SERVICE RECOMMENDATION VIA EXPLOITING GLOBAL AND LOCAL TRUST METRICS

Tang Mingdong, Dai Xiaoling, Cao Buqing, Liu Jianxun
School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China,
mdtang@hnust.edu.cn

Abstract

Recommending users with trustworthy services is a fundamental need in service-oriented environments. Various reputation-based methods have been proposed to address the issue of service trust evaluation. They typically yield a unique and global trust score for each service by aggregating the ratings on it awarded by the user community. However, this global trust score could be inconsistent with an individual's personal opinion on the service, especially when the service is highly controversial. To attack this issue, local trust metrics have been developed, aiming to measure trust on a service more personally and accurately. However, local trust metrics, which employ trust propagation in trust networks, may fail when the judging user has no trust chains to reach the target service. By exploiting both global and local trust metrics, this paper proposes a hybrid trust measure for trustworthy service recommendation. We firstly present a global trust metric and a local trust metric different from conventional ones, and then develop a reasonable strategy for combining them to predict a user's trust on a service. Evaluations show that our proposed method significantly outperforms the other three methods in service-oriented environments with social networks.

Keywords: Service recommendation; trust; reputation; service-oriented environment; social networks

1. INTRODUCTION

With the prevalence of service-oriented computing (SOC), a variety of e-services across various domains have been developed and provided to service users in a loosely coupled environment via various technologies (such as the Web service technology) (Papazoglou, Traverso et al., 2008). From the point of view of a service user, the trust status of a service is a critical issue to consider, particularly when the service is previously not experienced by the user. However, the trust status of a service is sometimes hard to assess due to the complexity and dynamics of the service-oriented environment. For example, service users and services can join or quit the environment arbitrarily; quality of service (QoS) declared by service providers is sometimes inaccurate; the service delivery process could be undependable; besides, the preferences of service users on a service are diverse. Therefore, it is important for a service user to take into account questions such as "Should I trust this service?" In seeking for an answer to the question, many trust evaluation methods have been proposed.

Trust evaluation is usually based on the user feedback of the service quality. For example, at eBay.com, after each transaction, a buyer can rate a service based on the quality of the transaction. Trust evaluation of a service is typically performed by calculating the reputation of the service, i.e., the proportion of positive ratings among all ratings on the service. The calculated reputation value (termed as the positive feedback rating) is public and unique to all users. Other potential buyers can refer to the reputation of the service to make a decision whether to consume the service or not. eBay represents a simple reputation system, and more complicated reputation models have been proposed to improve its accuracy and robustness (Sen and Sajja, 2002). However, most reputation-based trust evaluation research takes the assumption that every service (or product) has an objective (unique) trustworthiness value and the goal of the techniques is just to guess this correct value (Massa and Avesani, 2007). We refer to this kind of trust metric as global trust. However, such an assumption is misleading. In fact, different users can have different opinions about a specific service. Thus, the global trust score of a service may not be able to reflect an individual user's personal opinion on the service.

Recently, there is an increasing interest in inferring trust between participants in social networks from a local perspective (Liu, Wang et al., 2010; Kim and Song, 2011; Xu, Liu et al., 2012). If a source participant knows a target participant, the former's trust on the latter can be assessed based only the source participant's opinion. Otherwise, if the two participants are unknown to each other, the trust degree between them is usually assessed based on trust transitivity, which means that, for example, if A trusts B and B trusts C, then A will probably trust C. However, applying this sort of local trust metric to evaluate a service's trustworthiness has two weaknesses. Firstly, it would probably fail to work or its reliability would become very low when a user either has no trust path to reach the target service or can only reach it via a long chain. Secondly, it excludes the target service's reputation from its trust inference completely, which is sometimes unacceptable in practice.

Although, the global trust metric and the local trust metric both try to predict the trustworthiness of a given service, they have distinct meanings. Generally, a global trust metric will give a unique trust score to a service, which is typically built on all the ratings given by a community of users and is thus independent on a particular evaluating user. Differently, a local trust metric produces a trust score depending on the evaluating user. That is, the local trust score is usually generated based on only the opinions from the evaluating user or his/her trustees (either direct or indirect) in the trust network. Due to the different characteristics of the global and local trust metric, we believe that it is useful to combine both of them for the trust evaluation.

To address the above issue, this paper proposed a hybrid trust evaluation method for trusted service recommendation via combining global and local trust metrics. It has been observed that with the constant growth of online social networks, service users as well as providers become more connected than ever. The integration of social networking and service provision has become an ever increasing trend. For example, at Yelp.com (a well-known local business and service search Web site in the US), every service user, besides adding ratings to various services, has a friend list, by which users connect and interact with each other. At Epinions.com, a "Web of Trust" facility is provided, whereby members can decide to either trust or block another member in the network. Amazon and eBay also have recently added social networking features to their Websites (e.g., Amazon Friends and eBay Groups). Our trust evaluation method takes into account these emerging features of service-oriented environments, and exploits both user ratings and social trust relationships for service trust evaluation and recommendation.

The rest of the paper is organized as follows. Section 2 presents a motivating example. Section 3 describes preliminaries of this paper. Section 4 elaborates our proposed trust evaluation method. Section 5 presents an example to illustrate our proposed method. Section 6 discusses the experimental settings and experimental results. Section 7 surveys related work. Finally, Section 8 concludes this paper with future work.

2. MOTIVATING SCENARIO

In this section, a motivating scenario is given to help understanding the motivation of this work. We focus on the Service-Oriented Environments with Social Networks (SOE-SN), in which, as mentioned previously, a user can publish, request and rate services, and connect with other users via, for example, adding them to his/her friend list or other operations expressing trust or distrust. The connections between users eventually make up a social network, in which trust of a user on other users or services can be propagated.

Figure 1 shows the motivating scenario. Arrows between users represent trust relationships, and real numbers next to the arrows represent trust values. Suppose that all trust values are real numbers in the continuous range of [0,1], and larger numbers represent higher trust. Arrows between users and services represent ratings of the users awarded to the services, and real numbers next to the arrows represent the rating values. Similarly, we suppose that all rating values are in the continuous range of [0,1], and larger numbers represent higher ratings. Real numbers within brackets next to services represent reputation values of the services, which are calculated based on user ratings on the services. Again, for simplicity, we suppose that all service reputation values are in the range of [0,1], and larger numbers represent higher reputation. Please note that, except the five users (A,B,C,D,E), the three services (Service 1, Service 2, and Service 3) may have other users that gave ratings to them, which are not shown in Figure 1.

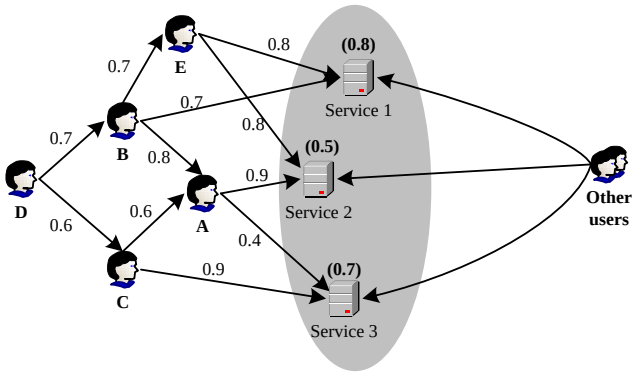


Figure 1. A service-oriented environment with social networks

As mentioned previously, a local trust metric is unnecessarily consistent with a global trust metric during trust evaluation of a service. A user may select a service because of the recommendation offered by one of his/her friends or trustees, even though the global reputation of the service is low. On the other hand, a user is also likely to reject a service because of the negative opinions from his/her friends or trustees, even though the global reputation of the service is high. These situations are quite common in practice. For example, in Fig. 1, service 1 is trusted by user A since he/she gives a high rating (0.9) to it, though the reputation of service 1 is average (0.5); whereas service 3 is distrusted by user A since he/she gives a low rating (0.4) to it, though the reputation of service 3 is relatively high (0.7).

However, it should be noted, in most cases, a user's rating or trust value on a service is likely to be consistent with its global trust value (such as reputation), like user E and service 1 in Fig.1. This is because a service's reputation is basically determined by the majority of user ratings on the service. Therefore, a service' reputation is undoubtfully an important reference to the judging user who is evaluating the service's trustworthiness. Whether the reputation of a

service should be emphasized or not in its trust evaluation actually depends on both the evaluating user's experience and the service's characteristics. To illustrate this issue, in this following we choose service 2 as the target service, and evaluate its trustworthiness from perspectives of different users A, B, C, D and E.

- As shown in Fig. 1, user A has direct experience on service 2, thus it is reasonable for him/her to heavily rely on his/her personal experiences in evaluating the trustworthiness of service 2. He/she may need to pay no or little attention to the reputation of the service.
- User B and C have no direct experiences on service 2. But they can refer to user A' opinions on service 2, since A is a trustee of them. If B and C is not very confident with the opinions of A, he/she may also need to take into account the reputation of service 2 for inferring its trustworthiness.
- User D has no direct experiences on service 2 either. Though D can ask B and C for advice, their opinions, however, may not be so dependable to D because they are second-hand experiences on service 2. In this case, User D will probably pay considerable attention to the reputation of service 2 when assessing its trustworthiness.
- User E has no trust paths to reach service 2, i.e., no trustees of him/her have experiences on service 2. He/she may have to heavily rely on the reputation of service 2 to evaluate its trustworthiness.

From the above discussions, we can see that neither a local trust metric nor a global trust metric is always sufficient for accurately predicting a service's trustworthiness degree. Therefore, we argue that, in SOE-SN, it is beneficial to combine local and global trust metrics to assess the trustworthiness of services for specific users.

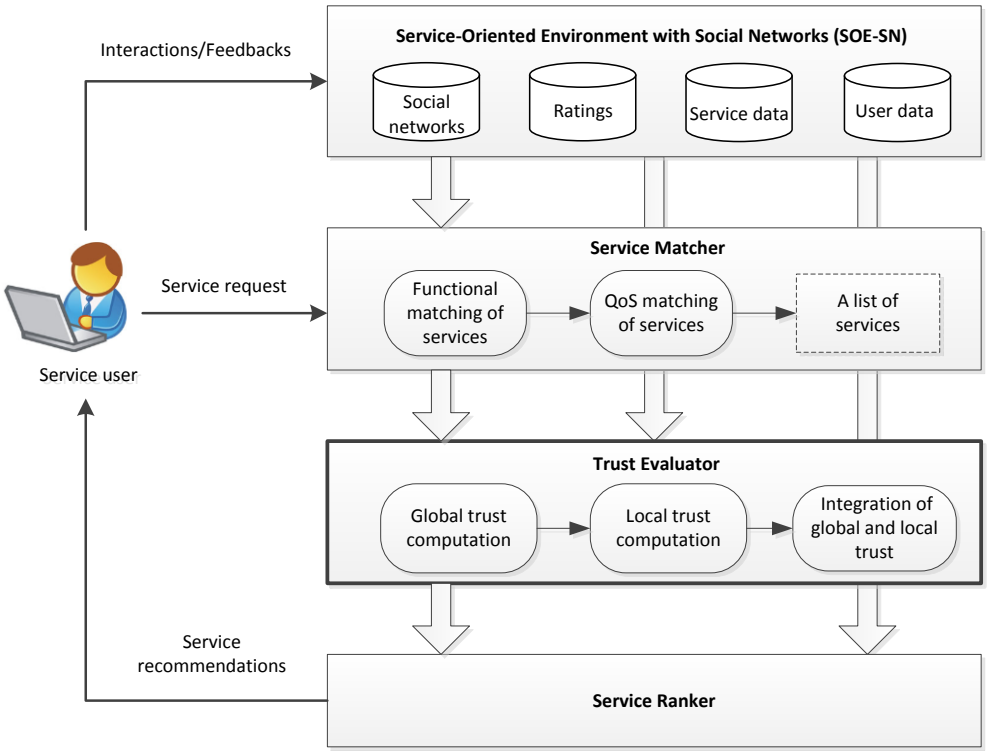


Figure 2. Trust-based service recommendation framework

3. PRELIMINARIES

In this section, we firstly present a trust-based framework for service recommendation, as shown in Fig. 2. This framework takes into account both the functional and non-functional requirements for the service discovery process by integrating functional and QoS matching of services. It also has a trust evaluation agent for identifying trustworthy services by exploiting global and local trust metrics. The proposed framework is a centralized architecture for the purpose of reducing the cost of transmission and promising high availability. In this work, we focus on the trust evaluation process, and will present the research problem and key issues in the following.

3.1 Trust-based Service Recommendation Framework

Figure 2 presents the framework of our service recommendation method for SOE-SN. The main

components and procedures in the framework are described as follows:

1) SOE-SN is a service-oriented environment that incorporates social network of users. Besides the description information of services, users, and user ratings awarded to services, it also contains the information of social relation and interactions between users.

2) Service matcher is responsible for matchmaking between the user's requirements and various services, regarding both functionality and QoS. The user's functional and QoS requirements can be extracted from his/her service request. This component firstly finds a list of services that satisfy the user's functional requirements. Then it refines the service list by filtering services with unsatisfactory QoS according to the user.

3) Trust evaluator is responsible for evaluating the trustworthiness of matched services. It firstly uses a global metric to compute the global trust scores of services via exploiting user ratings on them. Then, it computes the local trust scores of services based on the user' trust network via using a local trust metric. The global and local trust scores are finally integrated for more accurate trust inference.

4) Service ranker ranks all matched services according to their trustworthiness degrees, and recommends the most trustworthy services to the active user.

Since the issue of service matching has been widely and deeply studied, and hundreds of syntactic, semantic and QoS-based approaches for addressing it have been proposed (Plebani and Pernici, 2009; Pedrinaci, Domingue et al., 2010; Zhang, Zheng et al., 2010), how to implement the component of service matching is not a focus here and will not be discussed. Instead, we focus on the implementation of the trust evaluator in this paper.

3.2 Definition and Notation

To formally describe the problem of service trust evaluation in SOE-SN and for simplicity of discussion, we define the following notation:

- $U=\{u_1,u_2,\ldots,u_m\}$: A set of service users in a service-oriented environment.
- $S=\{s_1,s_2,\ldots,s_n\}$: A set of available services provided by different providers in a service-oriented environment.
- $G=(U,E,W)$: A social network composed of the service users in U, where $E=\{e_1,e_2,\ldots,e_x\}$ is a set of links, representing the social trust relationships between users. W is a set of weights on the links in E, representing the trust values between users. For example, $W(u,v)$ represents the direct trust value of u on v. Trust between two users could be asymmetric, i.e., $W(u,v) \neq W(v,u)$ probably holds. By this definition, the social trust network of users can be expressed using a weighted directed graph.
- $R=(U,S,E',W')$: A bipartite network composed of the users in U and the services in S, where $E'=\{e'_1,e'_2,\ldots,e'_y\}$ is a set of links between the users and the services and W' is a set of weights on the links. Each link between a user and a service represents that the user has rated the service, and its weight represents the rating value. For example, $W'(u,s)$ represents the rating value awarded by a user u to a service s.
- $SS=G\cup R$: A service-oriented environment with social networks can be abstracted as a union of G and R.

The problem researched in this paper is: for any user $u\in U$ and any service $s\in S$ in a service-oriented environment with social networks SS, how to exploit G and R, to accurately predict the trustworthiness of s for u.

We suppose that both trust levels between users and rating values on services have already been acquired. The rating values can be given by users when they rate services after service consumption. The trust levels between users can be explicitly declared by users, or be inferred from the social relationships or interactions between users. This issue has been addressed in previous work (Skopik, Schall et al., 2010). If there are no explicit social relationships or interactions between users, the trust levels between users can also be estimated based on other information concerning users, such as interest similarity and locality (Ma, King et al., 2011; Hang, Zhang et al., 2013). In the experimental evaluation section (Section 6), we will provide an example to illustrate this issue. For simplicity, we assume that all trust levels and rating values are in the continuous range [0,1], where a larger number indicates a higher trust level or rating value. In particular, number 0 represents a complete distrust or dislike, whereas number 1 represents a complete trust or like. We also assume that, the rating value given by a user to a service is equivalent to the user's trust value on the service. That is, if a user gives a high rating value to a service, then he/she would trust the service, and vice versa.

4. Trust Evaluation Using Global And Local Trust Metrics

As indicated by the motivating scenario, there is a clear need to exploit both global and local trust metrics to measure a service's trustworthiness in service-oriented environments with social networks. In this section, we firstly propose a global trust metric and a local trust metric by extending traditional ones, and then discuss how to integrate them for accurate and personalized trust evaluation of services.

4.1 Global Trust Metric

The most widely used global trust metric is reputation. Reputation systems have been widely adopted by real e-commerce systems such as eBay and Amazon, to evaluate trustworthiness of products or services. For example, at Amazon, reputation of a service is calculated by taking an

average of all rating values on the services. Though these reputation evaluation methods are very simple and easy to understand, they have some serious weaknesses. Firstly, they seldom took the controversy of services into consideration, and thus did not distinguish between controversial services and non-controversial services. Secondly, they are likely to be manipulated by malicious users that give unfair ratings to services. Our global trust evaluation method will address the first issue, and the second issue can be well addressed by incorporating local trust measures, which we will discuss later.

In the following we propose a global trust metric by extending the reputation model of Amazon. The global trust metric which has two parts can be denoted by $T_G(s) = <R(s), Con(s)>$, where *Rep(s)* is the reputation value of *s*, and *Con(s)* is the controversiality degree of *s*. Before we elaborate the details of *Rep(s)* and *Con(s)*, for simplicity of presentation, we define the following notation:

- $U(s)$: a subset of users in U that rate service s.
- $r(u,s)$: the rating value given by user u to service s; if u has more than one ratings to s, the value of his/her latest rating is employed.
- $\bar{r}(s) = \sum_{u \in U(s)} r(u,s) / |U(s)|$: the average ratings value on s.

In this work, the reputation value of s is simply calculated using $\bar{r}(s)$, like Amazon, i.e.

$$R(s) = \bar{r}(s) = \sum_{u \in U(s)} r(u,s) / |U(s)| \quad (1)$$

However, the above reputation value alone is insufficient to capture a service's trustworthiness to a user. For example, suppose there are two services s1 and s2 whose set of rating values are {0.7,0.7,0.7} and {0.4,0.8,0.9} respectively. The two services have the same averaging rating value 0.7, but clearly, their reputation values have different certainty degrees to a specific user. The averaging rating value 0.7 seems to be appropriate to represent s1' reputation, but unnecessarily appropriate to represent s2' reputation. Actually, it is hard to measure the reputation of services like s2, since they are controversial to different users. In real service-oriented environments, many services could be controversial, i.e., they are likely to be appreciated by many users but also disliked by many other users. This is because services are usually judged by users in very diverse ways. This paper defines the controversiality degree of a service, e.g., *s*, as follows

$$Con(s) = \frac{\min(\#trust\_users(s), \#distrust\_users(s))}{\max(\#trust\_users(s), \#distrust\_users(s))} \quad (2)$$

where #*trust_users*(*s*) represents the number of users that trust service *s* and #*distrust_users*(*s*) represents the number of users that distrust *s*.

From the above definition, we can see that the controversiality degree of a service is actually the ratio of the minority of ratings to the majority of ratings on the service, with respect trust or distrust. The larger the ratio, the greater is the controversiality degree. The values of *Con(s)* are always real numbers in the continuous range [0,1]. In particular, when *Con(s)* = 1, i.e., the number of users who trust the service *s* equals to the number of users who distrust *s*, it indicates that service *s* has the highest controversality. In contrast, *Con(s)* = 0 indicates that *s* has the lowest controversality. To determine #*trust_users*(*s*) and #*distrust_users*(*s*), we assume that if a user's rating value given to *s* is larger than 0.5, he/she is a member of *trust_users*(*s*); otherwise, if a user's rating value given to *s* is smaller than 0.5, he/she is a member of *distrust_users*(*s*); if a user's rating value given to s is exactly 0.5, he/she belongs to neither *trust_users*(*s*) nor *distrust_users*(*s*). The controversiality degree of a service can also be expressed using linguistic terms such as very low, low, medium, high and very high. This is done by mapping values of *Con(s)* to suitable linguistic terms, which depends on specific applications. Table I presents an example of mapping between controversiality values and linguistic terms.

Table I. Mapping Between Controverslality Values and Linguistic Terms

controversiality values	[0,0.1)	[0.1,0.2)	[0.2,0.4)	[0.4,0.6)	[0.6,1]
linguistic terms	very low	low	medium	high	very high

With the definition of controversiality, two services with similar reputation values can be well distinguished when evaluating its trustworthiness to a user. Again, let's take the two services *s1* and *s2* as examples, whose set of rating values are assumed to be {0.7,0.7,0.7} and {0.4,0.8,0.9} respectively. The two services have the same averaging rating value 0.7, but their controversiality values are quite different, i.e., *Con*(*s1*)=0 and *Con*(*s2*)=0.5. Obviously, *s2* has a much higher controversiality degree than *s1*. For a service with low controversiality, its reputation value is easier to measure and more certain to an evaluating user. However, for a service with a high controversiality degree, it is not appropriate to employ only its reputation value to predict its trustworthiness to a specific user, as indicated in (Massa, Avesani et al., 2007). Therefore, there is a need to use a local trust metric to evaluate the service's trustworthiness, from the very personal views of a user.

4.2 Local Trust Metric

Local trust metrics focus on employing the peculiar experiences (either direct or indirect) of an evaluating user for predicting trustworthiness of a service. Herein, by direct experiences we mean the experiences of the evaluating user and by indirect experiences we mean the experiences that are obtained from the trusted neighbors of the evaluating user in a community. In general, while local trust metrics can be more precise and tailored to the single user's peculiar

views and opinions, they are also computationally more expensive, since they have to be computed for every single user whereas global trust metrics are just performed once for all the community. Nevertheless, it is widely believed that local trust metrics can achieve higher accuracy than global ones in predicting the trust a specific user should place into a controversial object (Massa, Avesani et al., 2007). As mentioned previously, another advantage of local trust metrics is that they are more attack-resistant (Golbeck and Hendler, 2006): users who are considered malicious (from a certain user's point of view) can be excluded from trust propagation and thus have little influence on the evaluating users who don't trust them explicitly.

This section presents a local trust metric for accurately predicting an evaluating user's personal trust placed on a peculiar service. Similar to the proposed global trust metric, the local trust metric is also denoted by a two-tuple, i.e., $T_L(u,s)=<T(u,s), Rel(u,s)>$, where $T(u,s)$ represents the computed local trust value of u on s; and $Rel(u,s)$ represents the reliability of $T(u,s)$. In the following, we discuss how to compute $T(u,s)$ and $Rel(u,s)$.

To compute the local trust of a judging user on a service, we employ a trust propagation algorithm that is similar to the TidalTrust algorithm (Golbeck and Hendler, 2006), which is known as a famous and highly cited algorithm for inferring trust in trust networks. The major differences are two aspects: TidalTrust considers trust values to be numbers in a continuous range of [0,10], while we considers the trust values to be in [0,1]; TidalTrust predicts trust between users, whereas we predict trust between users and services. The employed trust propagation algorithm is described as follows.

Suppose that a user u wants to judge the trustworthiness of a target service s. If u has rated s, i.e., they are directly connected in R, we set $T(u,s)= W'(u,s)$, where $W'(u,s)$ is the rating value awarded by u to s. Otherwise, if u has no direct experiences on s, the following steps are involved.

1) u sends a request to all its trustees in G. This request is recursively forwarded until it reaches to users having ratings on s.

2) The trust values of these users on s are moved backward across the paths that their corresponding requests are came from. In the backward path, when a user receives more than one trust value, it uses WAO (weighted average operator) to combine them.

3) This scenario continues until trust values reach to u across the same paths of sending requests but in reverse direction.

4) After this, u uses WAO to combine its received trust values and compute final trust value from u to s.

Considering the fact that in social networks users usually have many paths to connect with each other, we apply two restrictions to the above algorithm, similar to the work (Golbeck and Hendler, 2006). Firstly, only the shortest paths from source to sink are considered in trust propagation. Secondly, the social trust links between users that have low trust values, for example, are smaller than a trust threshold, will be excluded from trust computation. The above algorithm can be performed efficiently, in time $O(m+n+x+y)$, where m and n represent the total number of users and services in SS respectively, and x and y represent the total number of links among users and links between users and services in SS respectively.

To clearly illustrate the above local trust computation algorithm, let's take Fig. 1 as an example. Suppose that service 2 is a target whose trustworthiness will be judged by users A, B, C and D. For user A, since he/she has experienced service 2 previously, he/she would like to predict the trustworthiness of service 2 based on his/her past rating value 0.9. For user B who has no experience on service 2, he/she sends his/her request to A and E, who then return their trust values on service 2, 0.9 and 0.8 respectively. B combines trust values from A and E using WAO to infer his/her trust on service 2, as follows

$$T(B,2)=\frac{T(B,A)\times T(A,2)+T(B,E)\times T(E,2)}{T(B,A)+T(B,E)}=\frac{0.8\times 0.9+0.7\times 0.8}{0.8+0.7}=0.85$$

In a similar manner, we have

$$T(C,2)=\frac{T(C,A)\times T(A,2)}{T(C,A)}=\frac{0.6\times 0.9}{0.6}=0.9$$

$$T(D,2)=\frac{T(D,B)\times T(B,2)+T(D,C)\times T(C,2)}{T(D,B)+T(D,C)}=\frac{0.7\times 0.85+0.6\times 0.9}{0.7+0.6}=0.87$$

One disadvantage of the TidalTrust algorithm is the way it treats the difference between the source users near to the sink and the source users remote to the sink. By observation we know that the shorter the trust chain from a source to a sink, the more reliable is the trust of the source on the sink. To overcome the uncertainty of trust prediction caused by long trust chains, TidalTrust simply restrict the distance from a source to a sink to a small hop number, e.g., 3. That is, if their distance is greater than a preset threshold, the trust of the source on the sink will not be computed. In this way, the source users with different but acceptable hop distances to the sink will be treated equally with respect to their calculated trust values. This is not reasonable in practice. For example, suppose that the preset threshold of hop distance is 3, by applying TidalTrust to the trust network in Fig.1, we get $T(A,2)=0.90$, $T(B,2)=0.85$,

$T(C,2)=0.90$, and $T(D,2)=0.87$, as calculated previously. Based on this calculation, all the users *A*, *B*, *C* and *D* have very close trust values on service 2, i.e., service 2 is almost equally trustworthy to users *A*, *B*, *C* and *D*. This is not the truth based on our observation. The reliability of the predicted trust values of *A*, *B*, *C* and *D* should be different, since their trust chains to service 2 have different lengths. This paper proposes a local trust metric via explicitly incorporating the reliability of local trust computation.

As mentioned above, the reliability of local trust computation in trust networks is heavily dependent on the length of the trust chain from the source to the sink. When the trust chain from the source to the sink is sufficiently short, the local trust computation is reliable. As the trust chain becomes longer, the reliability of local trust computation decreases correspondingly. Therefore, it is reasonable to use length of trust chain to infer the reliability of local trust computation. Herein, we employ the length of the shortest path from the judging user to the target service to infer the reliability of a local trust computation, as the following formula:

$$Rel(u,s)=\frac{1}{SPL(u,s)} \tag{3}$$

where $SPL(u,s)$ represents the shortest path length from *u* to *s*. Similar to controversiality, reliability of a local trust computation can also be expressed using linguistic terms, such as very low, low, medium, high and very high, depending on how distant is the judging user to the target service. Table II presents an example of mapping reliability values to linguistic terms, which is similar to what in Table I. In pratice, how to map values of $SPL(u,s)$ and $Rel(u,s)$ to linguistic terms can be customized for specific applications.

Table II. Mapping From Trust Path Length, Reliability Values To Linguistic Terms

SPL(u,s)	1	2	3	4	>4
Rel(u,s)	1	0.5	0.33	0.25	<0.25
linguistic terms	very high	high	medium	low	very low

Please note that *u* could have no trust path to reach *s* in a trust network. In this case, we treat $SPL(u,s)$ as infinite, and the above local trust computation will become infeasible. To address this issue, we set $T(u,s)=R(s)$ and $Rel(u,s)=0$ by default.

4.3 Combining Global and Local Trust Metrics

As discussed in the motivating scenario (see Section 2), merely using a global trust metric or a local trust metric cannot be always accurate for trust evaluation in SOE-SN. Therefore, it is useful to combine global and local trust metrics for accurate trust inference. Based on the motivating scenario and our observations, we develop the following heuristics for combining the global and local trust metrics in trust inference:

- If the evaluating user has first-hand or reliable second-hand experiences on the target service, his/her experiences should be emphasized in predicting the service's trustworthiness, especially when the target service is of high controversiality.
- If the evaluating user has neither first-hand nor reliable second-hand experiences on the target service, the reputation value of the service should be emphasized in predicting its trustworthiness, especially when the service is of low controversiality.
- The local trust metric should significantly weight more than the global one in trust prediction when the local trust computation has a high reliability and the controversiality of the service is high.
- The global trust metric should significantly weight more than the local one when the local trust computation has a low reliability and the controversiality of the service is low.

Based on the above heuristics, we use the following formula to aggregate the proposed global and local trust metrics for accurate and personalized trust inference:

$$PT(u,s)=\frac{R(s)\times w_R+T(u,s)\times w_L}{w_R+w_L} \tag{4}$$

Where *PT(u,s)* represents the predicted trust value between *u* and *s*, *R(s)* and *T(u,s)* represent the reputation value and local trust score of *s* respectively; w_R and w_L are the importance weight of *R(s)* and *T(u,s)* respectively. The values of w_R and w_L can be determined using Table III, which is designed based on the heuristics listed above. For instance, if the service's controversiality is very low and the reliability of the service's local trust computation is very high, the ratio of w_R over w_L is set to 1, which indicates that the global and local trust values computed are equally treated in their aggregation. If the service's controversiality is very low and the reliability of the service's local trust computation is also very low, the ratio of w_R over w_L is set to 5, which indicates that the global trust value computed is much more emphasized than the local trust value in their aggregate. Hence, different *Con(s)* and *Rel(u,s)* can lead to different ratios between w_R and w_L, and therefore lead to different combinations of the global and local trust metric. It is worth noting that, Table III is only an example to determine the ratios between w_R and w_L, which can also be customized for specific applications in practice.

Eventually, we predict the trustworthiness degrees of all service candidates that match the active user's functional and QoS requirements, and recommend the Top-K most trustworthy services to the user.

Table III. The Ratio Of The Weight Of Global Trust Over That Of Local Trust In Trust Prediction In Different Situations (W_R/W_L)

Rel / Con	*Very high*	*high*	*medium*	*low*	*Very low*
Very low	1	2	3	4	5
low	1/2	1	2	3	4
medium	1/3	1/2	1	2	3
high	1/4	1/3	1/2	1	2
Very high	1/5	1/4	1/3	1/2	1

5. AN ILLUSTRATIVE EXAMPLE

We take Fig.1 as an example to illustrate our proposed method, including the global trust computation, the local trust computation and the hybrid trust computation. Suppose that all ratings on service 1, 2 and 3 are provided in Table IV. As we can see, service 1(s1) has four ratings, with an average rating value 0.8; service 2(s2) has five ratings, with an average rating value 0.5; and service 3(s3) has six ratings, with an average rating value 0.7. The average rating values of the three services also have been displayed in Fig.1.

Table IV. Ratings On Services In Fig. 1

Rating ID	Service 1 (s1)	Service 2 (s2)	Service 3 (s3)
1	0.8	0.8	0.9
2	0.8	0.9	0.4
3	0.7	0.3	0.8
4	0.9	0.2	0.6
5	-	0.4	0.7
6	-	-	0.8

We calculated the global, local and final trust value on the three services using our proposed method, and reported the results in Table V. According to our proposed global trust metric, the controversiality degrees of s1, s2 and s3 are 0, 0.67 and 0.2 respectively. By employing Table I to transform the controversiality degrees to linguistic terms, we can obtain that s1 has a very low controversiality degree, s2 has a very high controversiality degree, and s3 has a medium controversiality degree. The global trust values on s1, s2 and s3 are independent of judging users, i.e., do not change for different users. In contrast, local trust values on services could be different for different judging users, as shown in Table V. Different from traditional local trust inference methods, in addition to predicting the local trust value of the judging user on the target service, we also compute a reliability value for the predicted local trust value. This distinguishes a local trust prediction with high reliability from one with low reliability. For example, both users B and D are predicted to have a local trust 0.7 on s1, however, the predicted trust values of B and D are 1.0 and 0.5 respectively, indicating that the former predicted trust value is more reliable than the latter. This is because B has direct experience on s1 while D has only second-hand experience. When no trust paths exist from a user to a service, the default local trust value is set with the reputation values of the service and its reliability is set to 0. Such examples include user A and service 1, user C and service 1, user E and service 3, etc.

The hybrid trust value of a service in Table V are calculated based on the global and local trust values of the service, as discussed in Section 4. The following discussion takes user A and service 2 as an example to illustrate the computation of hybrid trust. Their global and local trust values are (0.5, 0.67) and (0.9, 1.0) respectively. Since the controversiality of service 2 is 0.67, i.e., *very high* according to Table 1, and the reliability of local trust computation is 1.0, also *very high* according to Table 2. By checking Table 3 we can obtain that the ratio of the contribution weight of global trust over that of local trust for predicting trustworthiness of s2 to A should be 1/5. Therefore, the hybrid trust value of s2 is calculated as

$$PT(A, s2) = \frac{0.5 \times 1 + 0.9 \times 5}{1 + 5} = 0.83 \qquad (5)$$

As stated previously, our method can balance global trust and local trust very well when predicting services' trustworthiness in SOE-SN. This can be seen from Table V. Both the global trust metric and the local trust metric are likely to produce undesirable predictions, while these situations can be largely avoided by our proposed hybrid trust metric. For example, for service 1 whose controversiality is *very low*, the global trust metric can work well, but the local trust metric does not, especially for the judging users who have no trust paths to reach service 1, such as user A and C. Another example is for service 2 whose controversiality is *very high*. In this case, the global trust metric cannot work well, but the local trust metric does, especially for the judging users who have short trust paths to reach service 1, such as user A, B, C and E. In both examples, our hybrid trust metric can work very well and can produce more reasonable results than the other two trust metrics.

Table V. The results generated by our proposed method using the example in Fig. 1

Users	Service 1			Service 2			Service 3		
	Global Trust	Local Trust	Hybrid Trust	Global Trust	Local Trust	Hybrid Trust	Global Trust	Local Trust	Hybrid Trust
A	(0.80, 0.00)	(0.80, 0.00)	0.80	(0.50, 0.67)	(0.90,1.00)	0.83	(0.70,0.20)	(0.40,1.00)	0.48
B	(0.80, 0.00)	(0.70, 1.00)	0.75	(0.50, 0.67)	(0.85, 0.50)	0.78	(0.70,0.20)	(0.40, 0.50)	0.50
C	(0.80, 0.00)	(0.80, 0.00)	0.80	(0.50, 0.67)	(0.90, 0.50)	0.82	(0.70,0.20)	(0.90, 1.00)	0.85
D	(0.80, 0.00)	(0.70, 0.50)	0.77	(0.50, 0.67)	(0.87, 0.33)	0.78	(0.70,0.20)	(0.90, 0.50)	0.83
E	(0.80, 0.00)	(0.80, 1.00)	0.80	(0.50, 0.67)	(0.80, 1.00)	0.75	(0.70,0.20)	(0.70, 0.00)	0.70

6. EXPERIMENTAL EVALUATION

To evaluate the performance of our proposed trust evaluation method, we conducted a set of simulation-based experiments. The experiments are developed by using Matlab 7.0 and performed on a HP desktop computer with configuration as: Intel Core i3 3.20GHz CPU, 2GB RAM, and Windows 7 operating system. The simulation environment is built based on a real world dataset, which captures the actual interactions between users and services. The ratings for services by individual users are generated based on the observed rating patterns in this dataset. The trust relationships and trust values between users are generated based on the assumption that users have similar interests are likely to trust each other.

6.1 Experimental Design

During our experiments, we adopted a real-world Web service dataset, WSDream dataset 2 (Zheng, Zhang et al., 2014), published in www.wsdream.com. The original dataset contained the QoS records of service invocations on 5825 Web services from 339 users. Each QoS record is comprised of two quality attributes (response time and throughput) as well as their values. Since the 5825 Web services have a variety of functionalities, to meet the requirements of our method, we refine the services by searching them with the keyword "search". After that, we get 215 services, which can be considered having the same or similarity functionality. The experimental dataset after preprocess contains 339 users, 215 services, as well as the QoS data generated from their interactions. In the following, we describe how to generate rating values on services and trust values between users based on the above service data.

Based on the above dataset, we simulate users' ratings for each Web service via exploiting the QoS information. A user's rating value on a service can be estimated using his/her satisfaction degree on the quality of the service provided that the service's functionality has met the user's requirement. In previous work, the satisfaction degree of a service to a user in an individual QoS aspect is usually calculated with the following normalization formula (Kang, Liu et al., 2012),

$$r_i(u,s)=\begin{cases}\dfrac{q_i(u,s)-\min q_i}{\max q_i-\min q_i} & \text{if } i \text{ is a positvie QoS aspect}\\ \dfrac{\max q_i-q_i(u,s)}{\max q_i-\min q_i} & \text{if } i \text{ is a negative QoS aspect}\end{cases} \quad (6)$$

where $r_i(u,s)$ represents u's satisfaction degree on service s' QoS aspect i, which is supposed in [0,1], $q_i(u,s)$ represents the value of QoS aspect i for service s, and $\max q_i$ and $\min q_i$ represent the maximum and minimum value of QoS aspect i for all services respectively. The computation of $r_i(u,s)$ depends on whether QoS aspect i is positive or negative. A positive QoS aspect indicates that, the larger its QoS value is, the higher the satisfaction degree is. In contrast, a negative QoS aspect indicates that, the larger the value, the lower the satisfaction degree. However, the above satisfaction degree computation method may produce undesired results because of the influence of exceptional QoS values. For example, assume that the throughput values of all services are within [0,100], other than one service, whose throughput value is 1000. In this case, the satisfaction degrees calculated for all services except one will be below 0.l, which is unreasonable.

To address the above weakness, we use the Gaussian normalization method to compute the satisfaction degree of a user on a service's QoS aspect (Ortega, Rui et al., 1997). Given a QoS value of a positive QoS aspect such as throughput, the Gaussian normalization used in this work is,

$$r_i(u,s)=0.5+\frac{q_i(u,s)-\bar{q}_i(u)}{2\times 3\sigma_i(u)} \quad (7)$$

where $\bar{q}_i(u)$ represents the average QoS values observed by u on all service he/she invoked in the QoS aspect i, and $\sigma_i(u)$ represents the standard deviation of the values in QoS aspect i from u's observations. In Formula (7), $3\sigma_i(u)$ is used because of the well-known 3-σ rule, which indicates that, for the Gaussian normalization, 3σ is large enough to make more than 99% given values normalized into the range

[0,1]. For the normalized values outside [0,1], their values will be set to either 0 or 1, depending on which one is more closer. This significantly prevents the influence from exceptional QoS values. The above Gaussian normalization is for a positive QoS aspect. When a negative QoS aspect is considered, we replace Formula (7) with

$$r_i(u,s) = 0.5 + \frac{\bar{q}_i(u) - q_i(u,s)}{2 \times 3\sigma_i(u)} \quad (8)$$

After computing the satisfaction degrees of user *u* in all concerned individual QoS aspect for service *s*, we use the following formula to aggregate them to obtain the overall satisfaction degree of user *u* on *s*, i.e., the rating value *r*(*u*,*s*),

$$r(u,s) = \sum_{i=1}^{k} w_i \times r_i(u,s) \quad (9)$$

where w_i , which satisfies $\sum_{i=1}^{k} w_i = 1$, represents the contribution weight of $r_i(u,s)$ in computing $r(u,s)$, *k* is the number of QoS aspects concerned with service *i*. In our dataset, two QoS aspects, response time and throughput, are involved. For simplicity, we let $w_1 = w_1 = 0.5$.

We also simulated the trust values between users. To do this, we assume that users who have similar ratings on services are likely to trust each other to some extent. This assumption is reasonable in practice. Pearson Correlation Coefficient (PCC) is widely used to measure the similarity between two entities in recommender systems. In this work, we calculate PCC for two users *u* and *v* based on their ratings as

$$PCC(u,v) = \frac{\sum_{s \in I} \left(r(u,s) - \bar{r}(u)\right)\left(r(v,s) - \bar{r}(v)\right)}{\sqrt{\sum_{s \in I} \left(r(u,s) - \bar{r}(u)\right)^2} \sqrt{\sum_{s \in I} \left(r(v,s) - \bar{r}(v)\right)^2}} \quad (10)$$

where *I* represents the services that are rated by both *u* and *v*, $\bar{r}(u)$ and $\bar{r}(v)$ are the average rating value given by *u* and *v* respectively. The values of PCC are always in the continuous range [-1,1]. To be consistent with the range of our trust value, we adjust the PCC values and measure the similarity between users using

$$sim(u,v) = \frac{1 + PCC(u,v)}{2} \quad (11)$$

The values of $sim(u,v)$ should always be in the range [0,1], where larger values indicate higher similarity degrees. To generate the trust value between *u* and *v*, we specify that, if $sim(u,v)$ is greater than a threshold θ , then the trust value of *u* on *v* is $T(u,v) = sim(u,v)$; otherwise, *u* and *v* are not considered similar and there is no trust relationship between them.

After the above process, the final experimental dataset is called *SS*, which contains a user-user trust matrix where each item is a trust value between two users (denoted *G*), and a user-service rating matrix where each item is a rating value of a user on a service (denoted *R*).

6.2 Experimental Results

The evaluation study is conducted using the above dataset *SS*. We use the user-user matrix *G* and part of the user-service matrix *R* as the training set, and treat the remaining part of *R* as the testing set. Then we make the training set as input, and computes the trust value of the users on each service. For each user, we recommend top-K services that he/she has never invoked in the training set , i.e., the K services that the user would trust most. The recommendation quality is assessed according to the number of hits (recommendations that match the actual top-K services in the testing set). The following precision is calculated assess the quality of recommendations:

$$precise: P_u = \frac{\#hits}{K} \quad (12)$$

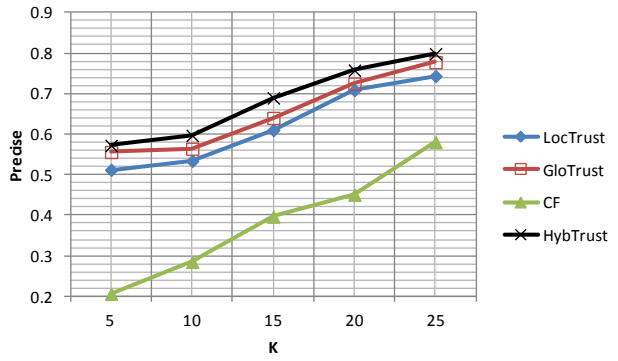
The precision is to assess the accuracy of the recommended services relative to the user's potential trust. For example, 20% precision indicates that two out of ten recommendations would actually be ranked in the top-10 by the target user. The above precision is computed for all testing users, and the average values is reported.

We implement the proposed trust-based service recommendation method and other three service recommendation methods:

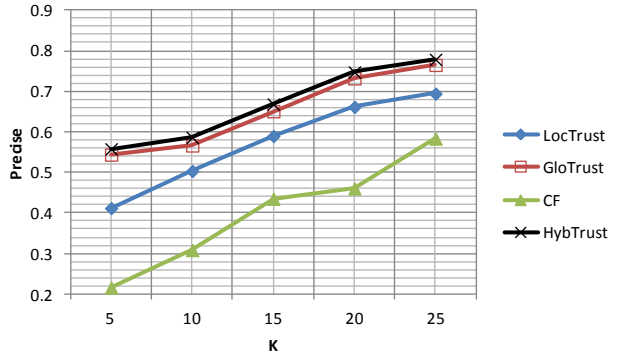
- Recommendation based on reputation of services. The trust score of a service is represented by its reputation, which is calculated by taking an average of the rating values of the service. This method uses a global trust metric and thus is named GloTrust for short.
- Recommendation based on local trust propagation. The method implements TidalTrust, which employs trust propagation to compute the trust of the testing users on services. This method uses a local trust metric and thus is named LocTrust for short.
- Recommendation based on collaborative filtering (Shao, Zhang et al., 2007). The method uses classic collaborative filtering techniques to predict the missing rating values on services for the testing users and recommends the top-K services with largest predicted rating values. This method is named CF for short.

Fig.3 presents some experimental results of our proposed method (denoted by HybTrust) and the above three methods. In this experiment, to imitate the real situations that a user probably only rates a small fraction of all services, we remove 80% of the items in the user-service matrix and use them as the testing set, the remaining 20% items are used as the training set. We vary top-*K* from 5 to

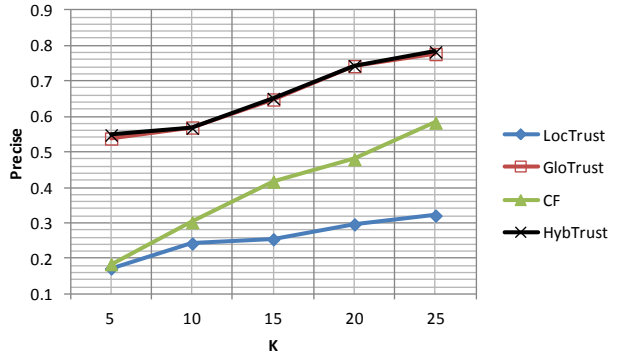
25 with a step 5 and the trust threshold θ from 0.5 to 0.7 with a step 0.1 during the experiment, to evaluate their influence on the performance of different methods. We can see that, all the tested methods' performance is getting better with the increase of *K*, and our method has the best performance, i.e., precise, among all methods.



(a) θ=0.5



(b) θ=0.6



(c) θ=0.7

Fig.3. Performance comparison between our proposed trust evaluation method and the other methods while varying K and θ

The parameter θ, which determines the density of the user-user trust matrix, has little influence on GloTrust and CF, but has significant influence on LocTrust on our method HybTrust. As we can see, the precise of LocTrust decreases significantly when θ becomes larger. This is because larger θ will lead to lower density of the user-user trust matrix, which subsequently reduces the reliability of trust propagation between the users. Since our method is a combination of GloTrust and LocTrust, it can also be affected by θ to some extent. This is also shown in Fig. 3. When the value of θ changes from 0.5 to 0.7, the difference between the precise of our method and GloTrust becomes smaller.

We also evaluated the influence of the density of the user-service rating matrix on the performance of the above methods using an experiment. In the experiment, we set top-K =10, θ=0.6, and vary the density of the user-service rating matrix from 0.05 to 0.5 with each step 0.05. The density *d*, means that, (1-*d*) proportion of the ratings in the original user-service matrix is removed as the testing set and the remaining *d* proportion of the matrix is used as the training set. For example, when we remove 80% ratings in the original user-service matrix, then the remaining matrix should have a density 0.2. Fig.4 presents the results of this experiment. We can see that, all the tested methods' performance is getting better with the increase of the density, and our method is demonstrated to have the best performance again.

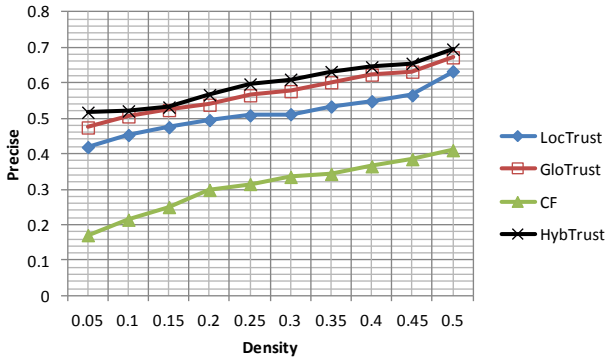


Fig.4. Impact of the density of the user-service rating matrix on precise

7. Related Work

Service selection or recommendation deals with choosing an appropriate service from a set of services so that the service can meet the user's requirements or preferences. Trust is a fundamental basis for a user to select a service. To select trustworthy services, QoS is widely taken into consideration. Many QoS-driven service selection and recommendation methods have been proposed (Zeng, Benatallah et al., 2004; Comuzzi and Pernici, 2009; Zhang, Zheng et al., 2010; Yau and Yin, 2011; Liu, Fletcher et al., 2012). They focus on identifying services with the most satisfactory QoS according to user's requirements and preferences. However, QoS data of services is hard to be acquired and therefore often incomplete (Zheng, Ma et al.,

2009). In addition, QoS data of services might not be reliable (Yau, Huang et al., 2010). Therefore, the conventional QoS-driven service selection methods such as those discussed above may be impractical.

To address reliability issue of QoS, researchers used QoS monitoring mechanism to improve QoS data quality. Kalepu, Krishnaswamy et al. (2004) proposed to assess reputation of services based on difference between claimed and actual delivered QoS. Vu, Hauswirth et al. (2005) proposed an advanced personalized selection and ranking algorithm based on monitored QoS data. However, these methods still can't resolve the incomplete QoS information issue since monitored QoS data may not be available. In these cases, other trusted data should be used for service selection such as experiences of service requesters and opinions from other users.

It is common to assess a user's trust in a service based on experiences and opinions from service users (Zeng, Benatallah et al., 2004; Xu, Martin et al., 2007; He, Yan et al., 2009; Noorian, Fleming et al., 2012). For example, Zeng, Benatallah et al. (2004) proposed a QoS-aware web service selection and composition method that takes service reputation into consideration. The value of the reputation is defined as the average ranking of the service by all users. Xu, Martin et al., (2007) proposed a model of reputation-enhanced QoS-based Web services discovery that combines an augmented UDDI registry to publish the QoS information and a reputation manager to assign reputation scores to the services based on customer feedbacks on their performance. However, they did not consider credibility of raters. He, Yan et al. (2009) proposed a trust management approach ServiceTrust to support reputation-oriented service selection. The ServiceTrust takes rater's credibility into consideration by combining a user's and other personal ratings to estimate a service provider's trust value. However, ServiceTrust models transactions as binary events (success or failure) and does not consider user's preference when performing service selection. Noorian Fleming et al. (2012) also evaluated a service provider's trust value by combining the service requester's and the other rater's experience opinions, and proposed a preference-oriented QoS-based service selection method.

Several works such as (Yau, Huang et al., 2010; Nguyen, Zhao et al., 2010) combined user experience opinions and QoS monitoring information to evaluate trust of services. For example, Yau, Huang et al. (2010) proposed an approach to improve trustworthiness of QoS of services to develop high-quality workflows in service-based systems (SBS). The approach is based on identifying the deviation between the QoS profiles of the services claimed by their service providers and the QoS profiles determined by monitors and service user feedbacks. Nguyen, Zhao et al.(2010) proposed a trust and reputation model based on Bayesian network for Web services. The model integrates users' ratings, QoS monitoring measure, and direct experience of the requester. However, Nguyen's model of user ratings as binary events (satisfactory or unsatisfactory) is crisp and impractical.

Most existing trust models for services focus on evaluating trustworthiness of a service (or provider) by inferring its reputation. Reputation of a service represents a sort of public trust in the service. Many trust models assume or generate a unique reputation with a service, which is global and unique to all users (Xu, Martin et al., 2007). However, personalized trust models are more desirable to satisfy different users' preferences. Therefore, more trust models for services take users' preferences into consideration when exploring user ratings or QoS monitoring information to infer reputation of a service. Still, existing trust models for services have significant weaknesses. They assume all users are independent, and rarely take connections among users into consideration. In reality, connections, i.e., social relationships, among users, usually play a crucial role in service selection and recommendation. For example, a user may depend on opinions of his/her trusted friends in service selection. A user is also likely to accept a service recommended by his/her trusted friends. Indeed, social relationship information provides another type of source for evaluating trustworthiness of services.

Hang (2011) proposed a couple of trust models for social and service networks. A service network is defined as a network composed of trust relationships between services as well as trust relationships between users and services (providers). In a service network, nodes are mainly services. Hang proposed several generalized trust propagation methods for inferring indirect trust between participants in either social networks or service networks. However, how to integrate social relationships among users with the service network is not considered in his work. Wu et al. [09] proposed a service ranking method also based on service networks. They rank services from two aspects: quality of service (QoS) and social information. Social information includes (1) how many users a service has and (2) how frequently a service is invoked. They did not take social connections among service users into consideration.

Liu, Wang et al. (2010) investigated trustworthy service provider selection in complex social networks composed of service users and providers. To assess the trustworthiness of a service provider, they proposed a trust propagation method that takes three kinds of social trust factors among participants into consideration: trust degree, social intimacy degree, and role impact factor. Various factors are aggregated for assessing trust. Xu, Liu et al. (2012) extended the above work and proposed a more efficient approach for selecting trustworthy service provider. However, they both did not consider other important sources of trust for services such as QoS of services, user' ratings on services and user preferences, thus cannot be directly applied to service selection and recommendation in service-oriented environments.

With the prevalence of online social networks, social connections among users actually have been considered to develop trust-based recommender systems (Ma, King et al., 2009; Ma, King et al., 2011; He and Chu, 2010). Unlike traditional recommender systems that only exploit the information in the user-item rating matrix, trust-aware recommender systems also exploit trust relationships among users, inferring preference similarity among users based on their social relationships. They can not only address the data-sparsity and the cold-start problem suffered by other types of recommender systems, but also can improve accuracy and efficiency of recommendations. Those trust-aware recommender systems, though inspiring, their ideas may not be directly applied to service recommendation in service-oriented environments, where QoS of services is crucial and QoS factors are typically objective.

8. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a trust evaluation method for services in service-oriented environments with social networks. The method exploited both a global trust and a local trust metric to measure a service's trustworthiness for personalized service recommendation. Simulation-based experiments showed that the method significantly outperforms the other methods which exploit either mere global trust or mere local trust.

In the future, we will investigate different aggregate methods for combining global and local trust metrics, such as using fuzzy operators. We also plan to evaluate our method and its improved version by conducting more experiments using real data.

9. ACKNOWLEDGMENT

The work described in this paper was supported in part by the National Natural Science Foundation of China under grant No. 61402168, No. 61572186 and No. 61572187, and in part by the Scientific Research Fund of Hunan Provincial Education Department of China under Grant 15K043.

10. REFERENCES

Comuzzi, M., Pernici, B. (2009). A Framework for QoS-Based Web Service Contracting, *ACM Transactions on the Web*, Vol. 3(3), Article 10.

Golbeck J., and Hendler J. (2006). Inferring trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, Vol. 6, No. 4, pp. 497–529.

Hang C.W., Wang Y., and Singh M. (2009). Operators for propagating trust and their evaluation in social networks. *In AAMAS'09*, pp. 1025–1032.

Hang C.W. (2011). Probabilistic Trust Models for Social and Service Networks. *Ph. D. dissertation*. North Carolina State University.

Hang C.W., Zhang Z., and Singh. Shin M. P. (2013). Generalized Trust Propagation with Limited Evidence, *IEEE Computer*, 2013, pp. 78-85.

He Q., Yan J., Jin H., and Yang Y., (2009). ServiceTrust: Supporting Reputation-Oriented Service Selection, *ICSOC-Service Wave* 2009, LNCS 5900, pp. 269–284.

He J. and Chu W. W., (2010). A Social Network-Based Recommender System (SNRS), *Annals of Information Systems*, 2010, pp. 47-74.

Kalepu S., Krishnaswamy S., and Loke S. W. (2004). Reputation = f(user ranking, compliance, verity). *In Proceedings of the IEEE International Conference on Web Services* (ICWS '04), Washington, DC, USA, 2004. IEEE Computer Society.

Kang G., Liu J., Tang M., Liu X. F. (2012). AWSR: Active Web Service Recommendation Based on Usage History. *2012 IEEE International Conference on Web Services* (ICWS 2012), research track, Hawaii, USA, June, 2012

Kim Y. A., Song H. S. (2011). Strategies for predicting local trust based on trust propagation in social networks, *Knowledge-Based Systems*, 24(8): 1360-1371.

Lesani M., Montazeri N. (2009). Fuzzy trust aggregation and personalized trust inference in virtual social networks, *Computational Intelligence*, 25 (2): 51–83.

Liu G., Wang Y., Orgun M. A., and Lim E. p. (2010). A heuristic algorithm for service provider selection in complex social networks. *In SCC'10*, pp. 130–137.

Liu X.F., Fletcher K.K., Tang M. (2012). Service Selection based on Personalized Preference and Trade-Offs among QoS Factors and Price. *The 1st IEEE International Conference on Services Economics* (SE 2012), June, 2012

Ma H., King I., and Lyu M. R. (2009). Learning to recommend with social trust ensemble. *In SIGIR*, pp 203–210.

Ma H., King I., Lyu M.R. (2011). Learning to recommend with explicit and implicit social relations, *ACM Transactions on Intelligent Systems and Technology*,2(3):29.

Massa P. and Avesani P. (2007). Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems* (IJSWIS), 3(1).

Nguyen H.T., Zhao W., Yang J. (2010). A Trust and Reputation Model Based on Bayesian Network for Web Services. *IEEE International Conference on Web Services*. 2010.

Noorian Z., Fleming M., and Marsh S. (2012). Preference-Oriented QoS-based Service Discovery with Dynamic Trust and Reputation Management, *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (SAC '12), 2012, 2014-2021.

Ortega M., Rui Y., Chakrabarti K., Mehrotra S., Huang T.S. (1997). Supporting similarity queries in MARS. *In: Proc. of the ACM Multimedia*. 1997. 403−413.

Papazoglou M. P., Traverso P., Dustdar S., and Leymann F. (2008). Service-oriented computing: a research roadmap. *Int. J. Cooperative Inf. Syst.*, 17(2):223–255.

Pedrinaci C., Domingue J., Sheth A. (2010). Semantic Web Services. In: Handbook on Semantic Web Technologies. *Volume Semantic Web Applications*. Springer (2010)

Plebani P. and Pernici B. (2009). URBE: Web Service Retrieval Based on Similarity Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 2009, 21(11):1629-1642.

Sen S. and Sajja N. (2002). Robustness of reputation-based trust: Boolean case. *In 1st International Jount Conference on Autonomous Agents and Multiagent Systems*, 2002.

Shao L., Zhang J., Wei Y., Zhao J., Xie B., and Mei H., (2007). Personalized QoS prediction for Web services via collaborative filtering, *Proc. IEEE International Conference on Web Services* (ICWS 07), 2007, pp. 439-446.

Sherchan W., Loke S.W., Krishnaswamy S. (2006). A fuzzy model for reasoning about reputation in web services. *In: Proceedings of ACM Symposium on Applied Computing*, ACM Press, 2006, pp.1886-1892.

Skopik F., Schall D., and Dustdar S., (2010). Modeling and mining of dynamic trust in complex service-oriented systems, *Information Systems*, vol. 35.

Vu L.-H., Hauswirth M., Aberer K., (2005). QoS-based service selection and ranking with trust and reputation management, *Proceedings of OTM'05*, R. Meersman and Z. Tari (Eds.), LNCS 3760, 2005, p.p. 466-483.

Wu Q., Iyengar A., Subramanian R., Rouvellou I., Silva-Lepe I., and Mikalsen T.A. (2009). Combining quality of service and social information for ranking services. *In Proceedings of the 7th International Conference on Service Oriented Computing*, 2009, pages 561–575.

Xu Z., Martin P., Powley W., Zulkernine F., (2007). Reputation-Enhanced QoS-based Web Services Discovery, *In ICWS*, 2007.

Xu Y., Liu J., Tang M., Cao B., Liu F. (2012). An Efficient Search Strategy for Service Provider Selection in Complex Social Networks. *IEEE International Conference on Services Computing*. 2012. pp.130-137.

Yau S., Huang J., Yin Y.(2010). Improving the Trustworthiness of Service QoS Information in Service-based Systems, *Proceedings of 7th Autonomic and Trusted Computing* (ATC), Xi'an, China, October 26-29, 2010, 208-218.

Yau S.S. and Yin Y., (2011) QoS-based Service Ranking and Selection for Service-based Systems, *Proceedings of the International Conference on Services Computing* (SCC), 2011, pp. 1-8.

Zeng L., Benatallah B., Ngu A., Dumas M., Kalagnanam J., and Chang H. (2004). Quality-aware middleware for Web service composition, *IEEE Trans. Softw. Eng*., 2004, 30(5), 311–327.

Zhang Y., Zheng Z., and Lyu M. R. (2010). WSExpress: A QoS-aware Search Engine for Web Services. *ICWS* 2010: 91-98

Zheng Z., Ma H., Lyu M. R., and King I., (2011). QoS-Aware Web Service Recommendation by Collaborative Filtering, *IEEE Transactions on Services Computing*, 2011,vol.4, no.2, pp. 140-152.

Zheng Z., Zhang Y., and Lyu M. R., (2014). Investigating QoS of Real-World Web Services, *IEEE Transactions on Services Computing* , 2014, vol.7, no.1, pp.32-39.

Authors

Tang Mingdong is currently an associate professor of the School of Computer Science and Engineering at Hunan University of Science and Technology, China. He received his Ph.D. degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2010. His current research interests include services computing, cloud computing, and social networks. He is a member of CCF, IEEE and ACM. He has also been the Vice Chair of Services Society Young Scientists Forum (China) and the CCF Young Computer Scientists & Engineers Forum (Changsha Branch).

Dai Xiaoling is a master student in the School of Computer Science and Engineering, Hunan University of Science and Technology, China. She received her B.S. degree in Computer Science and Technology from Hunan University of Science and Technology, Xiangtan, China, in 2013. Her research interests include service recommendation and selection.

Cao Buqing received his M.S. degree in computer science from Central South University of Technology in 2007, and Ph.D. degree from Wuhan University in 2010. He is now a lecturer of the School of Computer Science and Engineering at Hunan University of Science and Technology. His current interests include service computing, social network, software engineering. He has published more than 30 papers in well-know conferences and journals, such as ICWS, SCC, JWSR,etc.

Liu Jianxun received PhD degree in computer science from Shanghai Jiao Tong University in 2003. He is now a professor of Department of Computer Science and Engineering, Hunan University of Science and Technology. His current research interests include work-flow management systems, services computing, and cloud computing. He has published about 80 academic papers in international journals or conference proceedings.