time is especially useful when the network size is large and complex, which is common in OSNs. Moreover, during each iteration in OpinionWalk, a node's trustworthiness is updated only based on its value in the previous iteration, meaning OpinionWalk can be implemented in a distributed manner.

#### Differences between AssessTrust and OpinionWalk Algorithms

The AT algorithm's time complexity is  $O(n^k)$  where  $k$  is the number of hops on the longest path between the trustor and trustee, and  $n$  is the number of nodes in a DAG. The basic idea of AT algorithm is finding all possible paths between the trustor and trustee. From these paths, opinions will be discounted and combined to generate the final opinion between the trustor and trustee. However, finding all possible paths between two nodes is an NP-hard problem. In fact, we can conduct trust assessment between two nodes without tracking all possible paths between them. This is why we propose the OpinionWalk algorithm to solve the problem. OpinionWalk is designed to replace the AT algorithm in solving the trust assessment problem.

OpinionWalk algorithm is a breath first search (BFS) based algorithm that takes  $O(kn^2)$  time complexity to solve the trust assessment problem between any two nodes in a DAG network, where  $k$  is the number of hops of the longest distance between the trustor and trustee nodes. Analogous to BFS, OW searches the graph

level by level. On a certain level, every node on this level needs to search all of its children. After  $k$  rounds of searches, the algorithm will stop because it will never hit the trustee in future searches. OpinionWalk has a lower time complexity because it does not have to repeatedly consider some edges in the graph, which is inevitable in the problem of finding all possible paths.

When OpinionWalk searches the graph, we note that the trustworthiness of every other node can be computed as well. Therefore, OpinionWalk can be used to solve the MTA problem. Its time complexity in addressing MTA in a DAG is  $O(k^*n^2)$ , where  $k^*$  is the number of hops of the longest distance between the trustor and any other node in the graph. The reason why OpinionWalk has a lower time complexity in addressing MTA is that the trustor can use its opinions about nodes on a certain level to compute its opinions of nodes on the next level. As a result, when the OpinionWalk algorithm finishes searching the longest path(s), the opinions of all other nodes can be computed. In a DAG, we know  $k \leq n - 1$ . Therefore, the time complexity of OpinionWalk in a DAG can also be expressed as  $O(n^3)$ .

Note that neither AT nor OW can give exact results in a graph containing loops. The computed results will oscillate and the convergence is not well understood. Therefore, the complexity of trust assessment in the general case is currently unknown, which is considered our future work.

### Distributed OpinionWalk Algorithm

OpinionWalk can be implemented in a distributed manner, making it applicable to a distributed system. Suppose a user wants to know the trustworthiness of the users around him and these users are directly or indirectly connected to the user. If the OpinionWalk algorithm is executed, the opinions between any two connected users must be known. As the trust information between users is private and sensitive, some

users may not want to provide it. Even though some users are willing to share the data, they don't want the information to be stored on a central server. To mitigate users' concerns about their privacy, we design the distributed OpinionWalk, called D-OpinionWalk, as shown in Algorithm 5.3.

Algorithm 5.3: D-OpinionWalk Algorithm that is executed on user  $j$ .

**Require:** Message  $\{\Delta(\Omega_{is}, \omega_{sj}), k\}$ , where  $s \in S_j$  .  
**Ensure:** Message  $\{\Delta(\Omega_{ij}, \omega_{jo}), k + 1\}$ , where  $o \in O_j$  .

```

1: for all received messages  $\{\Delta(\Omega_{is}, \omega_{sj}), k\}$  on user  $j$  do
2: if  $s \in S_j$  and  $k < H$  then
3:    $Y_j[s] \leftarrow \Delta(\Omega_{is}, \omega_{sj})$ 
4: end if
5: end for
6: for all  $y \in Y_j[:]$  and  $y \neq j$  do
7:   if  $\Omega_{ij} \triangleq \emptyset$  then
8:      $\Omega_{ij} \leftarrow y$ 
9:   else
10:     $\Omega_{ij} \leftarrow \Theta(\Omega_{ij}, y)$ 
11:   end if
12: end for
13: for all  $o \in O_j$  do
14:   send a message  $\{\Delta(\Omega_{ij}, \omega_{jo}), k + 1\}$  to user  $o$ 
15: end for
```

Suppose that the D-OpinionWalk algorithm is running on a user  $j$  that receives a set of messages from users in  $S_j$ . After  $j$  executes the D-OpinionWalk algorithm, it will send another set of messages to users in  $O_j$ . For any user  $s \in S_j$ , there must be an edge/opinion from  $s$  to  $j$  in the corresponding social network. Similarly, for any user  $o \in O_j$ , there must be an edge from  $j$  to  $o$  in the social network. We call  $S_j$  and  $O_j$  user  $j$ 's in-neighbor and out-neighbor lists, respectively. In a received message, e.g., sent from user  $s$ , the following two pieces of information will be provided:  $i$ 's opinion about  $j$ 's trustworthiness  $\Delta(\Omega_{is}, \omega_{sj})$  and the number of hops  $k$ . User  $j$  also

maintains the individual opinion vector  $Y_j$  to store  $\Delta(\Omega_{is}, \omega_{sj})$  received from  $s$ . Here,  $\Delta(\Omega_{is}, \omega_{sj})$  can also be expressed as  $\Omega_{is|sj}$ .

The details of Algorithm 5.3 can be explained as follows. For lines 1-5, user  $j$  receives messages from users in  $S_j$  and initializes its individual opinion vector  $Y_j$ . In the messages,  $k$  denotes how far the trustor  $i$  is away from user  $j$ , measured by the number of hops. For each received message,  $j$  first checks if it is sent from any node in  $S$  and whether  $k$  is less than  $H$ . If so, it saves the information of  $\Delta(\Omega_{is}, \omega_{sj})$  into  $Y_j$ , in line 3. If there is already a record of  $\Delta(\Omega_{is}, \omega_{sj})$  in  $Y_j$ ,  $j$  will update it.

After  $j$  receives messages from all users in  $S_j$ , it applies the combining operation on all opinions saved in  $Y_j$ , as shown in lines 6-12. In the end, for each node  $o \in O_j$ ,  $j$  applies the discounting operation on the resulting opinion and  $j$ 's direct opinions to get  $\Delta(\Omega_{ij}, \omega_{jo})$ . Finally, it sends message  $(\Delta(\Omega_{ij}, \omega_{jo}), k + 1)$  to user  $o$ , in line 14.

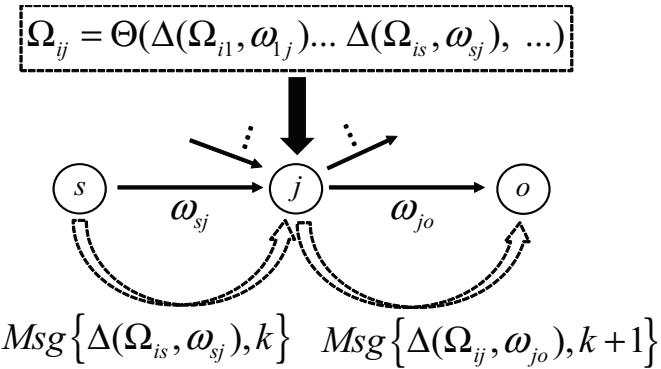


Figure 5.6: A general view of the D-OpinionWalk algorithm.

In Algorithm 5.3, there is an important feature that needs to be emphasized here. User  $j$  is unable to know his neighbor's direct opinion about himself, which protects the privacy of users in  $S_j$ . As shown in Fig. 5.6, what  $j$  received from his in-neighbors are  $\Delta(\Omega_{is}, \omega_{sj})$  and  $k$ , from which he cannot infer  $\omega_{sj}$ . Second, D-OpinionWalk requires each user to provide her/his direct opinion on the given out-neighbors and push the computed individual opinions to them, respectively. Similarly,

because  $j$  does not provide his opinion on any user  $o \in O_j$ , its privacy is protected as well.

## EVALUATIONS

In this chapter, we evaluate the properties and performance of the 3VSL model, AT algorithm and OpinionWalk algorithm. For 3VSL, we conduct a numerical analysis to show the properties of discounting and combining operations. Then, we present the results of applying 3VSL to compute trustworthiness in the bridge topology. We also evaluate the accuracy of 3VSL using a dataset collected from an online survey system. In the end, we conduct comprehensive experiments to evaluate its accuracy and compare its performance to 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. For OpinionWalk, we evaluate its accuracy and execution time in solving the MTA problem using the same two datasets.

### Numerical Analysis

To understand whether 3VSL accurately models trust in OSNs, we first conduct a numerical analysis on the discounting and combining operations defined in 3VSL. After that, we investigate whether 3VSL can be used to compute trust in the bridge topology. For the sake of simplicity, we denote the total evidence value in an opinion as  $\lambda$ , i.e.,  $\lambda = \alpha + \beta + \gamma$ .

### Discounting Operation

We first look at the discounting operation defined in 3VSL, i.e., an opinion  $\omega_{AB}$  discounts another opinion  $\omega_{BC}$ . As a result, opinion  $\omega_{AC}$  is derived from  $\Delta(\omega_{AB}, \omega_{BC})$ .

We assume the initial evidence in an opinion is 30, i.e.,  $\lambda_{AB} = \lambda_{BC} = 30$ . We also assume there is no neutral evidence in  $\lambda_{AB}$  and  $\lambda_{BC}$ . In the experiment, we change the numbers of positive evidence,  $\alpha_{AB}$  and  $\alpha_{BC}$ , from 0 to 27. We investigate the accuracy of 3VSL by observing the expected belief of  $\omega_{AC}$ .

As shown in Fig. 6.1, when  $\alpha_{AB}$  increases, the expected belief of  $\omega_{AC}$ , denoted by  $E_{\omega_{AC}}$ , increases as well. Similar results are observed when  $\alpha_{BC}$  increases. It implies that  $A$  tends to believe  $B$ 's opinion on  $C$  if  $A$  highly trusts  $B$ . When  $\alpha_{AB}$  is low, however,  $E_{\omega_{AC}}$  approaches 0.5, indicating  $A$  holds a neutral opinion on  $C$  because  $A$  does not trust  $B$  at all.

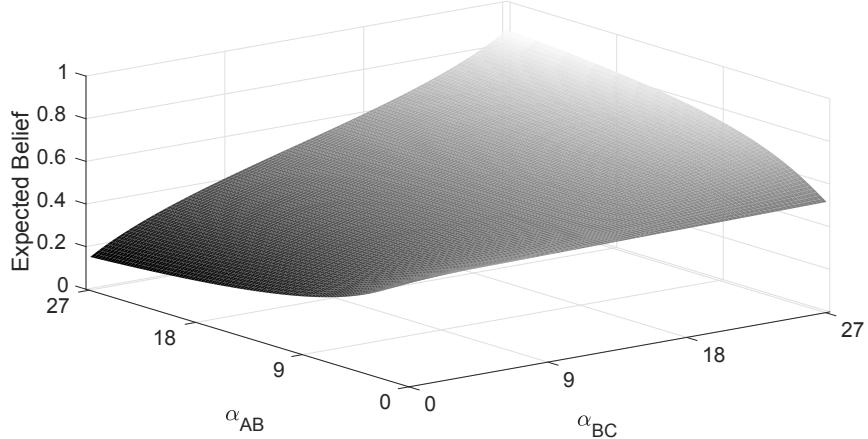


Figure 6.1: Influence of belief on discounting operation.

To further understand the impact of the evidence values, we vary  $\alpha_{AB}/\lambda_{AB}$  from 0 to 1, and change  $\lambda_{AB}$  from 0 to 300. In the experiment, we keep the original opinion  $\omega_{BC}$  unchanged, i.e.,  $\omega_{BC} = \langle 25, 5, 0 \rangle$ .

As we can see in Fig. 6.2, when  $\lambda_{AB}$  is large,  $E_{\omega_{AC}}$  increases as  $\alpha_{AB}/\lambda_{AB}$  increases. When  $\lambda_{AB}$  is small,  $E_{\omega_{AC}}$  is very close to 0.5. This phenomena indicates that when  $A$  is more certain about her opinion on  $B$ , i.e., larger  $\lambda_{AB}$ , she relies more on  $B$  to form her own opinion on  $C$ . Otherwise,  $A$ 's opinion on  $C$  tends to be neutral.

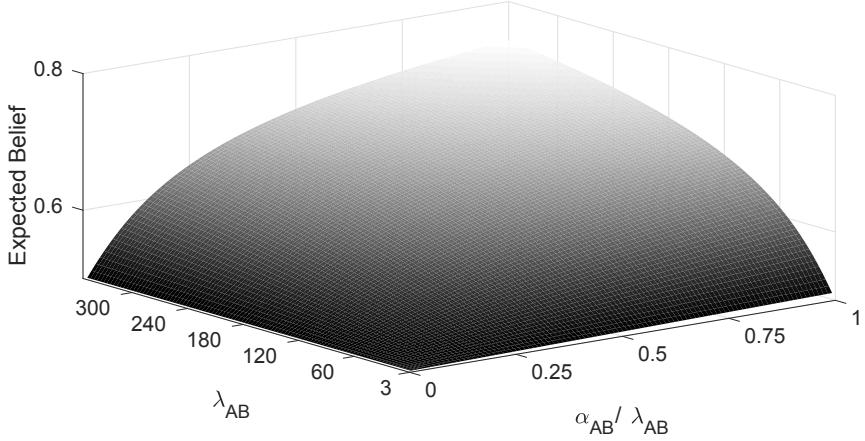


Figure 6.2: Influence of belief and uncertainty on the discounting operation.

### Combining Operation

Here, we assume two opinions  $\omega_{A_1C_1}$  and  $\omega_{A_2C_2}$  are combined to yield another opinion  $\omega_{AC} = \Theta(\omega_{A_1C_1}, \omega_{A_2C_2})$ . We vary  $\lambda_{A_1C_1}$  of opinion  $\omega_{A_1C_1}$  from 0 to 300, and keep the value of  $\alpha_{A_1C_1}/\beta_{A_1C_1} = 1/7$ . As such, we can denote opinion  $\omega_{A_1C_1}$  as  $\langle 0.125 \times \lambda_{A_1C_1}, 0.875 \times \lambda_{A_1C_1}, 0 \rangle$  (low trust). We set the second opinion  $\omega_{A_2C_2}$  as  $\langle 25, 5, 0 \rangle$  (high trust).

As shown in Fig. 6.3, when  $\lambda_{A_1C_1}$  is very large, the expected belief of the combined opinion  $\omega_{AC}$  approaches  $E_{\omega_{A_1C_1}}$ . When  $\lambda_{A_1C_1}$  is smaller,  $E_{\omega_{AC}}$  gets close to  $E_{\omega_{A_2C_2}}$ . It implies that a combining opinion yields a result similar to the opinion with a larger  $\lambda$ .

We further evaluate the combining operation by setting  $\omega_{A_2C_2}$  as  $\langle 25, 5, 0 \rangle$ ,  $\lambda_{A_1C_1}$  as 30, and varying  $\alpha_{A_2C_2}$  from 0 to 27. As shown in Fig. 6.4, when  $\alpha_{A_2C_2}$  and  $\alpha_{A_1C_1}$  are similar, the expected belief of the resulted  $\omega_{AC}$  is close to but higher than either  $E_{\omega_{A_1C_1}}$  or  $E_{\omega_{A_2C_2}}$ . When  $\alpha_{A_2C_2}$  and  $\alpha_{A_1C_1}$  are different,  $E_{\omega_{AC}}$  is close to the average of  $E_{\omega_{A_1C_1}}$  and  $E_{\omega_{A_2C_2}}$ .

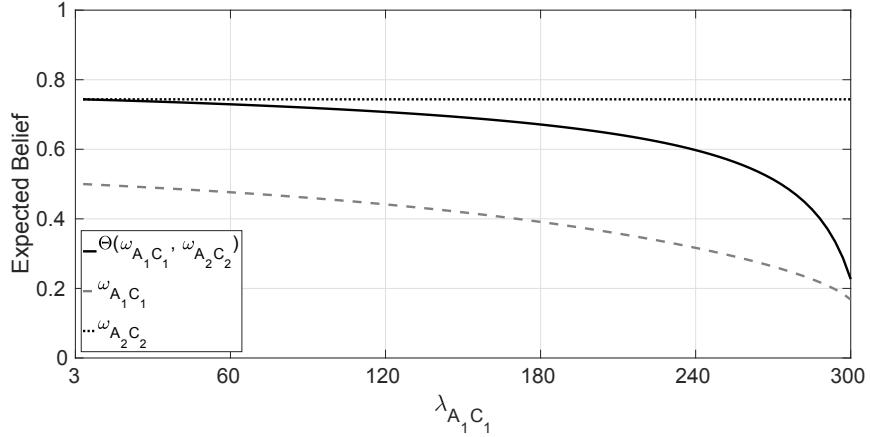


Figure 6.3: Influence of total evidence value  $\lambda$  on combining operation.

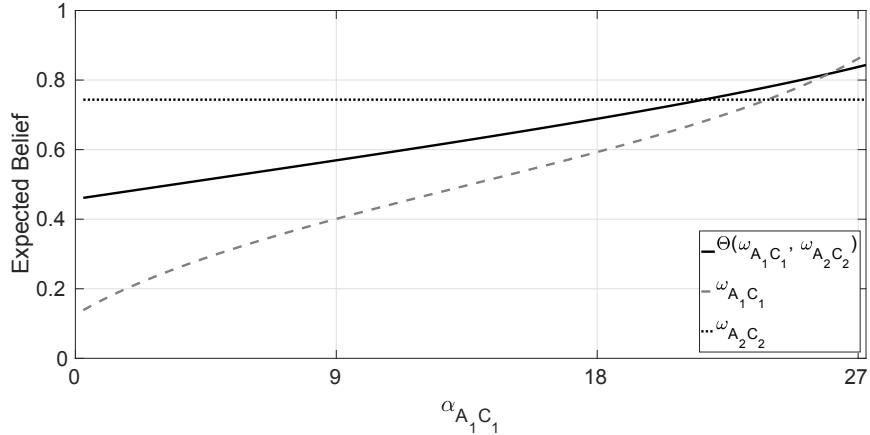


Figure 6.4: Influence of positive evidence value  $\alpha$  on combining operation.

We conclude that combining two opinions with similar positive evidence values will enhance the original opinion, due to the increased amount of positive evidence. On the other hand, combining opinions with the different amounts of positive evidence yields a neutralized result.

### Bridge Topology

Because the subjective logic model cannot handle the bridge topology, as shown in Fig. 4.5(a), an approximation solution is proposed by removing some edges from

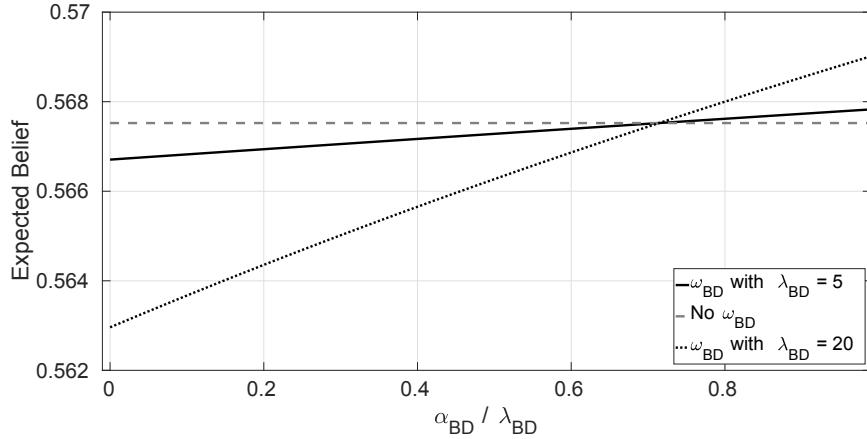


Figure 6.5: Influence of bridge opinion's positive/total evidence ratio ( $\alpha_{BD}/\lambda_{BD}$ ).

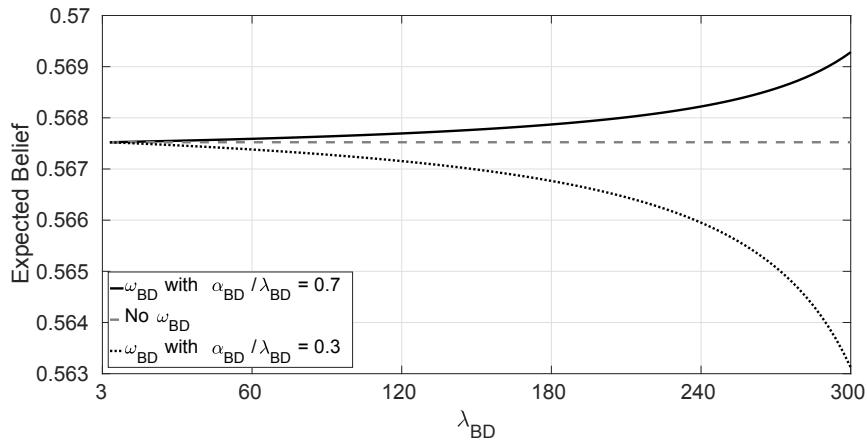


Figure 6.6: Influence of bridge opinion's total evidence value ( $\lambda_{BD}$ ).

the network, e.g.,  $\omega_{BD}$ . If  $\omega_{BD}$  is removed, the bridge topology becomes a parallel topology where  $A$  connects to  $C$  via two parallel paths  $A \rightarrow B \rightarrow C$  and  $A \rightarrow D \rightarrow C$ . We define  $\omega_{BD}$  as the bridge edge (or bridge opinion) and discover that the bridge edge could be very important in trust assessment and cannot be simply removed.

In the experiments, we set the opinions  $\omega_{AB}$ ,  $\omega_{AD}$ ,  $\omega_{BC}$  and  $\omega_{CD}$  as  $\langle 25, 5, 0 \rangle$ ,  $\langle 18, 12, 0 \rangle$ ,  $\langle 18, 12, 0 \rangle$ , and  $\langle 18, 12, 0 \rangle$ , respectively. We consider the cases where  $\omega_{BD}$  contains a small amount of evidence, e.g.,  $\lambda_{BD} = 5$ , and a large amount of evidence, e.g.,  $\lambda_{BD} = 20$ . By changing the value of  $\alpha_{BD}/\lambda_{BD}$  from 0 to 1, we obtain Fig. 6.5.

When  $\lambda_{BD} = 5$ , the expected belief of  $\omega_{AC}$  computed from 3VSL is similar to that computed by removing  $\omega_{BD}$ . When  $\lambda_{BD} = 20$ , however, 3VSL yields a different result. Moreover, we note that a more trustworthy bridge opinion, e.g.,  $\alpha_{BD}/\lambda_{BD} > 0.7$ , tends to yield a more trustworthy  $\omega_{AC}$ .

Furthermore, we vary  $\lambda_{BD}$  from 3 to 300, and set  $\alpha_{BD}/\lambda_{BD}$  as 0.7 and  $\alpha_{BD}/\lambda_{BD}$  as 0.3, respectively. As shown in Fig. 6.6, when  $\lambda_{BD}$  is large, the impact of the bridge opinion cannot be ignored. When  $\lambda_{BD}$  is small, a similar result can be obtained if the bridge opinion is removed, i.e., the approximation solution only works in this case.

### Survey Experiments

In addition to numerical analysis, we designed an online system to collect trust data. We use the collected dataset to validate the discounting and combining operations defined in 3VSL.

#### Setup of the Survey Experiments

More than 100 participates were invited to evaluate the trustworthiness of their 1-hop and 2-hop friends by answering a questionnaire proposed in [48]. The questionnaire consisted of 12 questions, and the answers of these questions were used to construct a participant’s opinions of his/her friends. The answer  $X$  for each question is scaled in 9 levels, where 8 represents “strongly trust” and 0 as “strongly distrust”. In addition, we add another question to let a participant indicate how certain they believe their answers are. The uncertainty score  $Y$  is scaled in 5 levels, where 0 represents “not sure at all” and 4 represents “very confident”.

After a participant logs into the online system, she will be asked to identify and evaluate the trustworthiness of her two direct friends  $B$  and  $D$ .  $A$  is then told that her friend  $B$  trusts  $C$  with an opinion  $\omega_{BC}$ , and she is asked to evaluate her opinion

on  $C$ . Finally,  $A$  is told that  $D$  also trusts  $C$ , with an opinion  $\omega_{DC}$ , and she is asked to evaluate  $C$ 's trustworthiness again, considering both the opinions from  $B$  and  $D$ . More details about the online system can be found at [33].

From the collected data, we construct trust opinions as follows. The average score of  $X$ , denoted as  $T$ , reflects the value of positive evidence. The uncertainty score  $Y$  is used to compute the total number of evidence  $\lambda$ , i.e. the higher the value of  $Y$ , the smaller the amount of evidence. It's difficult to obtain the accurate value of  $\lambda$  because participants may not recall the exact amount of evidence they used to make their judgments. Here, we assume 30 recent evidence indicators are good enough for a person to form an opinion, i.e.,  $\lambda$  of an opinion is 3 if  $Y = 0$  and 30, otherwise. Given the values of  $T$  and  $\lambda$ , an opinion vector is formed, according to the following equation.

$$\langle \alpha, \beta, \gamma \rangle = \langle T \cdot \lambda_{AX}, (1 - T) \cdot \lambda_{AX}, 0 \rangle, \quad (6.1)$$

where

$$T = \frac{\alpha_{AX}}{\lambda_{AX}}.$$

Notice that we assume the initial amount of neutral evidence is 0. This is because when a person makes decisions, neutral evidence is usually ignored, i.e., only positive and negative evidence are considered. In 3VSL, uncertainty only occurs when trust propagates within the network.

### Errors in Discounting and Combining Operations

We compute the errors in the discounting and combining operations as follows.

$$\begin{aligned} Err_{\Delta} &= \left| \frac{E_{\Delta(\omega_{AB}, \omega_{BC})} - E_{\omega_{A \rightarrow B \rightarrow C}}}{E_{\omega_{A \rightarrow B \rightarrow C}}} \right|, \\ Err_{\Theta} &= \left| \frac{E_{\Theta(\Delta(\omega_{AB}, \omega_{BC}), \Delta(\omega_{AD}, \omega_{DC}))} - E_{\omega_{A \rightarrow B, D \rightarrow C}}}{E_{\omega_{A \rightarrow B, D \rightarrow C}}} \right|. \end{aligned}$$

where  $\omega_{A \rightarrow B \rightarrow C}$  denotes  $A$ 's opinion on  $C$  based on  $B$ 's opinion, and  $\omega_{A \rightarrow B, D \rightarrow C}$  denotes  $A$ 's opinion on  $C$  based upon both  $B$ 's and  $D$ 's opinions. The errors  $Err_{\Delta}$  and  $Err_{\Theta}$  generated by the discounting and combining operations are shown in Fig 6.7 and 6.8, respectively.

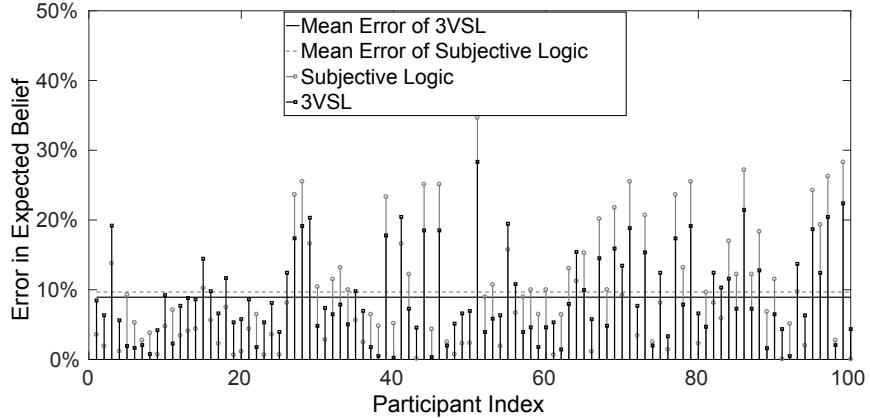


Figure 6.7: Absolute errors in expected belief of the discounting operation.

In these figures, we also plot the expected beliefs computed by the subjective logic. The average error of 3VSL is less than that of subjective logic i.e. 3VSL is a more accurate model in modeling trust in social networks.

We further plot the corresponding CDFs of the errors in Fig 6.9 and 6.10. For the combining operation, we can see around 90% of the results have errors less than 20%. For the discounting operation, 82% of the results have errors less than 20%. We also note that the number of accurate results, with an error  $< 20\%$ , computed

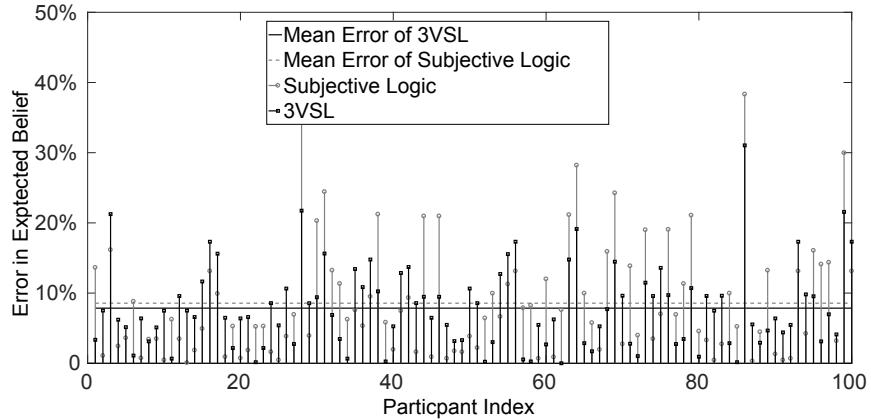


Figure 6.8: Absolute errors in expected belief of the combining operation.

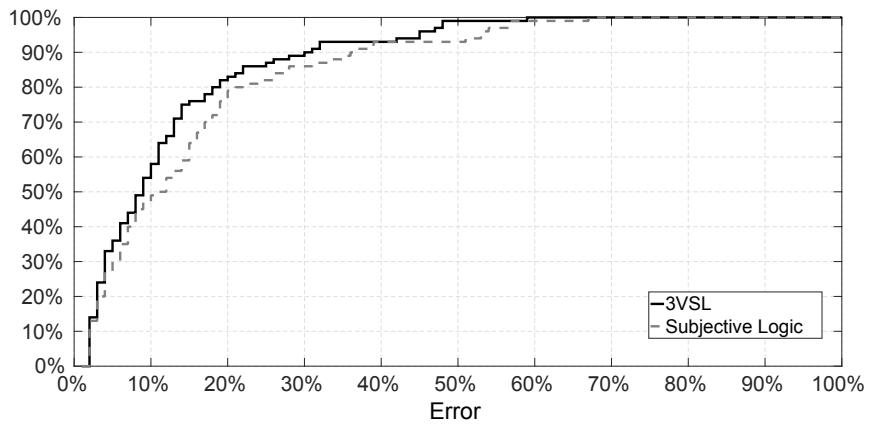


Figure 6.9: CDFs of errors in expected belief of the discounting operation.

by subjective logic is smaller. This indicates that 3VSL provides higher accuracy in trust assessments in OSNs.

### Experimental Evaluations

In this section, we will validate the 3VSL model and evaluate the performance of the AssessTrust and OpinionWalk algorithms using two real-world datasets: Advogato and PGP. To understand how accurate various models are in assessing trust within OSNs, we adopt F1 score [1] as the evaluating metric. The F1 score is

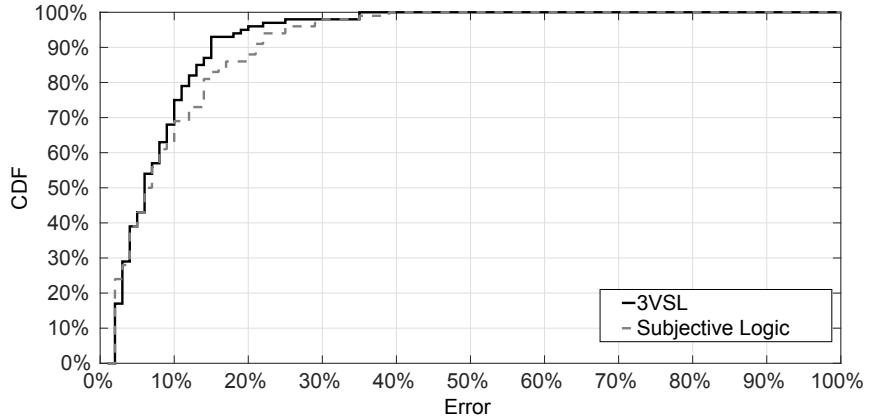


Figure 6.10: CDFs of errors in expected belief of the combining operation.

chosen because it is a comprehensive measure for different models in predicting or inferring trust [1].

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. In the end, we implement the OpinionWalk algorithm and compare its execution time to the benchmark algorithms.

### 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 6.1.

Table 6.1: 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

### 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  $\lambda$  as 10, 20, 30, 40, and 50. Given a certain  $\lambda$ , 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  $\lambda$ , 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 [80] 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  $\lambda$ 's and five different  $(\frac{\alpha}{\lambda})_0$ 's, we have a total of 25 combinations of parameters.

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

$$\left\langle \alpha, 0, \lambda \left(1 - \frac{\alpha}{\lambda}\right) \right\rangle.$$

Similar to Advogato, an opinion in PGP is determined by  $\lambda$  and  $\frac{\alpha}{\lambda}$ . We use the same transformation method to convert the trust relations in PGP into opinions.

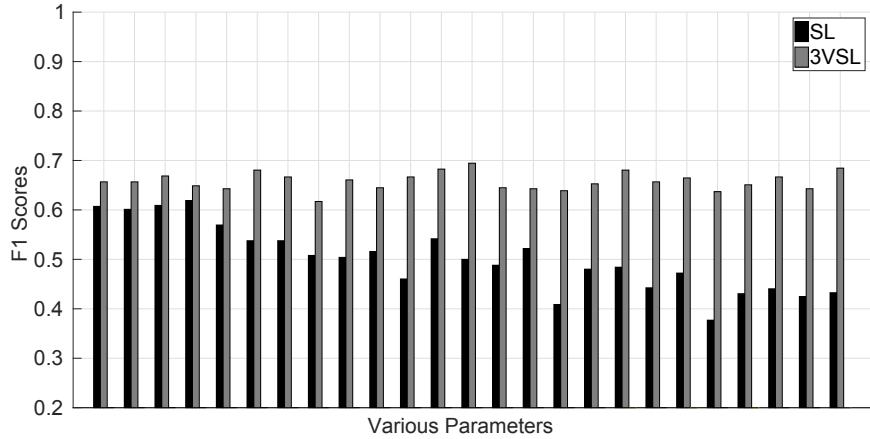


Figure 6.11: F1 scores of 3VSL and SL using the Advogato dataset. Parameters are the combinations between base trust levels (0.1, 0.2, 0.3, 0.4 and 0.5) and total evidence values (10, 20, 30, 40, and 50).

### 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

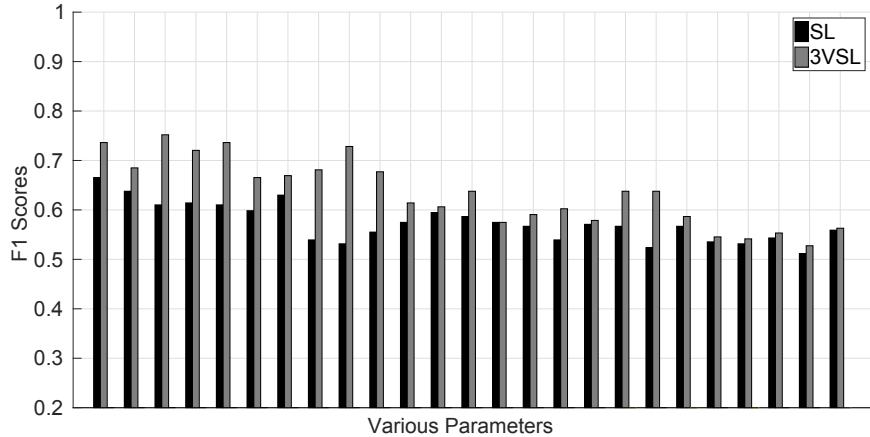


Figure 6.12: F1 scores of 3VSL and SL using the PGP dataset. Parameters are the combinations between base trust levels (0.1, 0.2, 0.3, 0.4 and 0.5) and total evidence values (10, 20, 30, 40, and 50).

$u$  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. 6.11 and 6.12, 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. 6.13. As shown in Fig. 6.13, the values of  $(\alpha + \beta)$  in

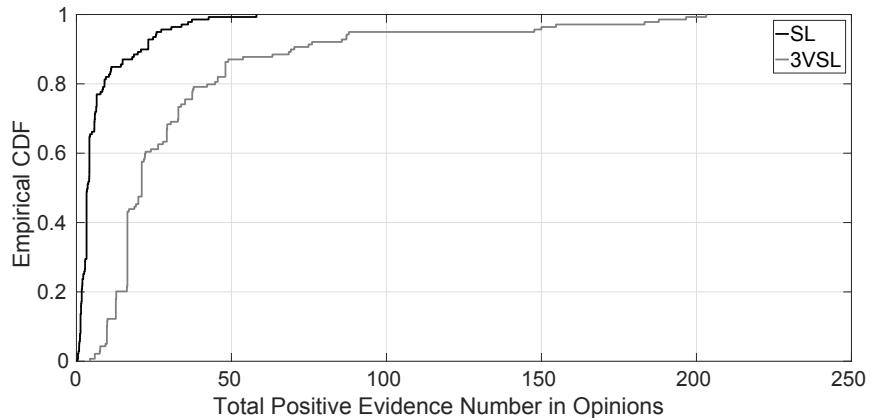


Figure 6.13: CDFs of  $\alpha + \beta$  in opinions computed by 3VSL and subjective logic using the Advogato dataset.

the opinions computed by SL are much lower than that of 3VSL. It results in an lack of evidence in computing the expected beliefs of opinions by SL. This observation

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

Table 6.2: 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.

matches the example introduced in Fig. 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.

#### Performance of the AssessTrust Algorithm

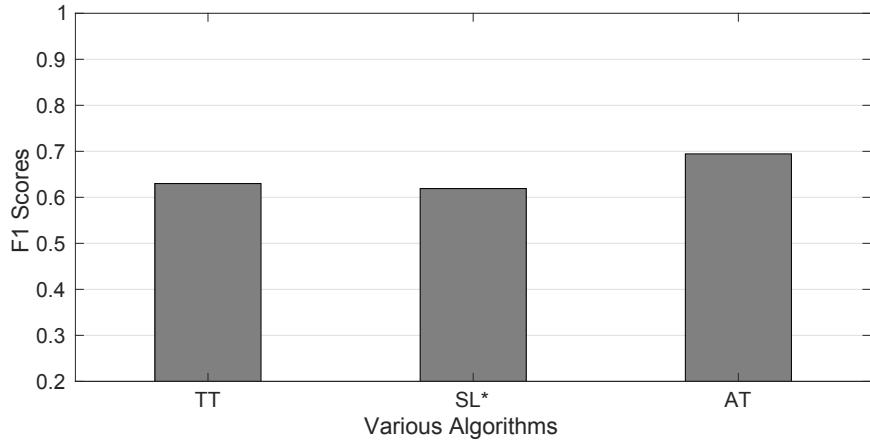


Figure 6.14: F1 scores of the trust assessment results generated by TT, SL\* and AT using the Advogato dataset.

After validating the 3VSL model, we study the performance of the AT algorithm and compare it to other benchmark algorithms, including TidalTrust (TT) [32], TrustRank (TR) [36] and EigenTrust (ET) [58]. TidalTrust is designed to compute the absolute trust of any user in an OSN. However, TR and ET are used to rank

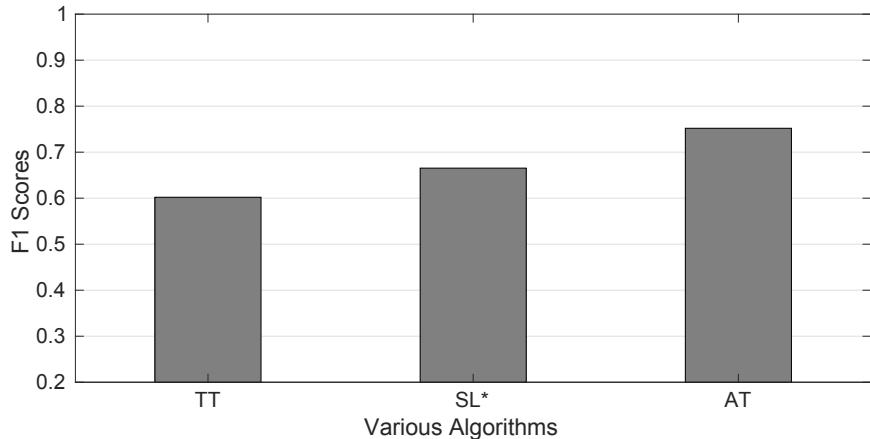


Figure 6.15: F1 scores of the trust assessment results generated by TT, SL\* and AT using the PGP dataset.

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 performance 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\*.

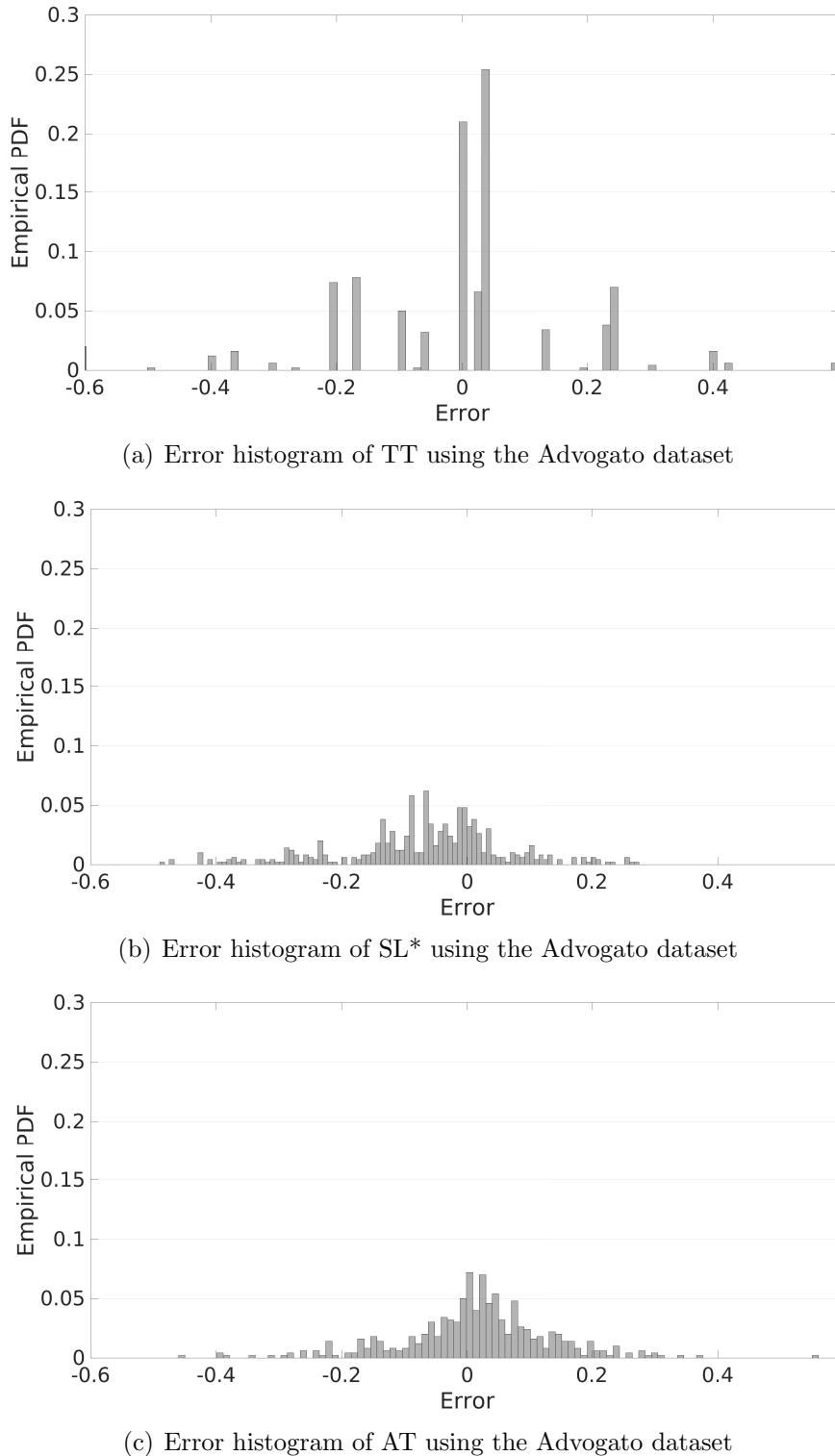
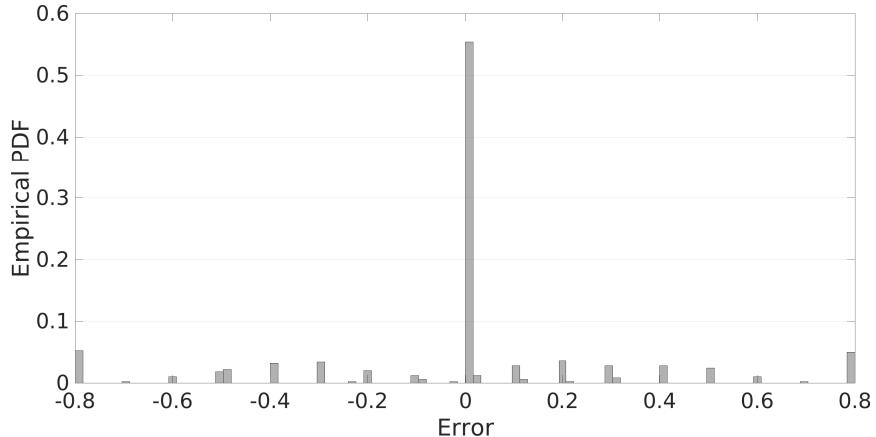
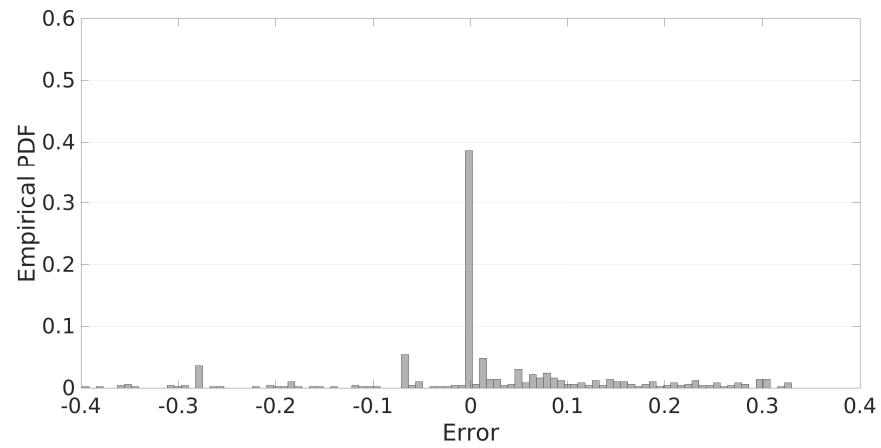


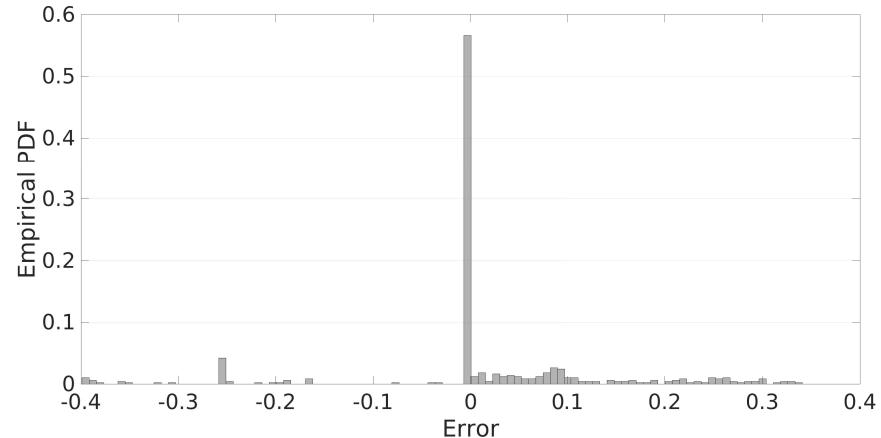
Figure 6.16: Histogram of the errors generated by TT, SL\* and AT using the Advogato dataset.



(a) Error histogram of TT using the PGP dataset



(b) Error histogram of SL\* using the PGP dataset



(c) Error histogram of AT using the PGP dataset

Figure 6.17: Histogram of the errors generated by TT, SL\* and AT using the PGP dataset.

The parameter settings for different algorithms in different datasets are shown in Table 6.2.

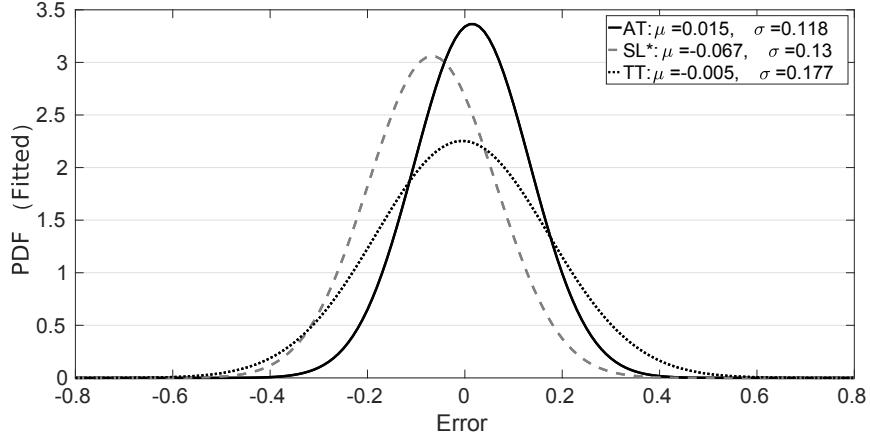


Figure 6.18: Fitted curves of the error distributions of TT, SL\* and AT using the Advogato dataset.

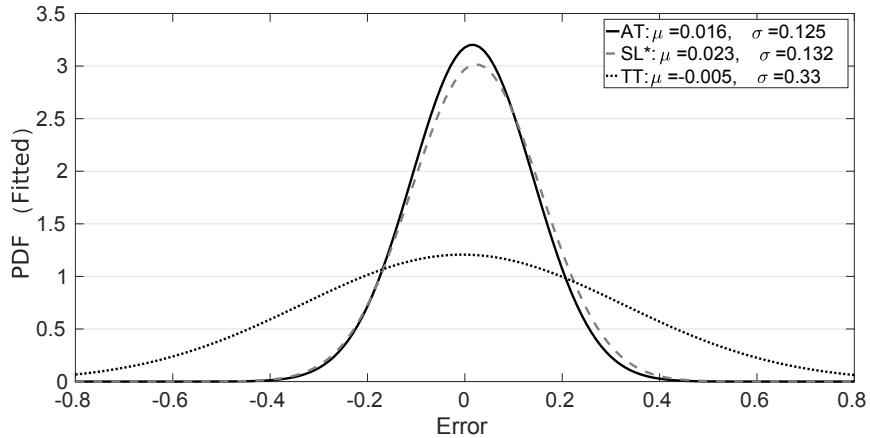


Figure 6.19: Fitted curves of the error distributions of TT, SL\* and AT using the PGP dataset.

We first look at the F1 scores of the trust assessment results generated by the three algorithms. The F1 scores are plotted in Figs. 6.14 and 6.15. As shown in Figs. 6.14 and 6.15, 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.

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. 6.16, 6.18, 6.17 and 6.19.

From Fig. 6.16(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, as shown in Figs. 6.16(b) and 6.16(c). If the PGP dataset is used, we observe the same phenomena, as shown in Figs. 6.17(a), 6.17(b) and 6.17(c).

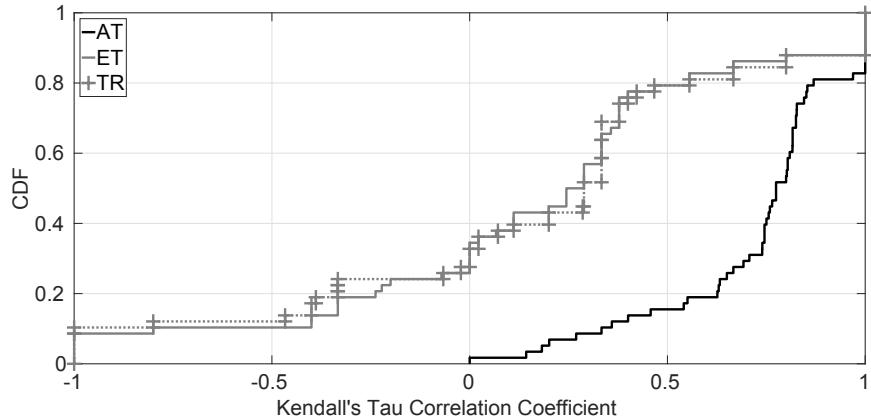


Figure 6.20: The CDFs of Kendall's tau ranking correlation coefficients of different algorithms using the Advogato dataset.

We further fit this histogram data using the Normal Distribution. As shown in Figs 6.18 and 6.19, 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 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

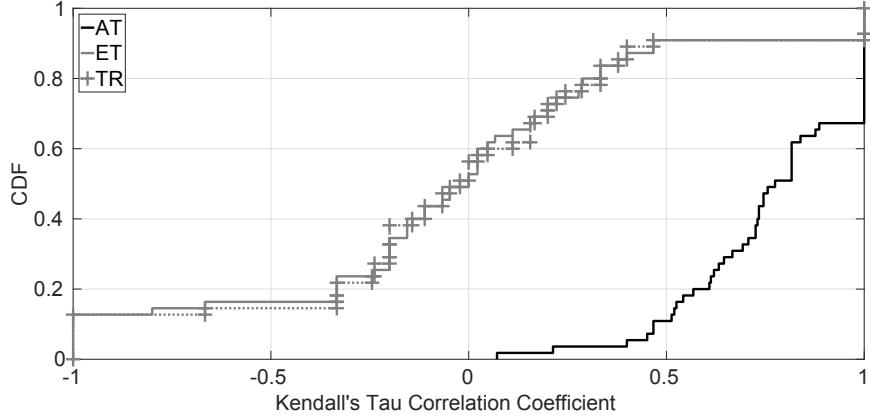


Figure 6.21: The CDFs of Kendall's tau ranking correlation coefficients of different algorithms using the PGP dataset.

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  $\omega_{uv}$ '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

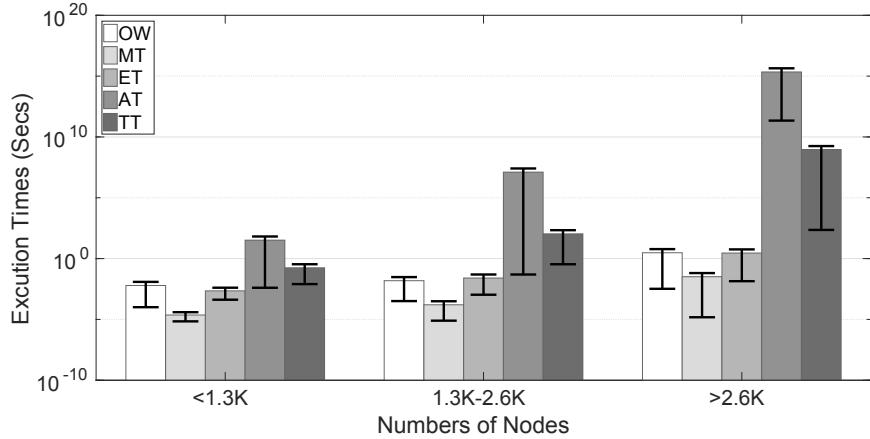


Figure 6.22: Execution times of different algorithms (OW, MT, ET, AT, and TT) using the Advogato dataset.

and the ground truth. We repeat each experiment 100 times in Advogato and PGP to get statistically significant results.

In Figs. 6.20 and 6.21, 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.

#### Performance of the OpinionWalk Algorithm

In this section, we evaluate the performance of the OpinionWalk (OW) algorithm, when it is used to address the MTA problem. Because OpinionWalk is designed based upon the 3VSL model, its accuracy in assessing trust does not need to

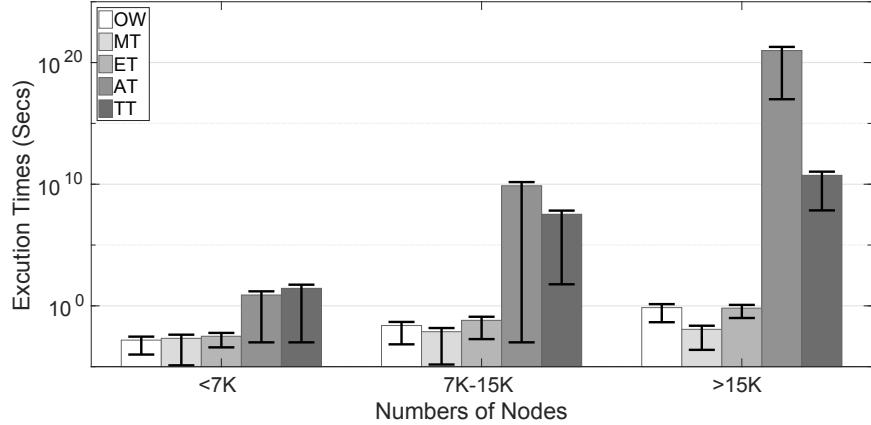


Figure 6.23: Execution times of different algorithms (OW, MT, ET, AT, and TT) using the PGP dataset.

be evaluated again. Therefore, we are only interested in its execution time. Here, we compare OpinionWalk to other benchmark algorithms, including EigenTrust (ET) ( $\alpha = 0.85$ ), TidalTrust (TT), MoleTrust (MT), and AssessTrust (AT). We do not consider TrustRank here because its running time is almost the same as EigenTrust's.

The experiments are conducted as follows. First, we randomly choose a trustor and its neighbors (trustees) within various hops. We run the above-mentioned algorithms on the sub-graph containing only the trustor and trustees. We plot the execution times of various algorithms, with respect to the number of nodes in each sub-graph. We group the node numbers into 3 categories. For Advogato, the node number categories are  $< 1.3K$ ,  $1.3K - 2.6K$  and  $> 2.6K$ . For PGP, the node number categories are  $< 7K$ ,  $7K - 15K$  and  $> 15K$ .

As shown in Fig. 6.22 and Fig. 6.23, TT and AT run much slower than OW when the numbers of nodes increase in both datasets. Particularly, when the number of nodes is large, corresponding to a larger searching depth, TT and AT are extremely slow. The reason is that both TT and AT need to be executed  $n$  times to solve the MTA problem where  $n$  is the number of nodes in networks. AT is the slowest algorithm

because it is a recursive algorithm and it keeps re-solving the same sub-problems over and over again. The execution times of OW, ET and MT are on the same order of magnitude, i.e.,  $O(n^3)$ . Because the discounting and combining operations in OW are more complicated than the multiplication and summation operations in MT, it runs slightly slower than MT. Considering both trust assessment accuracy and execution time, we conclude that OW is a better solution to MTA in OSNs.

## CONCLUSION

In this dissertation, 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. AT is designed for one-to-one trust assessment and is inefficient in addressing the MTA problem in OSNs. To solve the MTA problem, we design the OpinionWalk algorithm based on the 3VSL model. We prove that OpinionWalk is an equivalent implementation of the AT algorithm, yet offers a better time complexity of  $O(n^3)$  in addressing the MTA problem.

We validate 3VSL both in numerical and 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. In the end, experimental results show that OpinionWalk is an accurate, as well as fast, solution to the MTA problem.

## REFERENCES CITED

- [1] f1 score. [http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score).
- [2] facebook and lendingclub looks like its working. <http://blog.lendingclub.com/facebook-and-lending-club-looks-like-its-working/>.
- [3] facebook's patent may change your credit. <http://www.techinsider.io/facebook-s-patent-may-change-your-credit-2015-8>.
- [4] T. Ahn and J. Esarey. A dynamic model of generalized social trust. *Journal of Theoretical Politics*, 20(2):151–180, 2008.
- [5] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. Sok: The evolution of sybil defense via social networks. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 382–396, May 2013.
- [6] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: An axiomatic approach. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 199–208, New York, NY, USA, 2008. ACM.
- [7] R. Andersen, F. Chung, and K. Lang. Local partitioning for directed graphs using pagerank. In A. Bonato and F. Chung, editors, *Algorithms and Models for the Web-Graph*, volume 4863 of *Lecture Notes in Computer Science*, pages 166–178. Springer Berlin Heidelberg, 2007.
- [8] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58 – 71, 2007. Software Engineering and the Semantic Web.
- [9] A. Basu, J. Vaidya, J. C. Corena, S. Kiyomoto, S. Marsh, G. Guo, J. Zhang, and Y. Miyake. Opinions of people: Factoring in privacy and trust. *SIGAPP Appl. Comput. Rev.*, 14(3):7–21, Sept. 2014.
- [10] C. Bizer, R. Cyganiak, T. Gauss, and O. Maresch. O.: The triql.p browser: Filtering information using context-, content- and rating-based trust policies. In *Proceedings of the Semantic Web and Policy Workshop, held in conjunction with the 4th International Semantic Web Conference, 7 November, 2005*, pages 12–20, 2005.
- [11] C. Borgs, J. Chayes, A. T. Kalai, A. Malekian, and M. Tennenholtz. *A Novel Approach to Propagating Distrust*, pages 87–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

- [12] C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. Exploiting redundancy in question answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01*, pages 358–365, New York, NY, USA, 2001. ACM.
- [13] G. Danezis and P. 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.
- [14] Z. Despotovic and K. Aberer. Probabilistic prediction of peers' performance in {P2P} networks. *Engineering Applications of Artificial Intelligence*, 18(7):771 – 780, 2005.
- [15] L. Ding, P. Kolari, T. Finin, A. Joshi, Y. Peng, and Y. Yesha. On homeland security and the semantic web: A provenance and trust aware inference framework. In *In AAAI Spring Symposium on AI Technologies for Homeland Security*, pages 21–23. AAAI Press, 2005.
- [16] L. Ding, L. Zhou, and T. Finin. Trust based knowledge outsourcing for semantic web agents. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, WI '03*, pages 379–, Washington, DC, USA, 2003. IEEE Computer Society.
- [17] P. M. Doney and J. P. Cannon. An examination of the nature of trust in buyer-seller relationships. *the Journal of Marketing*, pages 35–51, 1997.
- [18] D. Downey, O. Etzioni, and S. Soderland. A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05*, pages 1034–1041, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.
- [19] 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.
- [20] E. ElSalamouny, V. Sassone, and M. Nielsen. HMM-based trust model. In *Formal Aspects in Security and Trust*, pages 21–35. Springer, 2010.
- [21] R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. In *Trust and deception in virtual societies*, pages 55–90. Springer, 2001.
- [22] C. J. Fung, J. Zhang, I. Aib, and R. Boutaba. Dirichlet-based trust management for effective collaborative intrusion detection networks. *Network and Service Management, IEEE Transactions on*, 8(2):79–91, 2011.

- [23] D. Gambetta. *Trust: Making and Breaking Cooperative Relations*, volume 52. Blackwell, 1988.
- [24] S. Ganesan. Determinants of long-term orientation in buyer-seller relationships. *the Journal of Marketing*, pages 1–19, 1994.
- [25] H. Gao, Y. Yang, K. Bu, Y. Chen, D. Downey, K. Lee, and A. Choudhary. Spam ain't as diverse as it seems: Throttling osn spam with templates underneath. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, pages 76–85, New York, NY, USA, 2014. ACM.
- [26] D. Gefen, E. Karahanna, and D. W. Straub. Trust and tam in online shopping: An integrated model. *MIS Q.*, 27(1):51–90, Mar. 2003.
- [27] Y. Gil and V. Ratnakar. Trusting information sources one citizen at a time. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 162–176, London, UK, UK, 2002. Springer-Verlag.
- [28] J. Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Trans. Web*, 3(4):12:1–12:33, Sept. 2009.
- [29] J. Golbeck and J. 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.
- [30] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. In M. Klusch, A. Omicini, S. Ossowski, and H. Laamanen, editors, *Cooperative Information Agents VII*, volume 2782 of *Lecture Notes in Computer Science*, pages 238–249. Springer Berlin Heidelberg, 2003.
- [31] J. A. Golbeck. Computing and applying trust in web-based social networks. 2005.
- [32] J. A. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, USA, 2005. AAI3178583.
- [33] X. L. Guangchi Liu. Online Trust Survey. <http://www.cs.montana.edu/yang/trust-us>, 2013.
- [34] R. Guha, R. Kumar, P. Raghavan, and A. 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.
- [35] R. Guha, R. Kumar, P. Raghavan, and A. 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] Z. Gyöngyi, H. Garcia-Molina, and J. 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.
- [37] C.-W. Hang and M. 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.
- [38] C.-W. Hang, Y. Wang, and M. 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.
- [39] C.-W. Hang, Y. Wang, and M. 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.
- [40] X. Hong, D. Huang, M. Gerla, and Z. Cao. Sat: Situation-aware trust architecture for vehicular networks. In *Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '08, pages 31–36, 2008.
- [41] L. T. Hosmer. Trust: The connecting link between organizational theory and philosophical ethics. *Academy of management Review*, 20(2):379–403, 1995.
- [42] X. Hu, J. Tang, Y. Zhang, and H. Liu. Social spammer detection in microblogging. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2633–2639. AAAI Press, 2013.
- [43] J. F. Hübner, E. Lorini, A. Herzig, and L. 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.
- [44] A. Jakoby, M. Liskiewicz, and R. Reischuk. Space efficient algorithms for series-parallel graphs. In *STACS 2001*, pages 339–352. Springer, 2001.
- [45] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.

- [46] S. L. Jarvenpaa, N. Tractinsky, and L. Saarinen. Consumer trust in an internet store: a cross-cultural validation. *Journal of Computer-Mediated Communication*, 5(2):0–0, 1999.
- [47] W. Jiang, J. Wu, G. Wang, and H. 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.
- [48] C. Johnson-George and W. C. Swap. Measurement of specific interpersonal trust: Construction and validation of a scale to assess trust in a specific other. *Journal of Personality and Social Psychology*, 43(6):1306–1317, 1982.
- [49] A. Jøsang. The consensus operator for combining beliefs. *Artificial Intelligence*, 141(1):157–170, 2002.
- [50] A. 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.
- [51] A. Josang. Conditional reasoning with subjective logic. *Journal of Multiple-Valued Logic and Soft Computing*, 15(1):5–38, 2008.
- [52] 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.
- [53] A. Jøsang, R. Hayward, and S. 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.
- [54] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer, 2006.
- [55] A. Jøsang and D. McAnally. Multiplication and comultiplication of beliefs. *International Journal of Approximate Reasoning*, 38(1):19–51, 2005.
- [56] A. Jøsang and S. 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.
- [57] A. JØSANG. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 09(03):279–311, 2001.

- [58] S. D. Kamvar, M. T. Schlosser, and H. 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.
- [59] S. D. Kamvar, M. T. Schlosser, and H. 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.
- [60] T. H.-J. Kim, P. Gupta, J. Han, E. Owusu, J. Hong, A. Perrig, and D. 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.
- [61] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.
- [62] U. Kuter and J. 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.
- [63] U. Kuter and J. 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.
- [64] J. Li, R. Li, and J. Kato. Future trust management framework for mobile ad hoc networks. *Communications Magazine, IEEE*, 46(4):108–114, April 2008.
- [65] Y. Li, B. Q. Zhao, and J. Lui. On modeling product advertisement in large-scale online social networks. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1412–1425, 2012.
- [66] G. Liu, Q. Yang, H. Wang, X. Lin, and M. Wittie. Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic. In *INFOCOM, 2014 Proceedings IEEE*, pages 1698–1706, April 2014.
- [67] G. Liu, Q. Yang, H. Wang, S. 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.
- [68] X. Liu and A. Datta. Modeling context aware dynamic trust using hidden markov model. In *AAAI*, 2012.
- [69] P. Massa and P. 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.

- [70] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 17–24, 2007.
- [71] P. Massa and C. Hayes. Page-rerank: using trusted links to re-rank authority. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 614–617, Sept 2005.
- [72] P. Massa, M. Salvetti, and D. Tomasoni. Bowling alone and trust decline in social network sites. In *Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on*, pages 658–663, Dec 2009.
- [73] D. H. McKnight, V. Choudhury, and C. Kacmar. Developing and validating trust measures for e-commerce: An integrative typology. *Information systems research*, 13(3):334–359, 2002.
- [74] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM, 2011 Proceedings IEEE*, pages 1943–1951, April 2011.
- [75] A. Mohaisen, H. Tran, A. Chandra, and Y. Kim. Trustworthy distributed computing on social networks. *Services Computing, IEEE Transactions on*, 7(3):333–345, July 2014.
- [76] C. Moorman, G. Zaltman, and R. Deshpande. Relationships between providers and users of market research: The dynamics of trust. *Journal of marketing research*, 29(3):314–328, 1992.
- [77] F. Moyano, C. Fernandez-Gago, and J. Lopez. A framework for enabling trust requirements in social cloud applications. *Requirements Engineering*, 18:321–341, Nov 2013 2013.
- [78] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [79] R. D. Pietro, F. Lombardi, F. Martinelli, and D. Sgandurra. Anticheetah: Trustworthy computing in an outsourced (cheating) environment. *Future Generation Computer Systems*, 48(0):28 – 38, 2015. Special Section: Business and Industry Specific Cloud.
- [80] D. A. Powers and Y. Xie. *Statistical methods for categorical data analysis*. Emerald Group Publishing, 2008.
- [81] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

- [82] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 351–368. Springer Berlin Heidelberg, 2003.
- [83] E. Riloff, J. Wiebe, and W. Phillips. Exploiting subjectivity classification to improve information extraction. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, pages 1106–1111. AAAI Press, 2005.
- [84] D. M. Rousseau, S. B. Sitkin, R. S. Burt, and C. Camerer. Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3):393–404, 1998.
- [85] L. Shi, S. Yu, W. Lou, and Y. Hou. SybilShield: An agent-aided social network-based sybil defense among multiple communities. In *INFOCOM, 2013 Proceedings IEEE*, pages 1034–1042, 2013.
- [86] V. Stoyanov, C. Cardie, and J. Wiebe. Multi-perspective question answering using the opqa corpus. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 923–930, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [87] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao. Unik: Unsupervised social network spam detection. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 479–488, New York, NY, USA, 2013. ACM.
- [88] W. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical Bayesian modelling. *Artificial Intelligence*, 193(0):149 – 185, 2012.
- [89] S. 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.
- [90] G. Vogiatzis, I. MacGillivray, and M. 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.

- [91] Y. Wang, C.-W. Hang, and M. P. Singh. A probabilistic approach for maintaining trust based on evidence. *J. Artif. Int. Res.*, 40(1):221–267, Jan. 2011.
- [92] Y. Wang and M. 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.
- [93] Y. Wang and M. P. Singh. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 1551–1556, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [94] Y. Wang and M. P. Singh. Evidence-based trust: A mathematical model geared for multiagent systems. *ACM Trans. Auton. Adapt. Syst.*, 5(4):14:1–14:28, Nov. 2010.
- [95] W. Wei, F. Xu, C. Tan, and Q. Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1951–1959, March 2012.
- [96] L. Xiong and L. Liu. Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. *Knowledge and Data Engineering, IEEE Transactions on*, 16(7):843–857, July 2004.
- [97] C. Yang, R. Harkreader, and G. Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *Information Forensics and Security, IEEE Transactions on*, 8(8):1280–1293, Aug 2013.
- [98] D.-N. Yang, H.-J. Hung, W.-C. Lee, and W. 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, New York, NY, USA, 2013. ACM.
- [99] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social network defense against sybil attacks. *Networking, IEEE/ACM Transactions on*, 18(3):885–898, June 2010.
- [100] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *IEEE/ACM Trans. Netw.*, 16(3):576–589, June 2008.
- [101] J. Zhang, R. Zhang, Y. Zhang, and G. Yan. On the impact of social botnets for spam distribution and digital-influence manipulation. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 46–54, Oct 2013.

- [102] W. Zhang, S. Das, and Y. Liu. A trust based framework for secure data aggregation in wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 60–69, Sept 2006.
- [103] Y. Zhang, H. Chen, and Z. Wu. A social network-based trust model for the semantic web. In L. Yang, H. Jin, J. Ma, and T. Ungerer, editors, *Autonomic and Trusted Computing*, volume 4158 of *Lecture Notes in Computer Science*, pages 183–192. Springer Berlin Heidelberg, 2006.
- [104] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *Parallel and Distributed Systems, IEEE Transactions on*, 18(4):460–473, April 2007.
- [105] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 338–346, 2013.
- [106] X. Zhu and S. Gauch. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 288–295, New York, NY, USA, 2000. ACM.
- [107] C.-N. Ziegler and G. Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4):337–358, 2005.
- [108] J. Zou and F. Fekri. A belief propagation approach for detecting shilling attacks in collaborative filtering. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 1837–1840, New York, NY, USA, 2013. ACM.
- [109] Y. Zuo, W.-c. Hu, and T. O'Keefe. Trust computing for social networking. In *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 1534–1539. IEEE, 2009.