A

On Selecting Recommenders for Trust Evaluation in Online Social Networks

Wenjun Jiang, Hunan University
Jie Wu, Temple University
Guojun Wang, Central South University

Trust is a central component of social interactions among humans. Many applications motivate the consideration of trust evaluation in online social networks (OSNs). Some work has been proposed based on a trusted graph. However, it is still an open challenge to construct a trusted graph, especially in terms of selecting proper recommenders, which can be used to predict the trustworthiness of an unknown target efficiently and effectively. Based on the intuition that people who are close to and influential to us can make more proper and acceptable recommendations, we present the idea of recommendation-aware trust evaluation (RATE). We further model the recommender selection problem as an optimization problem, with the objectives of higher accuracy, lower risk (uncertainty), and lower cost. Four metrics: $trustworthiness$, $expertise$, $uncertainty$, and $cost$, are identified to measure and adjust the quality of recommenders. We focus on a 1-hop recommender selection, for which we propose the FluidTrust model to better illustrate the trust-decision making process of a user. We also discuss the extension of multi-hop scenarios and multi-target scenarios. Experimental results, with the real social network data sets of Epinions and Advogato, validate the effectiveness of RATE: it can predict trust with higher accuracy (it gains about 20% higher accuracy in Epinions), lower risk, and less cost (about a 30% improvement).

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: online social networks (OSNs), recommendation-aware, recommender selection, trust evaluation

ACM Reference Format:
Wenjun Jiang, Jie Wu, and Guojun Wang, 2015. On Selecting Recommenders for Trust Evaluation in Online Social Networks. *ACM Trans. Internet Technol.* V, N, Article A (January YYYY), 20 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

With the rapid development of Internet technology, online social networks (OSNs) are becoming more and more popular. OSNs are organized around users. Participating users join a network, publish their profile and any content, and create links to any other users with whom they associate. The resulting social network provides a basis for maintaining social relationships, for finding users with similar interests, and for locating content and knowledge that has been contributed or endorsed by other users.

This work is supported by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963; NSFC grants 61272151 and 61472451, ISTCP grant 2013DFB10070, the China Hunan Provincial Science & Technology Program under Grant Number 2012GK4106; and the Chinese Fundamental Research Funds for the Central Universities 531107040845.

DOI: http://dx.doi.org/10.1145/0000000.0000000

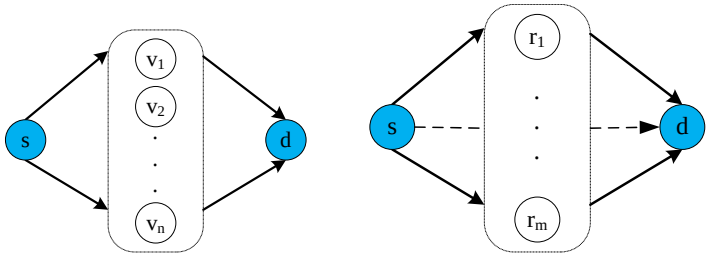


Fig. 1. (a) An example of a social network showing that s has many friends who can provide recommendations on t; (b) The objective graph, in which a subset of friends are selected as recommenders, $\{r_1, ..., r_m\} \subseteq \{u_1, ..., u_n\}$.

The trust mechanism is a tool that has been used to facilitate decision-making in diverse applications, ranging from ancient fish markets to state-of-the-art e-commerce. In OSNs, various applications have motivated the tremendous attention of trust evaluation, such as hiring managers who want to recruit new employees, service consumers who are looking for service providers, e-business buyers who are finding sellers, scientists who are searching for collaborators, etc. In other words, trust issues exist in any application whenever a person (e.g., source s) needs to estimate the trust level of another (e.g., target t), so as to decide whether or not to conduct further interactions.

1.1. The Motivation

A single person usually has a limit of known persons, due to his limited time and energy. A famous example is the Dunbar's number, or 150-rule [Dunbar 1992]: there is a suggested cognitive limit to the number of people with whom one can maintain stable social relationships. Hence, friends take an important role of making recommendations (Fig. 1). Several models have been proposed to estimate the trustworthiness of a given target from a source, taking advantage of the propagative (or weak-transitive) property of trust: if s trusts v, and v trusts d, then s can infer some trust on d. In the above process, v takes the role of a recommender, to help s make a proper estimation of d's trustworthiness. In this sense, it is similar to a referral system, which naturally captures the manner in which people help each other in finding trustworthy experts [Pushpa et al. 2010]. Many useful findings have been made. However, most of the existing trust models deal with the information aggregation in a small trusted graph, for which several challenges remain open:

(1) It is unclear which users should be selected as recommenders into the trusted graph. That is, "how to build the trusted graph, especially for selecting a proper subset of users."
(2) There is no ground truth on how much the real trust s falls on d, especially when s knows little about d.
(3) Research is lacking on how the trust evolution [Tang et al. 2012] will affect trust evaluation in future interactions.
(4) Existing models are solely based on trust ratings. Other closely related concepts in practical applications, e.g., a recommender's expertise (on the topic, target, and even the preference and bias of the source), the risk (uncertainty), and the possible cost, are usually overlooked.

In reality, it usually happens that a user has many friends; selecting different subsets of these friends may lead to making different decisions, taking different risks, and paying different costs. Therefore, we focus on exploring the factors that are involved in the process of trust evaluation, and developing an efficient scheme to solve the rec-

ommender subset selection problem, to meet the goals of higher prediction accuracy, lower risk (uncertainty), and less cost.

1.2. Main Ideas and Our Contributions

"It is not what you know, but who you know that makes the difference." Our basic idea is to find people who are the most suitable to serve as recommenders, as to help the source make a proper decision. That is, choosing the one who can enhance the "visibility" of the source. To make it simple, we are coping with "How do you find the users that reflect your tastes the most?" We try to apply the idea of recommendation [Massa and Avesani 2007a] to trust evaluation, based on the observation that people who are close and influential to us, and who have more expertise on the related topics and even the target, can make more proper and acceptable recommendations for us.

We propose a novel model to select proper neighbors, which we call recommenders, for evaluating a target's trustworthiness. Our goal is to develop a comprehensive model, which can tell what the trust level of a target is; more importantly, it can provide through whom the goal can be realized. Our contributions are as follows:

(1) We propose the idea of recommendation-aware trust evaluation (RATE for short). Since trust itself is usually personalized, it is natural to identify some proper recommenders who often have similar ideas or opinions as the source, and who have more expertise on the topic and the target, as to estimate the trustworthiness of the target.
(2) We identify four metrics, $trustworthiness$, $expertise$, $uncertainty$, and $cost$, to comprehensively measure and adjust the quality of recommenders. Those metrics can indicate the preference of a user (i.e., one's personal level of trust in a friend, who is a possible recommender), the degree to which a friend knows the source, the topic and the target, the historical behavior and fluctuation of his friend, and the cost and availability of his friend as a recommender, respectively.
(3) We focus on the 1-hop (i.e., the direct neighbor of source) recommender selection strategies and trust-decision making, for which we propose the FluidTrust model to better illustrate a user's trust-decision making process. We also analyze other more complex scenarios when it needs multi-hops for the source to reach the target, or when there are multiple targets.
(4) We evaluate RATE using two real trust networks, Epinions (www.epinions.com) and Advogato (www.advogato.org). The results demonstrate how each metric can impact the performance of RATE, and show that RATE can predict trust with higher accuracy (it gains about 20% higher accuracy in Epinions and 1.8% in Advogato), lower risk, and less cost (about a 30% improvement).

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 states the problem we address, and provides some preliminaries. Section 4 presents the overview of our approach. Sections 5 and 6 describe the 1-hop recommender selection scheme and possible extensions. Section 7 describes the experimental evaluation. Finally, Section 8 concludes this paper and suggests future work.

2. RELATED WORK

Generally speaking, trust models we are considering use a graph to reflect the relations between users, which is usually named *trusted graph*. A node in the graph represents a user in OSNs, and a directed edge represents the trust relation from a user to another, with its weight being the trust degree.

Golbeck proposed TidalTrust [Golbeck 2005], which can generate a recommendation regarding how much one person should trust the other, based on the shortest strongest trusted paths. Massa et al. [Massa and Avesani 2007b] proposed MoleTrust, which

uses a reduced directed acyclic graph for trust calculation. In both TidalTrust and MoleTrust, only one-dimensional information (i.e., the trust) is used.

Wang and Wu [Wang and Wu 2011] proposed a multi-dimensional evidence-based trust management system with multi-trusted paths. They proposed algorithms to simplify complex trusted graph. However, how to construct such a trusted graph is not mentioned. Our previous work in [Jiang et al. 2014] proposed SWTrust, to generate small trusted graphs for large OSNs. The information used to construct trust is users' active domains, which are stable and objective compared to subjective trust ratings. Liu et al. proposed two approaches in [Liu et al. 2010] and [Liu et al. 2013] to deal with the multi-constrained optimal trusted path selection problem. In their work, the trust, the social intimacy degree, and the role impact factor are considered to improve the quality of trust prediction. However, the uncertainty and cost are overlooked.

Some work considers the factor of uncertainty. Jøsang et al. [Jøsang et al. 2006] proposed a trust model with subjective logic, in which they use b, d, and u to represent belief, disbelief and uncertainty respectively, where $b, d, u \in [0, 1]$ and $b + d + u = 1$. The confidence of a trust value is equivalent to the certainty of the corresponding opinion. Liu et al. proposed a trust model using three-valued subjective logic [Liu et al. 2014]. They introduce the concepts of posteriori and priori uncertainty spaces.

Based on trusted graph, some trust models use the graph analogy-based approach. For instance, RN-Trust [Taherian et al. 2008] emulates a trusted graph with a resistive network. However, RN-Trust may not be suitable for large scale social networks, due to the complexity of calculating the equivalent resistance value. GFTrust [Jiang et al. 2015] calculates trust using generalized maximum flow, based on a trusted graph.

We can see there is still room for research on selecting proper recommender and considering multiple proper metrics. We have got some preliminary results in [Jiang et al. 2013], which briefly presents the idea of selecting recommenders. In the current paper, we provide more details for the metrics, concepts, and experiments. The new proposed FluidTrust model is incited by FluidRating, a rating prediction scheme proposed in [Jiang et al. 2014; Jiang et al. 2015]. In those works, we are trying to predict a user's opinion on a specific item. Meanwhile, FluidTrust in this paper is used to better illustrate the process of 1-hop recommender selection and trust-decision making, and the opinion is on a user, rather than on an item.

Similar problems have been studied in some other work [Yolum and Singh 2003; Yolum and Singh 2005; Liang and Shi 2008; Pushpa et al. 2010; Etuk et al. 2013]. [Yolum and Singh 2003] studies the properties of referral systems, and analyzes two elements of *expertise* and *sociability*. The former represents the quality of services a provider can provide, and the latter represents the quality of referrals a recommender can provide. In our paper, we focus on the recommender (referrals) and we identify four metrics to better measure the quality of a recommender. [Yolum and Singh 2005] further studies how to develop robust, self-organizing referral networks, and claims the flexible referrals are essential for locating trustworthy services: Consumers can help each other find desired service providers. [Liang and Shi 2008] studies two kinds uncertainties about the effects of ratings on building trust relationships: algorithm uncertainty and factor uncertainty. They find that ratings are not always helpful. [Pushpa et al. 2010] tries to find experts via referrals, in a co-author social network. They find that more than four neighbors can guarantee the number of experts found. [Etuk et al. 2013] describes a trust-based approach to information fusion, which exploits diversity among information sources. The aim is similar to our work, i.e., to select a small number of candidates to query for evidence.

3. PROBLEM FORMULATION

3.1. Recommender Selection Problem (RSP)

Definition 3.1. Recommender Selection Problem (RSP): Given a social network $G = (V, E)$, V is the set of nodes and E is the set of links (or edges). For two nodes, s and d in V, s is the source and d is the target. For the safety of user interactions in OSNs, we wonder how to design an efficient scheme to select the best recommenders $R = \{r_1, ..., r_m\}$, from the neighbor set of s $N_s = \{u_1, ..., u_n\}$ ($m \leq n$), with the goals of making a proper decision (to trust or not trust d), meeting the optimal requirements of higher accuracy, lower risk (uncertainty), and less cost.

According to the distance from recommenders to the source, the problem can be divided into two sub-issues, to cope with 1-hop neighbors (the direct neighbors of source) and multi-hop chains, respectively. The task of selecting 1-hop neighbors is similar to, but more challenging than, the Jury Selection Problem (JSP) [Cao et al. 2012]. [Cao et al. 2012] proves that JSP, within a given budget and with the goal of minimize the error rate, is NP-complete. Our RSP problem can also be specified into such an NP-complete problem, e.g., restrict the budget and set the goal of minimizing the mean error (or maximizing the Fscore). For the determination of multi-hop chains, it can be taken as executing 1-hop JSP multiple times, which is also NP-complete. A simplified problem is finding only an optimal path, which is a Multi-Constrained Optimal Path (MCOP) selection problem [Liu et al. 2013; Korkmaz and Krunz 2001], and also has been identified to be NP-complete [Korkmaz and Krunz 2001; Jaffe 1984].

Since our goal is to study the process by which a user selects recommenders and makes trust-decisions, we would like to place further analysis on the problem complexity in future work. At the current stage, we will focus on proposing some heuristic algorithms and checking the key factors in the process.

3.2. Preliminaries

There are two preliminaries that are closely related to our work.

(1) *Jury Selection Problem (JSP)*. It is used to choose the most reliable and feasible subset of all possible "jurors" to vote on a question. In JSP of [Cao et al. 2012], each juror has only two choices of 0 or 1; some jurors are selected by considering the two factors of the jury error rate and the cost. In our problem of RSP, each recommender may give a trust level in $[0, 1]$, i.e., more choices; and we consider more metrics to measure the quality of recommenders.
(2) Small-world characteristics of online social networks. Small-world network theory [Watts 1999] suggests that "there exist short paths between any two persons," and [Kleinberg 2000] provides an algorithmic view. Moreover, OSNs have been validated to bear the small-world network features [Watts 1999; Yuan et al. 2010]. Therefore, it can serve as the foundation of searching trusted chains from the source to the target.

4. SOLUTION OVERVIEW

The goal of trust evaluation is to estimate the trustworthiness of an unknown target, through proper intermediate recommenders. Our solution framework is shown in Fig. 2. There are mainly three parts, as follows:

(1) *Metrics identification*. We identify some metrics to describe the trust-related user features, and to regulate the trust evaluation systems.
(2) *1-hop recommender selection*. We aim to explore a rational approach to select an optimal subset of recommenders, when there are enough 1-hop (or direct) neigh-

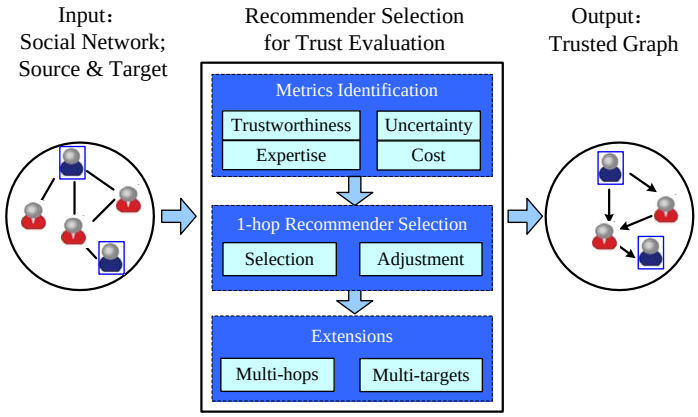


Fig. 2. Solution framework of RATE.

bors of s who know d. Many issues need to be addressed, such as "what kind of neighbor can be deemed as a good one," "how many neighbors should be selected," and "what can we do if the selected neighbors have different opinions towards d?"

(3) *Extension of multi-hop and multi-target scenarios*. In reality, it may need multiple hops for s to reach d. Moreover, s may need to evaluate the trustworthiness of multiple targets. We discuss both scenarios in the extension.

To address the recommender selection problem and to better illustrate a user's trust-decision making process, we extend FluidRating scheme in [Jiang et al. 2014], which uses fluid dynamics theory to predict a user's rating on an item. In FluidRating, we simulate the recommendation among users as fluid exchange and mixture. Fluid temperature indicates an opinion on an item. In this paper, we propose the FluidTrust scheme, which considers more factors on selecting qualified recommenders and the dynamic process of trust-decision making. Moreover, fluid temperature indicates trust on a user, instead of on an item. Details will be given in the next section.

4.1. Metrics

In order to select the most proper recommenders for current user a, we need to identify proper metrics and assign proper values for a neighbor b, from the view of a. Based on the observations in real life, we present the following four metrics:

(1) The trustworthiness of b from a's view, is denoted as t_{ab}.
(2) The expertise of b from a's view, is denoted as e_{ab}.
(3) The uncertainty of b from a's view, is denoted as u_{ab}.
(4) The cost of a's inviting b as a recommender, is denoted as c_{ab}.

We combine the four metrics as a metric vector $\mathcal{M}_{ab} =< t_{ab}, e_{ab}, u_{ab}, c_{ab} >$. The notations used in this paper are described in Table I. All the variables are normalized into the range of [0,1]. The details of these metrics are as follows.

4.1.1. Trustworthiness. We take the trustworthiness of a person b from a, equally with the trust that a puts on b, representing "a commitment to an action, based on a belief that the future actions of that person will lead to a good outcome [Golbeck 2005]." This metric is a subjective opinion of current user a, according to his direct experience of interactions with a friend b. We let the metric fall into the range of [0,1], with 0 representing no trust and 1 representing full trust.

Table I. Notations.

Symbol	Description
$G = (V, E)$	online social network
$a/b/l_{ab}$	two nodes a, b, and the link between them
N_a	the neighbor set of a
t_{ab}	the trustworthiness of b from a's view
e_{ab}	the expertise of b from a's view
u_{ab}	the uncertainty of b from a's view
c_{ab}	the cost of selecting b from a
$\mathcal{M}_{ab}$	the metric vector for a selecting b
$P_{a_1 \ldots a_n}$	a trusted path from a_1 to a_n

If two persons have direct interactions, they will build first-hand trust in each other. If not, they may build second-hand trust through intermediate users. In real trust networks including Epinions and Advogato, the trust opinions are explicitly expressed. We will use this information to assess the direct trust.

4.1.2. Expertise. In a particular trust evaluation task, we use the "expertise" to represent a neighbor's capability to make proper recommendations, including his knowledge on the topic, the target, and the source. Here, it is a broader concept, which combines the origin expertise and the affinity between two users [Wang et al. 2014].

This metric has two important features: (1) it is relatively objective compared to the subjective trust, since it considers some objective information; (2) it indicates some "mutual understanding," i.e., "I know that you know me well." Therefore, a higher expertise can lead to a higher probability, for recommenders to make a proper recommendation and for the current user to adopt such a recommendation. For instance, the expertise e_{ab} indicates both the knowledge of b and the degree that a will take b's advice, e.g., 80% adoption, or total adoption. Similarly, we let the metric fall into the range of [0,1], with 0 representing no expertise and adoption, and 1 representing highest expertise and total adoption.

To assess the expertise of a recommender b, we combine his topic-related degree $topic_b$, target-related degree $target_b$, and his affinity with a, aff_{ab}, as the following: $e_{ab} = \chi_1 \cdot topic_b + \chi_2 \cdot target_b + \chi_3 \cdot aff_{ab}$, where $0 < \chi_i < 1$, and $\sum \chi_i = 1$. For instance, topic (or target)-related degree in the Epinions web site can be calculated with the number of domains of the topic (or target) covered by b [Jiang et al. 2014]. The affinity can be gained by integrating their similarity and tie strength, as in [Wang et al. 2014].

4.1.3. Uncertainty. Uncertainty can increase the risks of transaction. One important goal of selecting qualified recommenders is to reduce the uncertainty, so as to lower the risk of failure. In the trust evaluation scenario, we use this metric to indicate an accumulative measure of the fluctuation of b, according to the historical interactions. It also indicates how much the confidence a can put on b. Therefore, we mainly focus on the success ratio of all the interactions in which a has rated b. When the evidence for success dominates, the uncertainty is lower, and vice versa. We let the metric fall into the range of [0,1], with 0 representing no uncertainty, and 1 representing complete uncertainty.

To assess the uncertainty of a user b, we count the number of interactions (e.g., reviews in Epinions) that are rated by a, denoted as num_{ab}. Within those interactions, we count the number of those a rated as helpful or very helpful, denoted as num_{ab}^{high}. Then, the uncertainty is calculated as $u_{ab} = num_{ab}^{high}/num_{ab}$.

4.1.4. Cost. Just as in daily life, when a user a wants to contact b, either directly or indirectly, some cost will be charged. The cost may be the time, communication, or simply the money he will take. Note that direct contacts to strangers may lead to a larger cost than those to indirect contacts. In an extreme case, a source can conduct multiple

direct contacts to any target, to test the trustworthiness himself, which may be quite resource-consuming. Therefore, the essence of trust evaluation is to search proper recommenders, who already know the target well, through their previous experiences.

In our work, we take the intuition that, the more close the relationship between two users, the less cost it will take to help each other. Therefore, we explore the information of tie strength to represent the cost. For instance, the more papers two authors have co-written in a scientific collaboration network, or the more the two users have corated each other in Epinions, the stronger the tie between them. The cost between a user and his direct neighbor is normalized into the range of [0,1], with 0 representing lowest cost, and 1 representing highest cost.

It is worth noting that, the assessments of the four metrics can be flexibly designed according to specific context.

4.2. Utility Functions and the Objective

In our model, we define two utility functions, denoted as $\mathcal{F}$ and $\mathcal{G}$, as the measurements of the quality and the risk/cost of social trusted paths, respectively. For a trusted path $P_{a_1 \dots a_n}$, the functions take the above four metrics t, e, u, and c as the arguments in the following two equations:

$$\mathcal{F}_{P_{a_1 \dots a_n}} = \omega_t \cdot t_{P_{a_1 \dots a_n}} + \omega_e \cdot e_{P_{a_1 \dots a_n}} \tag{1}$$

$$\mathcal{G}_{P_{a_1 \dots a_n}} = \omega_u \cdot u_{P_{a_1 \dots a_n}} + \omega_c \cdot c_{P_{a_1 \dots a_n}} \tag{2}$$

where ω_t, ω_e, ω_u and ω_c are the weights of t, e, u, and c, respectively (the weights are determined by the source s); moreover, $0 < \omega_t, \omega_e, \omega_u, \omega_c < 1$, $\omega_t + \omega_e = 1$, $\omega_u + \omega_c = 1$.

We combine $\mathcal{F}$ and $\mathcal{G}$ in a normalized utility function as follows:

$$\mathcal{U}_{P_{a_1 \dots a_n}} = \lambda \cdot \mathcal{F}_{P_{a_1 \dots a_n}} + (1 - \lambda) \cdot (1 - \mathcal{G}_{P_{a_1 \dots a_n}}), \lambda \in [0, 1] \tag{3}$$

For a trusted graph (i.e., multiple trusted paths), the utility can be calculated based on the above equations and the aggregation policy (e.g., common strategies like Max-Min and Max-Ave in [Jiang et al. 2014], or FluidTrust in this paper). The objective is to select the neighbors that satisfy multiple constraints and yields the best utility (i.e., maximize the normalized utility), with the weights specified by the source s. Note that, the utility function can also be used to measure a single recommender v, by considering the path P_{svd} (s is the source and d is the target).

As we have mentioned before, our focus will be on proposing some heuristic algorithms, taking the utility functions to serve as the selection rules.

5. RATE: 1-HOP RECOMMENDER SELECTION

In this section, we focus on the 1-hop recommenders selection, i.e., when the source can reach the the target via his direct neighbors. Four issues need to be addressed:

(1) *How to measure the quality of a recommender?* We should know what kind of recommender can be taken as 'good', before identifying good ones.
(2) *How many recommenders are enough, and are efficient for, decision-making?* Intuitively, it will be much safer if we select more recommenders, to avoid bias and to make a comprehensive decision. Then comes the question "when to stop (selecting more)?", i.e., determining the size of the optimal recommender set.
(3) *Trust aggregation and conflict resolution*. We should make a final decision by considering the opinions of the optimal recommender set. Most importantly, it usually happens that a different person has a different view on the same target. Therefore, we should design a flexible approach to resolve the possible conflicts.

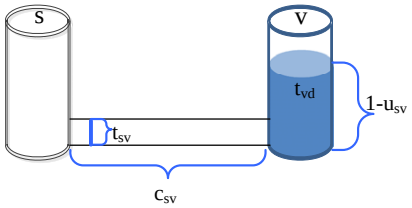


Fig. 3. Illustration of FluidTrust model.

(4) *Trust evolution and the state transition of recommenders*. If s decides to trust d, and conducts interactions with t, he will have first-hand information about d. Then, he may update his opinions (the four metrics we define in this paper) on all his neighbors, by comparing his own experience and their recommendations. Accordingly, some recommenders may receive higher scores because of their effective recommendations; meanwhile, some others' scores may be decreased.

Incited by what people do in real life, we take the 1-hop recommender selection process as follows: s first sorts his neighbors according to their qualities. Then, he requests the advice of the one with highest quality, takes part of the received advice, and forms his initial opinion. Next, he continues to request the second one. If the second one's opinion is consistent with s, s will increase his confidence on his initial idea; otherwise, he will doubt either the target, or the recommenders, or even both. The process can be continued until some m recommenders have been considered and s has got enough confidence. From real life experience, we know that the earlier a different opinion occurs, the higher the probability that s will have doubts about the target.

Taking the idea of FluidRating [Jiang et al. 2014], we model the system as shown in Fig. 3. Each user is modeled as a container, with a pipe connecting him and one of his neighbors, say s and v. The width of a pipe is equal to t_{sv}, i.e., the trustworthiness of v from s's point. The length of a pipe indicates the cost c_{sv}. Fluid temperature is equal to t_{vd}, i.e., the target's trustworthiness from the point of v. s owns a valve installed in the pipe connecting s and v. Moreover, s has the right to open the valve and listen to neighbors' advice, i.e., alow some fluid to flow into his container. Next, s will update his fluid height according to the degree to which he adopts v's advice, i.e., e_{sv}. Fig. 4(a) shows a toy example of recommender selection, and Fig. 4(b) shows the corresponding system constructed using FluidTrust model.

In the following subsections, we first introduce a new concept of "the quality of recommenders (QoR)." Then, we discuss the four issues in detail.

5.1. The Quality of Recommenders

Incited by the well-known concept of "quality of service (QoS)", which consists of several attributes, and is used to illustrate the ability of services to guarantee a certain level of performance, we present a new concept, quality of recommender (QoR).

Definition 5.1. *Quality of recommender* (QoR) comprises requirements on a recommender, taking $trustworthiness$, $expertise$, $uncertainty$, and $cost$, as attributes.

In RATE, users can set multiple quality constraints as the thresholds of the four metrics, denoted as $Q^{\theta}(\theta \in \{t, e, u, c\})$. Only neighbors who meet all the requirements can be taken as qualified and selected as recommenders. Moreover, the quality of a recommender can be measured by the utility function in Eq. 3. Since the quality of a recommender may change along with new interactions, we further define their states.

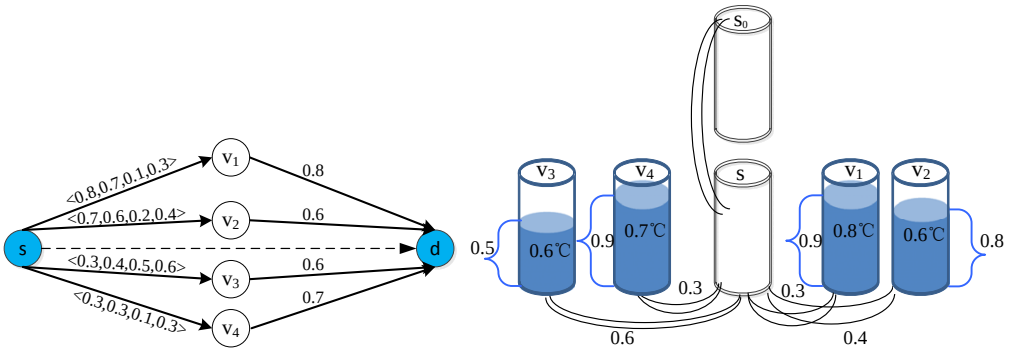


Fig. 4. (a) A toy example of 1-hop recommender selection. (b) illustration of FluidTrust model.

Definition 5.2. **State of a recommender**. A recommender has two possible states, $active$ and $inactive$, according to his QoR quality. Only the recommender who meets the QoR constraints is in the active state, i.e., he is ready to be selected.

Taking Fig. 4(a) for instance, s may specify $Q^t > 0.5$, $Q^e > 0.4$, $Q^u < 0.3$, $Q^c < 0.5$, then v_1 and v_2 can be selected. In another case, s, who cares more about trustworthiness, may even specify $Q^t \geq 0.8$; then, only v_1 can be selected.

From the example, we can see that there is a tradeoff between the quality requirements and the availability of qualified recommenders. We take a neighbor as "qualified" if he meets the quality requirements specified by the source (or a previous intermediate recommender in multi-hop scenarios).

5.2. Selecting Recommender Set

We first propose some coarse-grained approximation for the size of optimal recommender set. Based on this, we implement the reliability model to aggregate trust. In the next subsection, we will apply FluidTrust to refine the size and trust aggregation.

5.2.1. Coarse-grained Approximation. Suppose there are n neighbors, then the possible size of a subset has 2^n cases. Generally speaking, the more recommenders we select, the higher the probability that we may predict properly; however, this will lead to a higher cost to pay, and more complexity in the aggregation of different options. Therefore, we expect to select fewer recommenders, while guaranteeing a higher prediction accuracy and higher utility (larger $\mathcal{F}$ and smaller $\mathcal{G}$). Three possible choices are as follows, which can be taken as coarse-grained approximations:

(1) **Method 1**: Selecting all qualified neighbors.
(2) **Method 2**: Selecting a fixed number of qualified neighbors, e.g., 3, 6, etc.
(3) **Method 3**: Selecting a fixed proportion of qualified neighbors, e.g., 1/3, 1/6, etc.
(4) **Method 4**: Flexibly selecting some top m qualified neighbors, $m \leq n$.

For Choice 4, we present a heuristic approach (lines 5-9, Algorithm 1): we continue to select qualified recommenders until the number of next hop neighbors is larger than the current ones. The basic intuition comes from the view of information diffusion [Newman 2010], which says the information can be propagated if there are more next hop neighbors than current ones.

Algorithm 1 shows the process of 1-hop recommender selection. For line 2, a user v will assign a metric vector value for each neighbor, according to the description in the overview section (Section 4). Without loss of generality, we suppose each user has

ALGORITHM 1: BasicRATE(G, s, d)

Input: G, a social network; s, source; d, target.
Output: R_s, an optimal subset of referrals.

1 **for** *each neighbor* v *in* N_s **do**
2 Calculate the metric vector $\mathcal{M}_{sv}$.
3 keep the qualified neighbors who meet the quality constraints specified by s, in terms of the thresholds of the four metrics: trustworthiness, expertise, uncertainty, and cost.
4 Sort the qualified neighbors in descending order.
5 **for** $i : [0, m]$ **do**
6 Add the best recommender $r_i \in N_s$ to R_s.
7 $i \leftarrow i + 1$.
8 **if** *the neighbor set of* R_s *is larger than* N_s **then**
9 End the selection process.

already known the value. Then, line 2 takes $O(1)$ of time complexity; lines 1-3 takes $O(n)$, where n is the number of v's neighbors. For line 4, the time complexity is $O(n \cdot \log n)$ using the quick sorting approach. For lines 5-9, it takes $O(m)$ to select the top m recommenders, $m \leq n$. Therefore, it takes a total time complexity of $O(n \cdot \log n)$.

5.2.2. Trust Aggregation and Conflict Resolution using Reliability Model. We make use of the aggregation method in the reliability model (e.g., [Jiang et al. 2014]), where the trust value in the last hop to d is the direct trust, and the trust value from the source to the last intermediate node is taken as the reliability (of direct trust). Trust aggregation calculates the final trust value. Two commonly used aggregation functions are $MaxT$ and $WAveT$. $MaxT$ takes the trust value of the most reliable neighbor of d. $WAveT$ takes the weighted average value of all qualified neighbors of d. Taking Fig. 4(a) for instance, $t_{v_i d}, i \in \{1, 2, 3, 4\}$ are direct trust, while $t_{sv_i}, i \in \{1, 2, 3, 4\}$ are their reliability. Suppose v_1 and v_2 are selected qualified recommenders. Taking $MaxT$ aggregation, we will get $t_{sd} = t_{v_1 d} = 0.8$; taking $WAveT$ aggregation, we will get $t_{sd} = (0.8 \cdot 0.8 + 0.7 \cdot 0.6)/(0.8 + 0.7) = 0.7067$.

For the possible conflict, we take the way of post-treatment. That is, we first aggregate the opinions of qualified recommenders, by taking $MaxT$ or $WAveT$ aggregation. According to the final trust, if s decides to trust d and conducts direct interactions, he will get the first-hand information about d. That information will be used to validate and adjust the quality of recommenders. The process, together with the direct trust of s on d, is taken as "trust evolution."

5.3. Applying FluidTrust For Fine-grained Trust Aggregation

We design Algorithm 2 to apply the FluidTrust model into the process of recommender selection and trust-opinion formulation. Before introducing the details, we make the following assumptions: (1) All containers are the same size and are higher than 1. Since the confidence or certainty is at most 1, this setting will avoid overflow. (2) Each pipe is installed with a valve, and s has the right to open the valve to allow fluid to come in. (3) Pipes are installed at the bottom of all containers, and all containers are placed in the same horizontal level. (4) s has a special copy container s_0, which can absorb fluid from s. Moreover, we take a discrete and asynchronous approach. Each time that s wants to listen to some advice, he will open a valve and allow fluid to flow in for a duration of Δ. s_0 will keep and mix fluid absorbed from s. s's container will be cleaned up and kept empty at the end of a time slot.

There are two basic operations in FluidTrust: (1) allowing new fluid to come in (i.e., a person receives some recommendation); and (2) updating fluid temperature (i.e., the person refines his opinion). The former is implemented using Torricelli's law [Torricelli

ALGORITHM 2: FluidTrust(G, s, d)

Input: s, source; Q_s, sorted qualified neighbor list of s; d, target.
Output: R_s, an optimal subset of referrals.

```
SubProcess: AllowFluidFlowing(s,v)
  {s pays the cost proportional to the length of the pipe connecting s and v.
   s opens the valve to allow v's fluid coming in. The volume is calculated using Eq. 4.
   s adopts v's advice with a percentage of w_sv. }
Suppose u is the most qualified neighbor in Q_s.
Call AllowFluidFlowing(s,v) and s forms his initial opinion.
for other qualified and unvisited neighbor v' in Q_s do
    if s has reached his required confidence then
        Terminate the process.
    else
        Call AllowFluidFlowing(s,v');
        s will mix the new fluid with existing ones using Eqs. 5 and 6.
        if the advice of v' is opposite to the current opinion of s then
            s will drop some fluid to decrease his confidence/certainty using Eq. 7.
```

1643]. It states that the speed of efflux, σ, of a fluid through a sharp-edged hole at the bottom of a tank filled to a depth h is the same as the speed that a body (in this case a drop of water) would acquire in falling freely from a height h, i.e., $\sigma = \sqrt{2gh}$, where g is the acceleration due to gravity. As an application of this law, the speed of flowing fluid in our case will be $\sigma_{sv} = \sqrt{2gh_v}$. Considering the cross-sectional area t_{sv} of the pipe and the duration of the time slot Δ, the volume of flowing fluid in this time slot, Θ_{vs}, can be calculated as follows:

$$\Theta_{vs} = \sqrt{2gh_v} \cdot t_{sv} \cdot \Delta \tag{4}$$

Since s only adopts v's advice with a percentage of e_{sv}, s_0 will absorb the amount of $e_{sv} \cdot \Theta_{vs}$ fluid. The remaining fluid in s will be cleaned up. Suppose at the i^{th} time slot, s_0 already has the amount of fluid $\Theta_{s_0}(i)$, and the new coming fluid is Θ_{vs}. The mixed fluid volume will be:

$$\Theta_{s_0}(i+1) = \Theta_{s_0}(i) + e_{sv} \cdot \Theta_{vs} \tag{5}$$

According to the law of energy conservation, the fluid temperature after mixing up is calculated as follows:

$$\tau_{s_0}(i+1) = \frac{\tau_{s_0}(i) \cdot \Theta_{s_0}(i) + \tau_v \cdot \Theta_{vs} \cdot e_{sv}}{\Theta_{s_0}(i+1)} \tag{6}$$

Eq. 6 is essentially $\sum(\text{volume} \cdot \text{temperature}) / \sum \text{volume}$. In Eq. 6, τ_v is the fluid temperature in v's container.

In real life, if we hear some different opinion, we may doubt and rethink our current one, and our confidence regarding our current opinion will decrease more or less. To reflect this point, in FluidTrust, if the new coming opinion is different from the current one, we will decrease s's confidence with a parameter $\eta \in [0, 1]$. That is,

$$h_{s_0} = h_{s_0} \cdot \eta. \tag{7}$$

In Algorithm 2, the source visits at most all neighbors. Suppose that number is n. Each time he visits a neighbor, only a constant time is taken. Therefore, the complexity of Algorithm 2 is $O(n)$.

ALGORITHM 3: Evolution(G, s, d)

Input: t^*, the calculated trust; t_{sd}, the direct trust from s to d.
Output: Update the metric vector for each neighbor in N_s.

if $t_{sd} \geq Q^t$ **then**
  for *each neighbor* v *in* N_s **do**
    Let t_{vd} represent the trust value from v to d.
    if $t_{vd} \geq Q^t$ **then**
      s will update $\mathcal{M}_{sv}$ with $+$ operation.
    else
      s will update $\mathcal{M}_{sv}$ with $-$ operation.
else
  Conduct opposite operations with the above case.

ALGORITHM 4: $MultiHop\_RATE(G, s, d)$

Input: G, a social network; s, source; d, target, $d \notin N(s)$.
Output: $MCOP(s, d)$, multi-constraint optimal trusted paths.

Let L be the max length of a trusted path.
for $i : [1, L]$ **do**
  Start from s, do local greedy breadth-first search with basicRATE algorithm.
  $i \leftarrow i + 1$.
if *There exist paths from* s *to* d **then**
  Calculate the metric vector for the paths using Eqs. 3-6.
  Evaluate trust using Reliability model or FluidTrust model.

5.4. Trust Evolution and the State Transition of Recommenders

We assign each recommender two possible states, active or inactive (Fig. 5). According to their contributions of recommendation, s can adjust their quality. Through this way, s may change the states of some recommenders from inactive to active (with $+$ operation), due to their helpfulness; and vice versa.

Algorithm 3 shows the process of trust evolution. Without loss of generality, we only consider the evolution when the calculated trust $t^* \geq Q^t$, and s thus decides to conduct direct interactions with d. We do not cope with the scenarios in which a person does not take action, as his friends suggest.

Many functions can be defined for the $+$ and $-$ operations. Just taking the uncertainty for instance, we can increase the success times for $+$ and the failure times for $-$. We would like to put the design of the $+$ and $-$ operations into the future work.

6. EXTENSIONS

We discuss two possible extensions: one is the scenario that needs multi-hops to reach the target, the other involves multi-targets that need to estimate the trustworthiness.

6.1. Multi-hop Scenarios

When it needs multiple hops to reach the target, the propagation operations of the four metrics, $trustworthiness$, $expertise$, $uncertainty$, and $cost$ should be determined. We define the four equations for a trusted path.

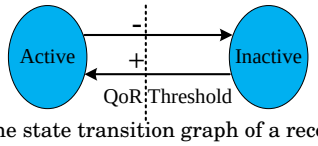


Fig. 5. The state transition graph of a recommender.

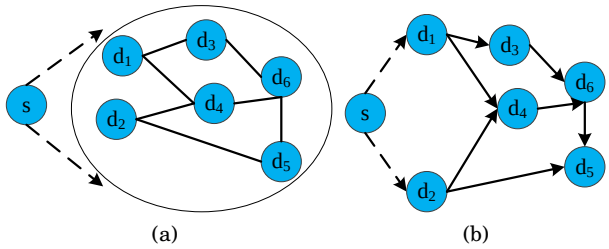


Fig. 6. (a) An example of a multi-target scenario: s needs to know multiple targets; (b) The optimal evaluation graph: s first evaluates some targets, then, it evaluates the remains via pre-known trusted ones.

6.1.1. Trustworthiness of a path. Again, we use the idea of the reliability model, in which trust propagation calculates the reliability of a trusted path. A commonly used method is $Multi$, which takes the product of trust values in all edges, as in the following:

$$t_{P_{a_1 \dots a_n}} = \prod_{l_{a_i a_{i+1}} \in P_{a_1 \dots a_n}} t_{a_i a_{i+1}} \tag{8}$$

6.1.2. Expertise of a path. We use a similar method with trustworthiness when calculating the expertise through a path $P(a_1, ..., a_n)$, as in the following:

$$e_{P_{a_1 \dots a_n}} = \prod_{l_{a_i a_{i+1}} \in P_{a_1 \dots a_n}} e_{a_i a_{i+1}} \tag{9}$$

6.1.3. Uncertainty of a path. If we take uncertainty as the probability of failure, then the uncertainty of a path is defined as follows:

$$u_{P_{a_1 \dots a_n}} = 1 - \prod_{l_{a_i a_{i+1}} \in P_{a_1 \dots a_n}} (1 - u_{a_i a_{i+1}}) \tag{10}$$

The above equation can be seen like this: the probability that the trusted path $P_{a_1 \dots a_n}$ becomes a success, is the same that all intermediate nodes behave well and finally come to a success, i.e., $\prod_{l_{a_i a_{i+1}} \in P_{a_1 \dots a_n}} (1 - u_{a_i a_{i+1}})$. Then, the probability of failure is what remains.

6.1.4. Cost of a path. It is natural to summarize all the costs of each intermediate recommender to get the cost of a path, as in the following:

$$c_{P_{a_1 \dots a_n}} = \sum_{l_{a_i a_{i+1}} \in P_{a_1 \dots a_n}} c_{a_i a_{i+1}} \tag{11}$$

It is straightforward that a shorter path will lead to less cost if the average cost on each edge is the same.

The process of the multi-hop scenario is as follows: start from s, and do a local greedy breadth-first search with Algorithm 1. If there exist paths from s to d, calculate the metric vector for the paths. Then, aggregate their results by $MaxT$ or $WAveT$ in the reliability model, or FluidTrust model, as mentioned before.

Algorithm 4 shows the process of multi-hop recommender selection, which aims to find multi-constraint optimal trusted paths. We take a local greedy approach. It takes a maximum of L hops, while each step takes $O(n \cdot \log n)$ for 1-hop recommender selection. According to the small-world characteristic of OSNs [Watts 1999], L is usually a very small constant number (e.g., 6). Therefore, it takes a total time complexity of $O(n \cdot \log n)$.

ALGORITHM 5: $MultiTarget\_RATE(G, s, D)$

Input: G, a trusted graph; s, source; D, target set; $s \notin D$.
Output: $MT(s, D)$, multi-target trusted graph.

while *there exists a target* $d \in D$ *that hasn't been evaluated trust* **do**
    Step 1: s selectively conducts direct contacts with some d, to evaluate his trustworthiness, as well as the quality as a recommender.
    Step 2: s continues to evaluate the trustworthiness of other targets, via pre-known targets.
s gradually adjusts his trust opinions according to his direct interaction experiences.

6.2. Multi-target Scenarios

We mainly consider the multi-target scenario as shown in Fig. 6(a): a new manager joins a department; to make his future work run smoothly, he needs to become familiar with his staff as soon as possible. To this end, he can frequently and directly contact each staff member; he can also meet some of his staff initially, while evaluating other staff members through some pre-known, trusted ones (Fig. 6(b)). According to further direct interactions with other staff members, he can gradually adjust his initial trust opinions (Algorithm 5).

7. EXPERIMENTAL EVALUATION

We evaluate the performance of RATE in two real social network data sets.

7.1. Experimental Design

7.1.1. Evaluation Technique. We use a standard evaluation technique in machine learning: leave one out. If there is an edge between two nodes, that edge is masked, and trust is calculated through algorithms; then, we compare the calculated value with the masked value.

7.1.2. Data Set and Preprocess. We use two real trust network data sets, i.e., Advogato (www.advogato.org) and Epinions.com (www.epinions.com). Advogato is an online social networking site dedicated to free software development. We use the snapshot collected in June 2012. It contains 56,667 links and 7,436 users. The average degree is 18.969, and the max degree is 969. On Advogato, users can certify each other on 4 different levels: Observer, Apprentice, Journeyer, and Master, which we assign 0.4, 0.6, 0.8, and 1.0, respectively, to numerate the level of trust.

Epinions is an online community web site where users can write reviews and rate other users' reviews. We use the sub set in [Jiang et al. 2014], which has 3,168 nodes and 51,888 edges; the average degree is 32.758, and the maximum degree is 748. Epinions has only a binary trust value (i.e., 1 represents trust and 0 represents no trust). However, we want the trust value to be real-valued. So we introduce Richardson's technique [Richardson et al. 2003], which uses the concept of the quality of users assigning a trust value to each node. Each user has a quality measurement $q_i \in$[0,1]. For our experiments, the quality of a user is chosen from two normal distributions of $\mu = 0.25$ (representing bad nodes) and $\mu = 0.75$ (representing good nodes). Without loss of generality, we set the proportions of good and bad nodes to be 90% and 10% (Since it has shown that Epinions is a very friendly environment). For any pair of users, a and b, the trust rating from node a to node b, denoted as t_{ab}, is uniformly chosen from $[max(q_b - \delta_{ab}, 0), min(q_b + \delta_{ab}, 1)]$. In addition, $\delta_{ab} = 1 - q_a$ is a noise parameter that determines how accurate users are at estimating the quality of the user that they are trusting. Other metrics are set as described in Section 4.2.

7.1.3. Accuracy Metrics. We mainly consider the metric of *trust accuracy*, which represents the ability of predicting whether a user will be trusted or not. We use the same metrics in [Jiang et al. 2014]:

— $Precision : A_t \cap B_t / B_t$. A_t is the number of s/d pairs on which s trusts d directly, B_t is the number of s/d pairs on which s trusts d, by trust calculated through an algorithm. *Precision* is the ratio of both the real and predicted trust users over the predicted trust ones. A higher *Precision* indicates a higher prediction accuracy.
— $Recall : A_t \cap B_t / A_t$. *Recall* is the ratio of both the real and predicted trust users over the real trust ones. A higher *Recall* indicates a higher prediction accuracy.
— $FScore : 2 \cdot recall \cdot precision / (recall + precision)$. Usually, there is a tradeoff between *Precision* and *Recall*. Therefore, we use *FScore* to measure them jointly.

7.1.4. Trust Prediction Strategies of Comparison. We implement four trust prediction strategies for comparison, they are: AveR-MaxT, AveR-WAveT, MaxR-MaxT, and MaxR-WAveT. If there are multiple paths from s to a node in N_t, *AveR* will take the average path weight as the reliability, while *MaxR* will take the maximal one. Since there is no model considering all the metrics identified in this paper, we choose typical graph-based trust models for accuracy comparison. The models are: TidalTrust [Golbeck 2005], MoleTrust [Massa and Avesani 2007b], and SWTrust [Jiang et al. 2014].

Experimental parameters are set as the following (in default): $L \in [2, 6]$, $th \in [0.5, 0.9]$, $w_t = w_e = w_u = w_c = 0.5$, $\lambda = 0.8$, and $(Q^t, Q^w, Q^u, Q^c) = (0.5, 0.5, 0.5, 0.7)$. For **Method 1**: selecting all neighbors, we actually select the first 20 trusted paths to construct the trusted graph. For **Method 2**: the fixed-number approach, we select at most 6 qualified neighbors in default. For **Method 3**: the fixed-proportion approach, we select at most 1/3 of all qualified neighbors. To test the effects of RATE, we conduct experiments for the four strategies. Since the results show similar patterns, we only present the results using AveR-MaxT.

7.2. Experimental Results and Analysis

We first compare the strategies with or without sorting recommenders by their QoR value, as to validate the effects of our proposed method. Next, we analyze the impact factors such as the maximum length and the trust threshold. Then, we test the FluidTrust scheme. Finally, we present the comparison with other trust models.

The Effects of QoR. Tables II and III show the results of prediction accuracy. We gain several findings: (1) It shows significant improvements by sorting qualified neighbors with their QoR, which indicates the effects of RATE. It gains at about 22.4% higher accuracy. (2) Also, the fixed-number strategy shows its advantage when compared to the heuristic and fixed-proportion strategies. The reason is that it actually uses more qualified neighbors than the other two strategies. We also record the uncertainty and cost. Here, only the results of fixed-proportion selection are shown in Figs. 7(a) and 7(b). The results indicate that both the average uncertainty and the average cost are decreased with sorted neighbors, which shows the advantage of RATE. In all the possible parameters settings, the least improvement occurs when $L = 6$; it is 33.45% for uncertainty, and 52.13% for cost (Note that it is only a rough estimation since some edges may be counted multiple times).

The Effects of Max Length. If the max length is large, then there will be more hops from source to destination. Also see the results in Tables II and III. With the increase of max length, the prediction accuracy is decreased. Taking the fix-number selection (without sorting) strategy for instance, the decrease percentage is at least 2.55% comparing $L = 6$ to $L = 2$, in Epinions, while 0.1% in Advogato. Meanwhile, the uncertainty and the cost are increased. The increase percentage is 232.94% and 148.03% in Epinions, while 29.04% and 36.86% in Advogato. The finding is consistent

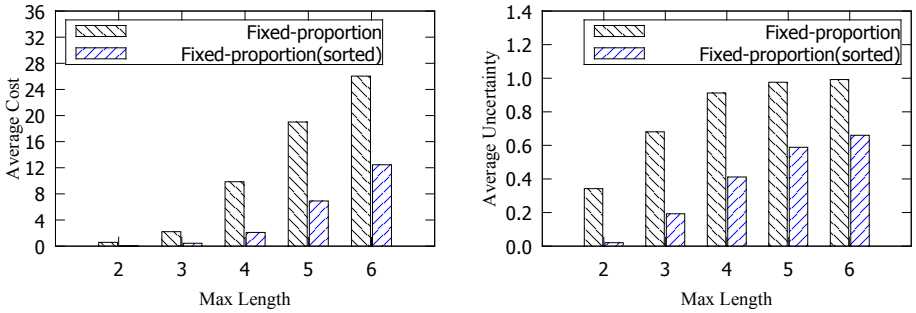
Table II. The accuracy comparison of the four methods, in Epinions.

Max Length	Method 1		Method 2		Method 3		Method 4
2	0.5285	0.8442	0.7077	0.8492	0.6815	0.8178	0.8705
3	0.5321	0.8645	0.7	0.861	0.6429	0.7907	0.861
4	0.5065	0.833	0.6622	0.8145	0.6545	0.805	0.8145
5	0.5116	0.8393	0.6605	0.8124	0.6382	0.785	0.8124
6	0.5095	0.8367	0.6513	0.8011	0.6474	0.7963	0.8011
Trust threshold	**Method 1**		**Method 2**		**Method 3**		**Method 4**
0.6	0.4173	0.7116	0.5786	0.7116	0.429	0.5277	0.5133
0.7	0.3123	0.5887	0.4786	0.5887	0.3418	0.4204	0.3841
0.8	0.2099	0.41	0.3333	0.41	0.0952	0.1171	0.2581
0.9	0.1754	0.2158	0.1754	0.2158	0.18	0.2214	0.2158

Note: (1) For the same method, the 1st column shows the FScore that does not sort neighbors with QoR, and the 2nd column does. (2) The default threshold is 0.5, while the default maximum length is 4.

Table III. The accuracy comparison of the four methods, in Advogato.

Max Length	Method 1		Method 2		Method 3		Method 4
2	0.9357	0.9798	0.956	0.9688	0.9524	0.9717	0.985
3	0.9373	0.9798	0.9604	0.9688	0.9695	0.9719	0.9818
4	0.9326	0.9796	0.9606	0.9686	0.9738	0.972	0.9808
5	0.9275	0.9788	0.9576	0.9678	0.9709	0.972	0.9797
6	0.9266	0.9768	0.9548	0.9658	0.96	0.972	0.9797
Trust threshold	**Method 1**		**Method 2**		**Method 3**		**Method 4**
0.6	0.8716	0.9757	0.9375	0.9672	0.9542	0.9719	0.985
0.7	0.8716	0.9757	0.9375	0.9672	0.9598	0.9719	0.9837
0.8	0.8716	0.9757	0.9375	0.9672	0.9598	0.9719	0.9759
0.9	0.7063	0.9733	0.8729	0.9538	0.8471	0.9599	0.985



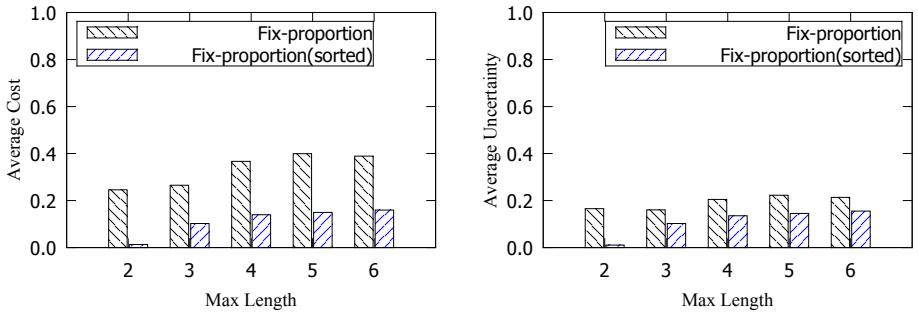
(a) (b)

Fig. 7. The comparison of accuracy, average cost, and uncertainty in Epinions.

with both real life and previous work, i.e., shorter path is better, since people tend to trust close friends rather than strangers.

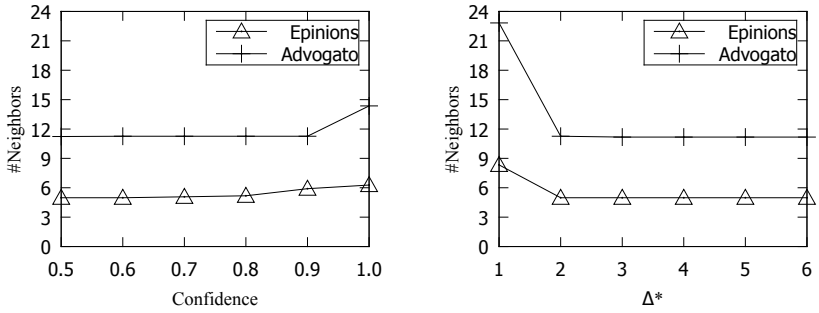
The Effects of Trust Threshold. Tables II and III show the effects of increasing trust threshold. The prediction accuracy decreases more significantly compared to that of increasing max length. Taking fixed-number (without sorting) for instance, the FScore of $Q^t = 0.5$ is 0.6622, while $Q^t = 0.9$, only 0.1754 remains. We analyze the reason to be that: too large of a trust threshold makes many paths untrustworthy, and less information can be used to predict trust. This finding validates that there is a tradeoff between the quality of recommenders and the availability of qualified recommenders.

Test FluidTrust. We test FluidTrust with the setting $L = 2$, that is, there are two hops from source to target. Other parameters are set as default. We let $\Delta^* = \sqrt{2g} * \Delta$, to adjust the speed of fluid flows. Some representative results are shown in Fig. 9. (1) Fig. 9(a) shows that, in both Epinions and Advogato, when the source requires more



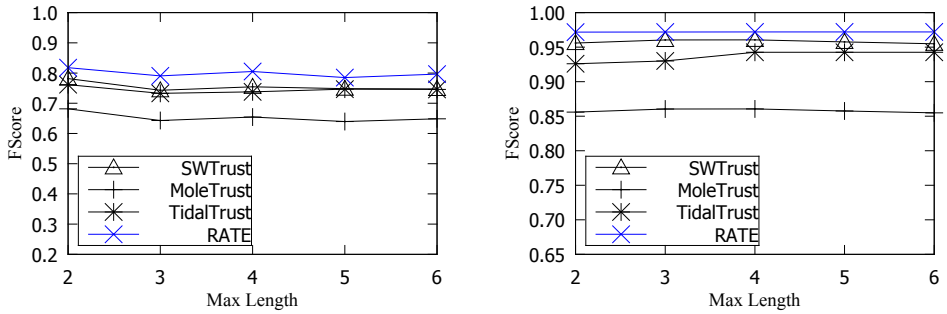
(a) (b)

Fig. 8. The comparison of accuracy, average cost, and uncertainty in Advogato.



(a) (b)

Fig. 9. The required number of neighbors with respect to the confidence and $\Delta^* = \sqrt{2g} * \Delta$.



(a) Epinions (b) Advogate

Fig. 10. The comparison with other models.

confidence on his estimation, the number of visited neighbors will increase. It is consistent with real life experience that, one's confidence is built through reinforcements from others. (2) The parameter of Δ^* indicates the amount of received recommendation from a recommender, and a larger Δ^* indicates a larger amount. Fig. 9(b) shows that, in both Epinions and Advogato, when Δ^* is small (e.g., $\Delta^* = 1$), the required number of neighbors will be large. However, when Δ^* is large enough, i.e., $\Delta^* \geq 2$, the required number of neighbors remains stable. (3) We also test the effects of η (i.e., the decrease ratio when meeting conflict recommendation). Surprisingly, it has little effect on the result. We analyze the meta results, and find the reason: although the source has vis-

ited several neighbors before he reaches a specified confidence, he actually only takes the advice of about three qualified neighbors; this is because some qualified neighbors have no idea about the target (and cannot construct a trusted path).

Comparative Study. Figure 10 shows the FScore with respect to the max length, using TidalTrust [Golbeck 2005], MoleTrust [Massa and Avesani 2007b], SWTrust [Jiang et al. 2014], and RATE. The results indicate that RATE has a better and more stable performance compared to other models. SWTrust performs the second best, since it also considers selecting neighbors with topic and target related degrees. TidalTrust performs a little better than MoleTrust, since the latter considers some less-reliable paths. RATE also has an advantage in terms of less uncertainty and cost. We do not display results here, since it may be unfair to make such a comparison.

7.3. Summary of Experiments

The above experiments validate the effectiveness of our proposed RATE, especially in that it can improve the prediction accuracy (it gains about 20% higher accuracy in Epinions and 1.8% in Advogato), while decreasing the risk (uncertainty) and cost (about 30% improvement). Meanwhile, we get some interesting findings from the experiments, including that the increased number of intermediate nodes (more hops) will decrease the accuracy while increasing the risk (uncertainty) and cost; there is a tradeoff between QoR constraints and the availability of qualified recommenders.

8. CONCLUSION AND FUTURE WORK

We propose a recommendation-aware trust evaluation (RATE) scheme, where we take a new perspective on the selection of good recommenders, to help people make proper decisions. We identify four metrics: $trustworthiness$, $expertise$, $uncertainty$, and $cost$, to measure the user features, and to adjust the recommenders dynamically. We focus on the 1-hop recommender selection strategies. We also make a simple description of coping with other more complex scenarios with multi-hops or multiple targets.

We validate the effectiveness of RATE with experiments in two real social network data sets. In the future work, we would like to analyze the theoretical bounds of the size of an optimal sub set (of recommenders) and the probability of successfully making a trust decision. Another interesting direction includes analyzing the complexity of the RSP problem, and applying the RATE scheme into real trust evaluation applications. Also, many works can be done for the design of $+$ and $-$ operations in trust evolution.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers who gave valuable suggestions that have helped to improve the quality of the manuscript.

REFERENCES

C. Cao, J. She, Y. Tong, and L. Chen. 2012. Whom to Ask? Jury Selection for Decision Making Tasks on Micro-blog Services. *Proc. VLDB* 5(11) (2012), 1495–1506.

R. I. M. Dunbar. 1992. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution* vol. 20 (1992), 469–493.

A. Etuk, T.J. Norman, M. Sensoy, C. Bisdikian, and M. Srivatsa. 2013. TIDY: A trust-based approach to information fusion through diversity. In *Proceedings of the 16th International Conference on Information Fusion (FUSION)*. 1188 – 1195.

J. Golbeck. 2005. Computing and Applying Trust in Web-based Social Networks. *PhD Thesis, University of Maryland* (2005).

J. M. Jaffe. 1984. Algorithms for finding paths with multiple constraints. *Networks* 14 (1984), 95–116.

W. Jiang, G. Wang, and J. Wu. 2014. Generating Trusted Graphs for Trust Evaluation in Online Social Networks. *Future Generation Computer Systems* 31 (2014), 48–58.

W. Jiang, J. Wu, F. Li, G. Wang, and H. Zheng. 2015. Trust Evaluation in Online Social Networks Using Generalized Flow. *IEEE Transactions on Computers (TC)* (2015). DOI:http://dx.doi.org/10.1109/TC.2015.2435785

W. Jiang, J. Wu, and G. Wang. 2013. RATE: Recommendation-aware Trust Evaluation in Online Social Networks. *Proc. IEEE NCA* (2013), 149–152.

W. Jiang, J. Wu, G. Wang, and H. Zheng. 2014. FluidRating: A Time-Evolving Rating Scheme in Trust-based Recommendation Systems Using Fluid Dynamics. *Proc. IEEE INFOCOM* (2014), 1707–1715.

W. Jiang, J. Wu, G. Wang, and H. Zheng. 2015. Forming Opinions via Trusted Friends: Time-evolving Rating Prediction Using Fluid Dynamics. *IEEE Transactions on Computers (TC)* (2015). DOI:http://dx.doi.org/10.1109/TC.2015.2444842

A. Jøsang, R. Hayward, and S. Pope. 2006. Trust Network Analysis with Subjective Logic. *Proc. ACSC* (January 2006), 85–94.

J. Kleinberg. 2000. The small-world phenomenon: An algorithmic perspective. *Proc. 32nd ACM Symposium on Theory of Computing* (2000).

T. Korkmaz and M. Krunz. 2001. Multi-Constrained Optimal Path Selection. *Proc. IEEE INFOCOM* (2001), 834–843.

Z. Liang and W. Shi. 2008. Analysis of ratings on trust inference in open environments. *Performance Evaluation* 65(2) (2008), 99–128.

G. Liu, Y. Wang, M. A. Orgun, and E. Lim. 2010. A Heuristic Algorithm for Trust-Oriented Service Provider Selection in Complex Social Networks. In *Proc. IEEE SCC*. 130–137. DOI:http://dx.doi.org/10.1109/SCC.2010.47

G. Liu, Y. Wang, M. A. Orgun, and E. Lim. 2013. Finding the Optimal Social Trust Path for the Selection of Trustworthy Service Providers in Complex Social Networks. *IEEE Transactions on Services Computing* 6(2) (2013), 152–167.

G. Liu, Q. Yang, H. Wang, X. Lin, and M.. Wittie. 2014. Assessment of Multi-Hop Interpersonal Trust in Social Networks by Three-Valued Subjective Logic. *Proc. IEEE INFOCOM* (2014).

P. Massa and P. Avesani. 2007a. Trust-aware recommender systems. In *Proc. ACM RecSys*. 17–24.

P. Massa and P. Avesani. 2007b. Trust Metrics on Controversial Users: Balancing Between Tyranny of the Majority and Echo Chambers. *International Journal on Semantic Web and Information Systems* 3 (2007), 39–64.

M. Newman. 2010. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.

S. Pushpa, K. S. Easwarakumar, S. Elias, and Z. Maamar. 2010. Referral based expertise search system in a time evolving social network. In *Proc. Compute*.

M. Richardson, R. Agrawal, and P. Domingos. 2003. Trust Management for the Semantic Web. *Proc. ISWC* 2870 (2003), 351–368.

M. Taherian, M. Amini, and R. Jalili. 2008. Trust Inference in Web-Based Social Networks using Resistive Networks. *Proc. ICIW* (2008), 233–238.

J. Tang, H. Gao, H. Liu, and A. D. Sarma. 2012. eTrust: understanding trust evolution in an online world. In *Proc. ACM KDD*.

Torricelli. 1643. http://en.wikipedia.org/wiki/Torricelli's_law (1643).

G. Wang, W. Jiang, J. Wu, and Z. Xiong. 2014. Fine-Grained Feature-Based Social Influence Evaluation in Online Social Networks. *IEEE Transactions on Parallel and Distributed Systems* 25(9) (2014), 2286–2296.

G. Wang and J. Wu. 2011. Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems (Elsevier)* 27(5) (2011), 529–538.

D. J. Watts. 1999. Small Worlds: The Dynamics of Networks Between Order and Randomness. *Princeton University Press, Princeton, NJ* (1999).

P. Yolum and M. P. Singh. 2003. Emergent Properties of Referral Systems. In *Proc. AAMAS*. ACM, New York, NY, USA, 592–599. DOI:http://dx.doi.org/10.1145/860575.860670

P. Yolum and M. P. Singh. 2005. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 35, 3 (2005), 396–407.

W. Yuan, D. Guan, and Y. Lee. April 2010. Improved trust-aware recommender system using small-worldness of trust networks. *Knowledge-Based Systems* 23 (April 2010), 232–238.