Deng SG, Huang LT, Wu J *et al.* Trust-based personalized service recommendation: A network perspective. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(1): 69–80 Jan. 2014. DOI 10.1007/s11390-013-1412-7

Trust-Based Personalized Service Recommendation: A Network Perspective

Shui-Guang Deng (邓水光), *Member, ACM, IEEE*, Long-Tao Huang (黄龙涛), Jian Wu* (吴　健) and Zhao-Hui Wu (吴朝晖), *Senior Member, IEEE*

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

E-mail: {dengsg, hlt218, wujian2000, wzh}@zju.edu.cn

Received June 27, 2013; revised November 16, 2013.

Abstract Recent years have witnessed a growing trend of Web services on the Internet. There is a great need of effective service recommendation mechanisms. Existing methods mainly focus on the properties of individual Web services (e.g., functional and non-functional properties) but largely ignore users' views on services, thus failing to provide personalized service recommendations. In this paper, we study the trust relationships between users and Web services using network modeling and analysis techniques. Based on the findings and the service network model we build, we then propose a collaborative filtering algorithm called Trust-Based Service Recommendation (TSR) to provide personalized service recommendations. This systematic approach for service network modeling and analysis can also be used for other service recommendation studies.

Keywords personalized service recommendation, trust, network modeling and analysis, collaborative filtering

1 Introduction

The rapid growth of Web services on the Internet requires effective service recommendation mechanisms, aiming to recommend high quality Web services to the users. Many Web services deliver similar functions with different quality properties such as response time, throughput, and availability. These properties can be used to reflect the quality of the underlying Web services and are called Quality-of-Service (QoS) in Web service domain[1]. Existing service recommendation mechanisms focus on utilizing the QoS information of individual services[2–4] and often provide the same service(s) to various users. However, different users may have distinct requirements and views on the QoS of their desired services. Therefore, a personalized service recommendation mechanism[5] is needed to recommend services that satisfy each individual user based on his or her view/trust on Web services.

In this paper, we study trust relationships between users and services from a network perspective. Based on the findings and the network model we build, we develop a collaborative filtering algorithm called Trust-Based Service Recommendation (TSR), to rank Web services for a user based on the level of this user's trust towards the QoS of the services. To evaluate TSR's performance, we conduct a series of large-scale simulation experiments based on a real world dataset which contains information about 2000 services and 100 users. The experimental results show that TSR outperforms other traditional techniques. To summarize, TSR makes personal recommendations for individual users based on the trust relationships between users and services.

We claim three major contributions as follows:

1) First, we investigate the impact of trust relationships between users and Web services for providing personalized service recommendations, which is neglected by most existing service recommendation methods.

2) Second, we introduce the network-based modeling and analysis method and the systematic evaluation experiments, which can also be used in other Web service research domains (e.g., Web service selection[6], Web service composition[7]).

3) At last, we develop a network-based algorithm, which can effectively provide users with personalized service recommendations based on their trust and views on the QoS of desired services.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the

Regular Paper

This research is supported in part by the National Key Technology Research and Development Program of China under Grant No. 2013BAD19B10, and the National Natural Science Foundation of China under Grant No. 61170033.

*Corresponding Author

network-based model for trust relationships between users and Web services. Section 4 describes the trust-based service recommendation algorithm (TSR) based on collaborative filtering techniques. Section 5 describes our evaluation experiments and results. Section 6 concludes this paper and discusses the future directions.

2 Research Background and Related Work

2.1 Web Service Recommendation

In recent years, various recommendation techniques have been developed and adopted to improve Web service discovery and selection. Early researches on service recommendation mainly focus on the individual properties of Web services[8-12], especially on QoS[4,13-16]. In [4], a collaborative filtering approach was designed to predict QoS value of a Web service based on its past performance. In [15], Zhu *et al.* proposed a novel landmark-based QoS prediction framework and developed two service recommendation algorithms based on clustering techniques. In addition, Blake and Nowlan[11] proposed to compute a score for each Web service based on their performance when interacting with users. However, such recommendation methods provide the same services to all the users with similar requests, and largely ignore the heterogeneity among users' views on the QoS of the recommended services.

Therefore, in order to choose the most appropriate services that best accommodate the users' QoS needs, personalized service recommendation techniques are greatly needed. In [12], Maamar *et al.* proposed a model to characterize the context of Web service interactions, and highlighted that the resources on which Web services were performed had impacts on Web service personalization. In [17], a region-sensitive recommendation approach was proposed to predict the personalized QoS value of a service for each user by analyzing the contextual information of interactions between users and services such as region location and network delay. It makes personalized prediction mainly based on information about QoS properties while largely ignores users' specific QoS needs. To address this problem, our work aims to provide personalized service recommendations by studying individual users' preferences on QoS and how such preferences affect their trust relationships with the underlying services.

2.2 Trust for Web Service

Although several studies[16-17] also address personalized service recommendations, they mainly focus on individual user characteristics rather than users' views on QoS which largely determine users' satisfactory with the recommended services. Therefore, we study the trust relationships between users and services in Web service recommendation, aiming to develop better mechanisms for personalized service recommendation.

Trust is a kind of relationship between users and services which can express users' subjective perception and preference on services[18]. There are many trust and reputation mechanisms proposed for Web service selection[19-20], which are mainly inspired by trust management in traditional distributed environment such as P2P networks[21]. The trust value rated by a user is largely affected by his/her interaction experiences with the service. A user's rating score for a service is a very important factor for users to share their direct experiences of interaction with the Web services. Most trust models for Web services regard feedback rating by users as a key source of trust[22-26]. In [22], Malik and Bouguettaya provided a reputation evaluation by integrating a feedback mechanism for the target services. The reputation value is impacted by users' credibility, preferences, and temporal sensitivity. In [23], a trust model based on Bayesian network was proposed, which regards the direct experience opinion, the recommendation from other consumers, and the QoS monitoring information as the key sources of trust. Most of the existing trust models do not consider the heterogeneity of users' trusts and views on QoS of the same service(s). They mainly reflected the collective or aggregated perceptions of all users on a targeted service, but could not model and predict individual user's trust on this service. Since the preference of different users may be various, our trust model is going to stress the impact of ratings from users with similar tastes while weaken the impact from users with different tastes.

2.3 Collaborative Filtering

Collaborative filtering (CF), which was firstly proposed by Rich[27], has been proved to be a very successful approach in utilizing relational information such as trust for personalized recommendation. In this study, we adopt this approach to develop a trust-based service recommendation algorithm to provide personalized service recommendations.

There are several studies applying CF techniques to provide Web service recommendation. Shao *et al.*[28] proposed a user-based CF algorithm to predict QoS values. Zheng *et al.*[29] proposed a hybrid CF algorithm to recommend Web services. Rong *et al.*[30] applied the CF algorithm and used MovieLens[31] data for experimental analysis. However, these studies focus on the individual properties (e.g., QoS) of services while do not consider information about personal preferences of QoS and subject views on services from different users, which is vital for personalized service recommendation.

Therefore, our study utilizes the CF techniques to provide personalized service recommendations for a user by leveraging ratings from users that share similar QoS trust with him/her. Two kinds of CF algorithms, user-based CF[32] and item-based CF[33] are adopted for our recommendation algorithm. User-based collaborative filtering algorithms attempt to discover a group of users who have similar tastes with the target user. Several similarity measurement models have been developed to compute their similarity[34-35]. A larger similarity score indicates that two users are more similar in terms of their views on products/services that need to be recommended. Then this score can be used to predict the likelihood of users' satisfactory on the recommended items. On the other hand, item-based algorithms differ from the user-based algorithms only in that they compute item similarity measurements instead of user similarities. They also predict rating values for the new items in the target user's profile.

In addition, collaborative filtering techniques have provided an effective approach for utilizing users' (trust) relationships towards products/items for personalized recommendation. Therefore, our study aims to address the limitations in the existing service recommendation studies by:

1) modeling and analyzing trust relationships between users and services as networks;

2) applying collaborative filtering techniques on the modeled networks to learn users' trust preferences on QoS;

3) providing personalized service recommendation based on the learned information about users' preferences.

3 Network Modeling for Trust Relationships

In this section, we model the trust relationships between users and services using network modeling techniques. The two networks modeled (the user trusting network and the service trusted-by network) are then analyzed to discover users' trust preferences on QoS.

3.1 Network Perspective of Trust Relationships

Trust relationships between users and services can be naturally represented as a network by treating users and services as vertices and trust relation user-service pairs as edges. We refer to the network of trust relationships as the trust network *T-Net*, which is a bipartite network consisting of two disjoint sets of vertices. And there is no edge connecting vertices in the same set. In the area of social network, this kind of network is also called an affiliation network, of which edges represent affiliation relations[36]. The trust network is defined formally as follows.

Definition 1 (Trust Network). *A trust network is a network of trust relationships between users and services based on users' ratings on services, and it can be modeled as a* 3*-tuple T-Net* (U, S, R) *where*:

1) $U = \{u_1, u_2, \ldots, u_m\}$ *is a set of users, where* u_i $(1 \leqslant i \leqslant m)$ *denotes a user,* m *is the number of all users, and* u_i *is a target user that needs Web service recommendation.*

2) $S = \{s_1, s_2, \ldots, s_n\}$ *is a set of services with similar functionalities, where* s_i $(1 \leqslant i \leqslant n)$ *denotes a service, and* n *is the number of all services.*

3) $R = \{rel(u, s) | u \in U,\ s \in S\}$ *is a set of trust relations between users and services. Many existing work*[6] *has regarded feedback ratings for services from users as a proper measure to assess the extent to which users trust services. Hence,* R *can be also treated as a set of ratings for services from users, and the value of* $rel(u, s)$ *is defined as the score of* s *rated by* u.

Fig.1 shows an example of a trust network in which we use square vertices to represent users, cycle vertices to represent services, directed edges to represent users' rating relationships with services. The weights of edges are equal to the values of users' ratings.

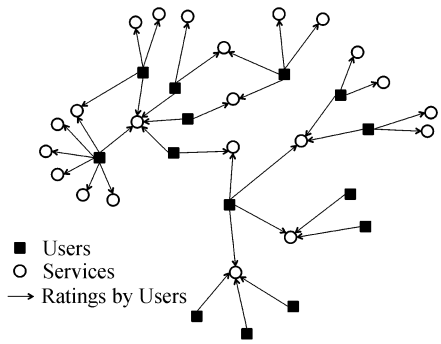


Fig.1. Example of a trust network.

3.2 Projection of Trust Network

The previous network-related studies mainly focus on unipartite networks. Thus, most of previous empirical studies on bipartite networks often project such networks into two unipartite networks, each with only one type of vertices[37]. We adopt a similar method in our research. The above bipartite trust network is projected into a user network and a service network, which is defined as follows.

Definition 2 (User Network). *A user network is a network of users where an edge between two user vertices represents that the two users have rated at least* *one common service previously* (*co-rating relationship*), *and it can be modeled as U-Net* $\langle U, CR\rangle$ *where* U *is a set of users and* CR *is a set of co-rating relationships between users.*

Fig.2(a) shows an example of user network, where U is presented as vertices in the network, and CR is presented as edges. The weights of the edges are the similarities of users and the calculation is introduced in detail in Subsection 4.2.

Definition 3 (Service Network). *A service network is a network of services where an edge between two service vertices represents that the two services have previously been rated by at least one common user* (*co-rated relationship*), *and it can be modeled as S-Net* $\langle S, CR'\rangle$ *where* S *is a set of services and* CR' *is a set of co-rated relationships between services.*

Fig.2(b) shows an example of service network, where S is presented as vertices in the network, and CR' is presented as edges.

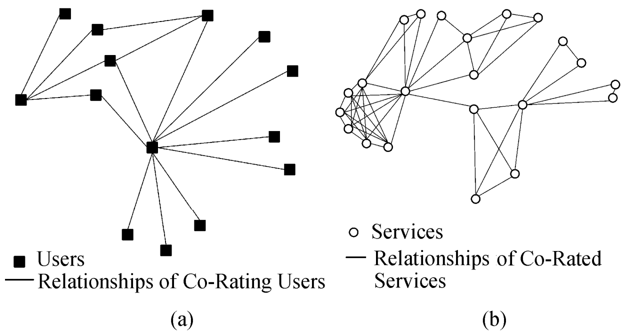


Fig.2. Unipartite networks projected from the trust network depicted in Fig.1. (a) User network. (b) Service network.

Fig.2(a) and Fig.2(b) present two sample unipartite projections of the bipartite network shown in Fig.1, one for the users and the other for the services. From them, we can see that the clustering tendency of a projected network can be attributed to the projection process itself to a large extent. If each user has rated multiple services, a fully connected complete subgraph in the projected service network will be resulted; therefore, the service network is actually comprised of complete subgraphs of this kind. In general, a unipartite network projected from a bipartite network is guaranteed to have large clustering coefficients, which may help improve the precision of recommendation in our research.

4 Trust-Based Service Recommendation

In this section, we propose a collaborative filtering algorithm called Trust-Based Service Recommendation (TSR). First, we formally define the problem of personalized service recommendation. We then describe the method for calculating the similarities between connected vertices in *U-Net* and *S-Net*. At last, we show the details of the TSR algorithm.

The TSR algorithm calculates the similarities of users through their ratings on common services. Higher similarity between two users implies that their preferences on QoS of the underlying Web services are more similar. Then a user's trust score of a service can be calculated based on the rating information derived from other users with high similarities to him/her. The output of the algorithm is a rank list of services based on their trust scores. Services with high trust scores will be recommended to the targeted user.

4.1 Problem Definition

TSR aims to recommend Web services with high trust scores to the target user. Trust value is a quantitative indicator that reflects the user's perception on the QoS of the underlying services. The trust value of a user on a service can be formally defined as follows.

Definition 4 (Trust Value). *Given a service* s *and a user* u, *the trust value of* s *from* u, *denoted as* $T(u, s)$, *is a metric which measures* u*'s subjective perception and personalized preference on multiple QoS properties of* s *including response time, throughput, price, and availability. If* u *has invoked* s *before, the value of* $T(u, s)$ *equals the latest rating of* s *given by* u. *In this study, we use ratings to represent trust values, so the range of trust value depends on the range of ratings. For the experiments, the trust value in this study ranges from* 1 *to* 5.

TSR is a trust-based collaborative filtering service recommendation algorithm. For a target user u_t, TSR first calculates values of $T(u_t, s)$, where s belongs to the set of services that u_t has never invoked before. Then TSR recommends services with high values of $T(u_t, s)$ to u_t. The algorithm has two phases: similarity calculation and trust-based recommendation.

4.2 Similarity Calculation

Similarity calculation is a key step in memory-based collaborative filtering algorithms[34]. It aims to capture the similarities among users' preferences on QoS of services by learning from their historic ratings. In TSR, the similarity calculation includes two parts: users' similarity and services' similarity.

With the similarity scores, we can learn the similarities of subjective perception and personalized preferences on multiple QoS of services from different users. Then the ratings from users with high similarities can make greater impact on the recommendation for the target user. For example (shown in Fig.3), the users a and b both rate high for the service s, but the user c makes the opposite rating. This may be caused by the differences of their preferences on QoS properties. Maybe users a and b rate high for fast response time, while user c rates low for low throughput. This may also be caused by the differences of subjective perceptions. Users a and b may think it is good if the service responds within one minute. But user c has much higher requirement and he/she would like the service to respond within one second. All in all, users a and b are more likely to have similar views on QoS of such type of services, then services that the user a likes can be recommended to the user b, or oppositely.

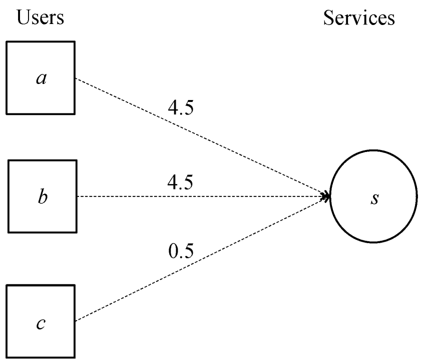


Fig.3. Example of co-rating users.

Pearson correlation, which has been widely used to measure the extent to which two entities linearly relate with each other[38], is adopted to calculate the similarities of users. Given two users u_i and u_j which are two connected vertices in *U-Net*, the similarity between u_i and u_j is:

$$SimU(u_i, u_j) = \frac{\sum\limits_{s \in S_c} (T(u_i, s) - \overline{T}_{u_i})(T(u_j, s) - \overline{T}_{u_j})}{\sqrt{\sum\limits_{s \in S_c} (T(u_i, s) - \overline{T}_{u_i})^2} \sqrt{\sum\limits_{s \in S_c} (T(u_j, s) - \overline{T}_{u_j})^2}}, \quad (1)$$

where the $s \in S_c$ summations are over the services that both users u_i and u_j have rated, $T(u_i, s)$ and $T(u_j, s)$ are trust values for s from u_i and u_j respectively, $\overline{T}_{u_i}$ and $\overline{T}_{u_j}$ are the average ratings of the co-rated services of users u_i and u_j respectively.

The calculation of services' similarities is also based on the Pearson correlation in item-based collaborative filtering. We measure the similarities between the invoked services of the target user and services to be recommended. The target user may have similar views on the services with high similarities. For example in Fig.4, the user u rates the services x, y, and z. In u's opinion, x and y may perform with the similar level of QoS. Then for another user v who has similar preference with the user u, if v rates x but has never invoked y, we may predict that v might give a similar score as x to y.

Then we utilize the Pearson correlation in item-based collaborative filtering to calculate similarities of services. Given two services s_i and s_j which are two connected vertices in *S-Net*, the similarity between s_i and s_j is:

$$SimS(s_i, s_j) = \frac{\sum\limits_{u \in U_c} (T(u, s_i) - \overline{T}_{s_i})(T(u, s_j) - \overline{T}_{s_j})}{\sqrt{\sum\limits_{u \in U_c} (T(u, s_i) - \overline{T}_{s_i})^2} \sqrt{\sum\limits_{u \in U_c} (T(u, s_j) - \overline{T}_{s_j})^2}}, \quad (2)$$

where $u \in U_c$ denotes the set of users who have rated both services s_i and s_j, $T(u, s_i)$ and $T(u, s_j)$ are trust values of user u on services s_i and s_j respectively, $\overline{T}_{s_i}$ and $\overline{T}_{s_u}$ are the average ratings of services s_i and s_j by those users respectively.

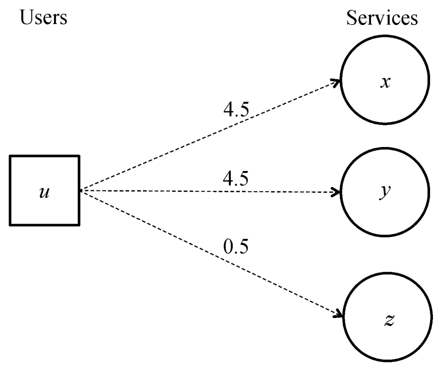


Fig.4. Example of co-rated services.

For two connected vertices in *U-Net* or *S-Net*, we can calculate their similarity based on (1) and (2). While for two vertices that do not connect to each other directly, we first find the shortest path between them, and then use the probability product to derive the similarity through the path. This is because that similarities for different vertices are calculated independently and the final similarity is affected by all the transitive similarities. This can also ensure the decrease of transitive similarities with the increment of the length of the path. Finally, we can calculate all the similarities between any vertices in *U-Net* or *S-Net*. The work of similarity calculation is carried out offline in order to reduce workload.

4.3 Trust-Based Recommendation Algorithm

Based on the similarity scores acquired through the method mentioned in the previous subsection, we can predict a user's preference on the QoS of specific service.

We then calculate the trust value and make personalized service recommendation. As similarities of connected vertices in *U-Net* or *S-Net* can both make contribution to the final recommendation, we adopt the user-based CF and the item-based CF to calculate trust value from users' perspective and services' perspective respectively.

With the user-based CF method, we calculate the trust value of each to-be-recommended service s_0 for the target user u_t from the perspective of *U-Net*. Then the user-based trust value of s_0 for the target user u_t is calculated by the following formula:

$$T_U(u_t, s_0) = \frac{\sum\limits_{u \in U_c} SimU(u, u_t) T(u, s_0)}{\sum\limits_{u \in U_c} SimU(u, u_t)}, \qquad (3)$$

where U_c denotes the set of the top-k most similar users that have rated s_0. For our experiments, we set $k = 10$. And the value of $T(u, s_0)$ equals the latest rating of s_0 given by users in the set of U_c.

Similarly, we utilize item-based CF method to calculate the trust value of s_0 for the target user u_t from the perspective of *S-Net*:

$$T_S(u_t, s_0) = \frac{\sum\limits_{s \in S_c} SimS(s, s_0) T(u_t, s)}{\sum\limits_{s \in S_c} SimS(s, s_0)}, \qquad (4)$$

where S_c denotes the set of the top-k most similar services that has been rated by u_t. For our experiments, we set $k = 10$. And the value of $T(u_t, s)$ equals the latest rating of s given by u_t.

In addition, we also provide a unified single-index performance measure for balancing these two types of trust values by using (5). The balancing of the two trust values is inspired by F-measure, which has already been adopted as a comprehensive evaluation metric for two independent metrics precision and recall in information retrieval area.

$$T(u_t, s_0) = \frac{2 \times T_U(u_t, s_0) \times T_S(u_t, s_0)}{T_U(u_t, s_0) + T_S(u_t, s_0)}. \qquad (5)$$

After calculating each trust value of services to be recommended for the target user, we get a rank list of services according to their trust values. Then the top-k services in the list will be recommended to the target user. The overall process of the proposed algorithm TSR is represented in Algorithm 1.

Algorithm 1. Algorithm of TSR

```
Input: T-Net: a trust network
       u_t: the target user
       k: the number of recommendations
       S_r: the set of services that can be recommended
Output: top-k recommended services for u_t
//Initialize
  U-Net = the user-based projection of T-Net
  S-Net = the service-based projection of T-Net
//Offline Stage
for each two vertices v_i, v_j in U-Net and S-Net
respectively {
   if v_i, v_j are connected
   | Calculate SimU(v_i, v_j) or SimS(v_i, v_j)
   else {
   |     Find the shortest path between v_i and v_j
   |     Similarity between v_i and v_j is calculated by
   |     multiplying the edge weights in the shortest
   |     path
         }
}
//Online Stage
for each s_0 in S_r {
      Calculate T_U(u_t, s_0) and T_S(u_t, s_0)
      T(u_t, s_0) = 2T_U(u_t, s_0)T_S(u_t, s_0)/(T_U(u_t, s_0)+
                    T_S(u_t, s_0))
}
Sort S_r in the order of trust value
return top-k services in S_r
```

5 Experiments and Analysis

In order to evaluate the accuracy and completeness of TSR's recommendations, we conduct a set of simulation-based experiments. TSR and related algorithms (for comparison purposes) are implemented using Java. 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 performance of TSR algorithm is then evaluated by observing the variance between the actual users' rating choices and the recommendations provided by TSR.

5.1 Setup

In this study, we use a real world dataset from WS-DREAM①[39]. The characteristics of the dataset are described in Table 1. 100 users invoked 2 000 real world services from public sources on the Web. Each user in-

①http://www.wsdream.net/, Nov. 2013.

voked each service 30 times. The behaviors of services in each interaction, as well as the observed QoS performance (response time and throughput) were recorded.

Table 1. Characteristics of the Dataset

Parameters	Values
Number of users	100
Number of services	2 000
Interaction times for each service per user	30
Observed QoS quality	Response time, throughput

Based on this dataset, we simulate users' ratings for each Web service. The generation of ratings is based on users' direct interactions. C_j is the compliance value of quality j of service i in an interaction with user x. Then, we calculate C_j for each service as follows:

$$C_j = \begin{cases} \dfrac{q_{pj} - q_{oj}}{q_{pj}}, \text{ if the lower value of } q_j, \text{ the better,} \\ \dfrac{q_{cj} - q_{pj}}{q_{pj}}, \text{ if the higher value of } q_j, \text{ the better,} \end{cases}$$

where q_{pj} is the promised QoS value of service i on quality j, q_{oj} is the observed value of j during the interaction. The conformance level C_{qj} of quality j is based on the value of C_j:

$$C_{qj} = \begin{cases} 1, & \text{if } -1 \leqslant C_j < -0.5, \\ 2, & \text{if } -0.5 \leqslant C_j < 0, \\ 3, & \text{if } C_j = 0, \\ 4, & \text{if } 0 \leqslant C_j < 0.5, \\ 5, & \text{if } 0.5 \leqslant C_j < 1. \end{cases}$$

For each interaction, we can get one value of C_{qj} on the quality j. Then we get $Avgh\_C_{qj}$ by averaging all the values of C_{qj} during all the interactions between the user x and the service i. With the calculated values of Avg_C_{qj} from each quality of service i, we can derive the rating of the service i from the user x.

$$Rating(x, i) = \frac{\sum_{j=1}^{n_i} w_{qj} \times Acg\_C_{qj}}{\sum_{j=1}^{n_i} w_{qj}},$$

where w_{qj} is the preference weight on quality j which ranges from 0 to 1. n_i is the number of QoS qualities of service i. In order to simulate the variety of preferences from different people, we divide the 100 users into 5 groups. They weight different values on response time w_{Rt} and throughput w_{Tp}. The partition is presented in Table 2. Then we calculate all the ratings for each service from each user, and generate a rating dataset $\boldsymbol{RD}$, which is a $100 \times 2\,000$ matrix.

Table 2. Preferences of Users

Users	w_{Rt}	w_{Tp}
Group 1	0.0∼0.2	$1 - w_{Rt}$
Group 2	0.2∼0.4	$1 - w_{Rt}$
Group 3	0.4∼0.6	$1 - w_{Rt}$
Group 4	0.6∼0.8	$1 - w_{Rt}$
Group 5	0.8∼1.0	$1 - w_{Rt}$

5.2 Evaluations

The evaluation study is conducted using the rating dataset $\boldsymbol{RD}$ which consists of ratings for each service from every user. We use part of $\boldsymbol{RD}$ as the training set, and treat the remaining part as the testing set. Then we make the training set as input, and recommend a rank list of k services based on the proposed method and other mechanisms. For each user, we recommend top-k services that he/she has never invoked in the training set, that is, the k services that the user would trust most. The recommendation quality is assessed according to the number o hits (recommendations that match the actual top-k services in the testing set). The precision is calculated to assess the quality of recommendations:

$$\text{precision}: \ P_u = \frac{\text{number of hits}}{k}.$$

The overall metric for the algorithm is derived by averaging values over all users tested. The precision is to assess the accuracy of the recommended services relative to the users' potential ratings. For example, 10% precision indicates that one out of 10 recommendations would actually be ranked in the top-10 by the target user.

We implement the proposed algorithm TSR and other five service recommendation methods for comparison as shown in Table 3. All of these methods can be classified into three types:

1) *Recommendation Based on Relationships Between Users and Services.* Besides TSR, we also implement a Reputation-Based Recommendation algorithm[25], which is a kind of global recommendation that calculates the reputation value of each service by averaging the ratings for the services from the invoking users and then recommends services with higher reputations.

2) *Recommendation Based on Link Analysis Mechanisms.* We choose HITS and PageRank to make service recommendation, which are most famous link analysis mechanisms widely used in network-based recommendation[40-42].

3) *Recommendation Based on QoS Prediction.* Most of existing service recommendation approaches are

based on this kind of methods[4,28]. Such methods first predict the future QoS of services based on historic observation, and then recommend services with high predicted values of QoS properties.

Table 3. Recommendation Approaches Compared

Types	Recommendation Approaches
Relationships-based	TSR Reputation-based
Link analysis based	HITS PageRank
QoS prediction based	QoS prediction on response time QoS prediction on throughput

We use TSR, Reputation, HITS, PageRank, Qos(Rt), QoS(Tp) for short to describe the above mentioned methods respectively in the following part.

5.2.1 Impact of Number of Links

In order to evaluate the impact of the numbers of links from users to services, we generate four training sets based on the numbers of services each user invoked. Fig.5 presents the distributions of users' links to services in the four training sets.

In the first three training sets, the numbers of services each user invoked were generated randomly with the upper bounds of 2 000, 1 000, and 500 respectively. For the last training set, we investigate a film rating dataset[43], capture the rating distribution of the film site, and make the last training set follow the distribution. Then we fix the number of recommendations, and then compare the precision with the four mentioned training sets. The numbers of recommendations are set to 10, 50, 100, 200, and 500 respectively. Fig.6 presents the results of comparison. For example, Fig.6(a) shows the comparison result of precision using different algorithms with the four training sets when 10 services are recommended.

From the results we can intuitively find that TSR and reputation-based recommendation behave better with more links. That is because the two methods are based on relationships between users and services. Hence, more links can provide more evidences to reflect the quality of services, and then the recommendation based on enough evidences can be more precise. In training set 4, the number of links is much less than the other three training sets, but TSR also performs better than others and not too worse than on the other three training sets. Besides, we can also find that QoS-based methods are less impacted by the number of links than other methods. The reason is that the two methods make more focus on the properties of individual services while neglect the relationships between users and services.

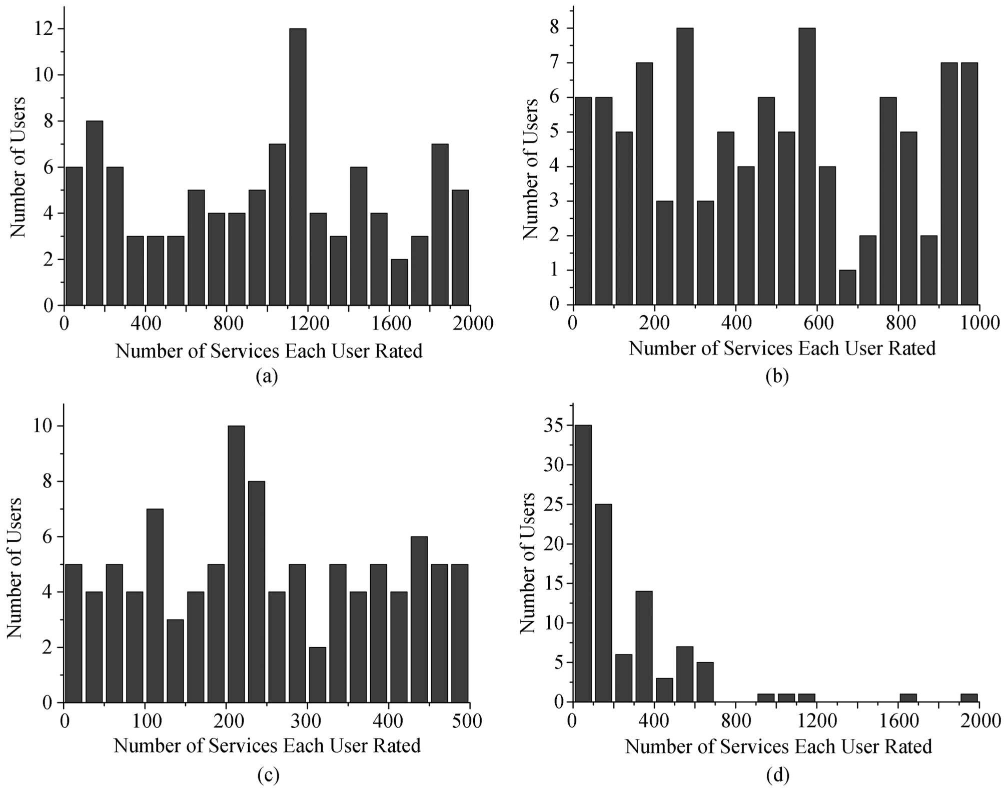


Fig.5. Distributions of the 4 training sets. (a) Training set 0. (b) Training set 1. (c) Training set 2. (d) Training set 3.

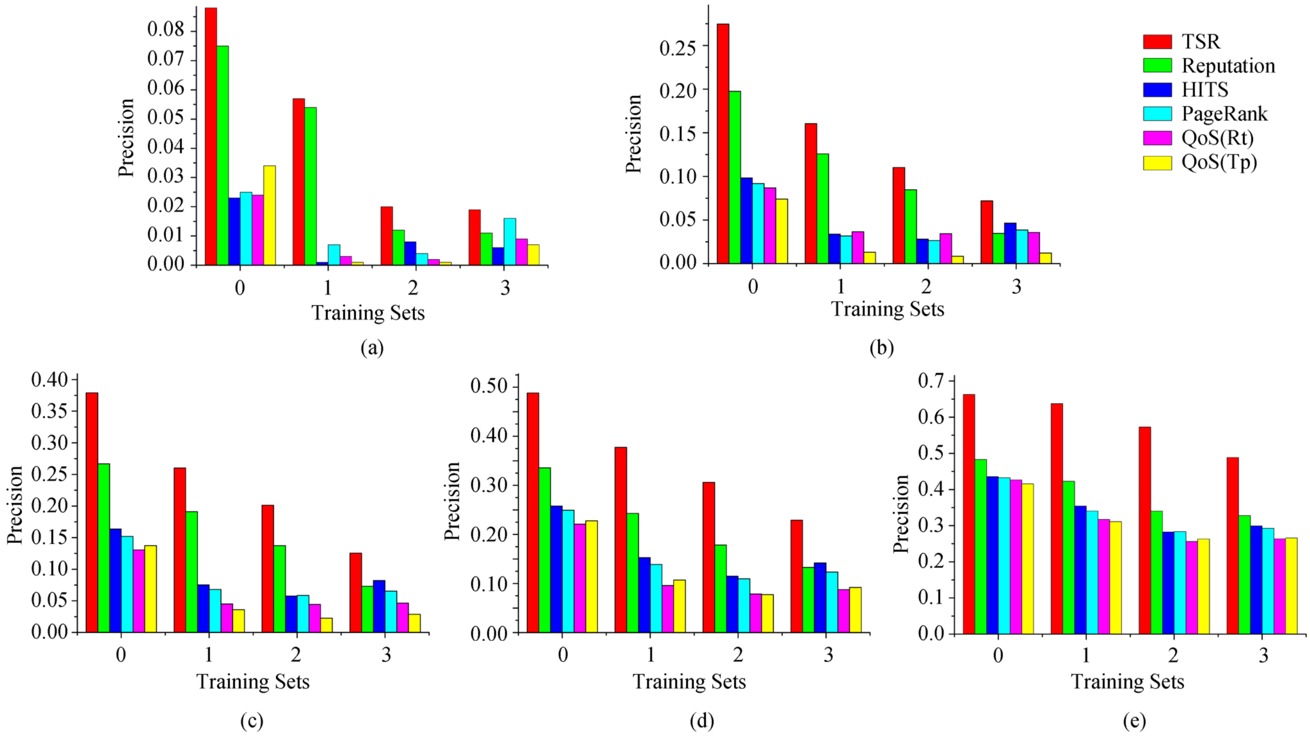


Fig.6. Precision with the 4 training sets. (a) Top-10. (b) Top-50. (c) Top-100. (d) Top-200. (e) Top-500.

5.2.2 *Impact of Number of Recommendations*

The experiments to this point show the effect of the number of recommendations. We verify the effect under the same training sets. The number of recommendations is set to 10, 50, 100, 200, and 500 respectively. The comparisons of precisions of the six mentioned methods with different numbers of recommendations are shown in Fig.7.

The results shown in Fig.7 indicate that the precision would be higher with more recommendations. That is because it may be difficult for the recommendation algorithms to recommend the optimal services for users, while it is relatively easier to recommend some suboptimal services. So when the number of recommendations becomes larger, these algorithms may return more suboptimal services to users, thus the precision becomes better with the growing number of recommendations. TSR behaves better than others with every number of recommendations. But when the number of recommendations is small (shown in Fig.7(a)), the advantage of TSR is not so obvious as in other cases. It shows that there is still much room for improvement.

5.2.3 *Summarized Comparison and Analysis*

Table 4 presents the overall experimental results. The performance data in the bold font indicates the best performance among all the mentioned mechanisms under study for the corresponding training set and number of recommendations. The precision of our method is about 10% higher than the second best.

Among the six mechanisms, TSR and reputation-based outperform the others. That is because the two methods both consider the relationships between users and services, which are neglected by others. The reason why TSR is better is that we consider not only the subjective of trust but also the customization of recommendation. Reputation-based method thinks that all the ratings from users have the same importance, which is not the case actually. Different users have different preferences, which makes the rating standard of each user different. So the ratings from users that have similar preference may give more contribution to making the final recommendation.

HITS and PageRank are classic link analysis algorithms. But they do not perform well in personalized service recommendations, since link analysis mechanisms only care about the quality of links and the quality of link sources. That means more links from better users will be recommended. Here, better users are the users who have rated more services that have high ratings. It is a kind of global recommendation rather than an individual one. Nevertheless, trust of Web services is individual and customized, so both methods cannot compete with our method.

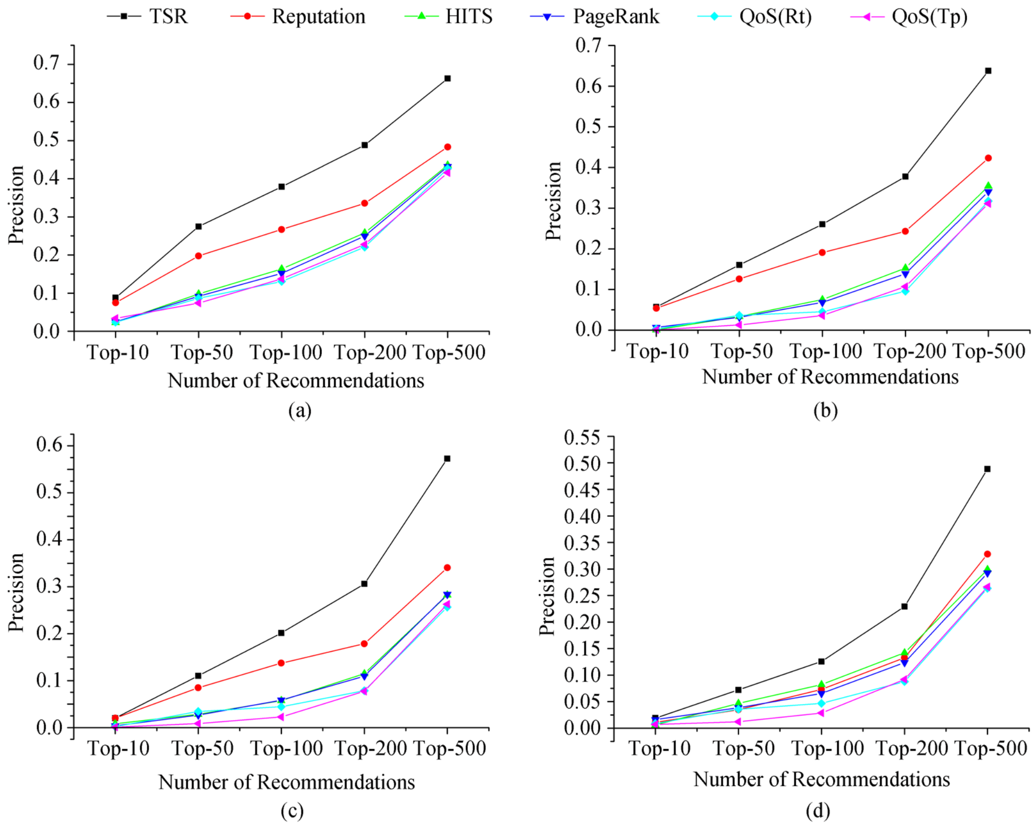


Fig.7. Precision with different numbers of recommendations. (a) Traning set 0. (b) Traning set 1. (c) Traning set 2. (d) Traning set 3.

Table 4. Overall Precision Comparison Results

Number of Recommendations (k)	Training Set	TSR	Reputation	HITS	PageRank	QoS(Rt)	QoS(Tp)
10	0	**0.088 0**	0.075 0	0.023 0	0.025 0	0.024 0	0.034 0
	1	**0.057 0**	0.054 0	0.001 0	0.007 0	0.003 0	0.001 0
	2	**0.020 0**	**0.020 0**	0.008 0	0.004 0	0.002 0	0.001 0
	3	**0.019 0**	0.011 0	0.006 0	0.016 0	0.009 0	0.007 0
50	0	**0.274 6**	0.197 6	0.098 2	0.091 8	0.086 8	0.074 0
	1	**0.160 4**	0.126 0	0.033 8	0.031 8	0.036 6	0.013 0
	2	**0.110 2**	0.084 8	0.028 2	0.026 6	0.034 4	0.008 6
	3	**0.072 0**	0.034 8	0.046 6	0.038 6	0.035 8	0.012 2
100	0	**0.379 0**	0.266 8	0.163 7	0.151 9	0.130 7	0.137 4
	1	**0.260 4**	0.191 0	0.075 3	0.068 2	0.045 2	0.036 0
	2	**0.201 4**	0.137 2	0.057 5	0.058 5	0.044 4	0.022 6
	3	**0.125 5**	0.073 0	0.082 1	0.065 4	0.046 6	0.028 7
200	0	**0.488 1**	0.335 8	0.257 9	0.249 6	0.220 8	0.227 8
	1	**0.377 6**	0.243 1	0.152 7	0.139 0	0.096 0	0.107 0
	2	**0.306 1**	0.178 4	0.114 8	0.109 7	0.078 6	0.077 4
	3	**0.229 2**	0.132 7	0.142 0	0.123 6	0.087 8	0.092 0
500	0	**0.662 7**	0.483 3	0.435 7	0.432 5	0.426 8	0.415 9
	1	**0.637 6**	0.423 0	0.354 4	0.340 7	0.317 5	0.311 5
	2	**0.572 9**	0.340 5	0.282 5	0.283 8	0.256 6	0.263 0
	3	**0.488 5**	0.328 1	0.299 0	0.292 6	0.263 6	0.266 2

The QoS-based methods can return the services which performed best in past, but the precision is not so good. That is because the best ones do not mean the most suitable ones. These methods only consider the properties of individual services while totally neglect users' perception. This kind of methods can return services that perform well on a certain property, such as response time and throughout. But they can hardly identify the specific properties that users expect and provide proper services that users really like. That is the reason why the both methods based on QoS observation behave worse than other compared algorithms.

6 Conclusions and Future Work

In this paper, we proposed a novel method to provide personalized service recommendations to individual users based on trust relationships between users and services. We utilized network modeling and analysis methodology to study trust relationships between users and services. Based on the findings and the proposed trust network model, we developed a trust-based service recommendation method based on collaborative filtering named TSR. The experimental results show that the recommendation precision of TSR improves with a great extent compared with other five compared approaches.

However, the dataset in this paper is not so sparse as in reality. In reality, the rating matrix can be very sparse and normal recommendation algorithms can hardly keep high precision. So our future research will focus on developing a mechanism on trust-based service recommendation that aims at solving the sparse problem of recommendation. And we will also carry out some researches on trust-aware service composition. Different from service recommendation, service composition needs to return an optimal composition of multiple services rather than an optimal single service, which is more challenging. We will also work on providing a formal benchmark for trust evaluation for services to assess the behavior of different approaches.

References

[1] Zeng L Z, Benatallah B, Ngu A H, Dumas M, Kalagnanam J, Chang H. QoS-aware middleware for Web services composition. *IEEE Transaction on Software Engineering*, 2004, 30(5): 311-327.

[2] Liang Q H, Wu X, Lau H C. Optimizing service systems based on application-level QoS. *IEEE Transaction on Services Computing,* 2009, 2(2): 108-121.

[3] Xiong P C, Fan Y, Zhou M C. QoS-aware Web service configuration. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 2008, 38(4): 888-895.

[4] Zheng Z B, Ma H, Lyu M R, King I. QoS-aware Web service recommendation by collaborative filtering. *IEEE Transactions on Services Computing*, 2011, 4(2): 140-152.

[5] Liang T P, Lai H J, Ku Y C. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems*, 2007, 23(3): 45-70.

[6] Yao J H, Tan W, Nepal S, Chen S P, Zhang J, Roure D D, Goble C. ReputationNet: A reputation engine to enhance ServiceMap by recommending trusted services. In *Proc. the 9th IEEE Int. Conf. Services Computing,* June 2012, pp.454-461.

[7] Yahyaoui H. Trust assessment for Web services collaboration. In *Proc. IEEE International Conference on Web Services*, July 2010, pp.315-320.

[8] Sheth A P, Gomadam K, Lathem J. SA-REST: Semantically interoperable and easier-to-use services and mashups. *IEEE Internet Computing*, 2007, 11(6): 91-94.

[9] Thio N, Karunasekera S. Web service recommendation based on client-side performance estimation. In *Proc. the 18th Australian Software Engineering Conference*, April 2007, pp.81-89.

[10] Zhang C, Han Y B. Service recommendation with adaptive user interests modeling. In *Proc. Distributed Computing and Internet Technology*, December 2007, pp.265-270.

[11] Brian M B, Nowlan M F. A Web service recommender system using enhanced syntactical matching. In *Proc. IEEE International Conference on Web Services*, July 2007, pp.575-582.

[12] Maamar Z, Mostefaoui S K, Mahmoud Q H. Context for personalized Web services. In *Proc. the 38th Annual Hawaii International Conference on System Sciences*, January 2005.

[13] Deng S G, Wu B, Yin J W, Wu Z H. Efficient planning for top-K Web service composition. *Knowledge and Information Systems*, 2013, 36(3): 579-605.

[14] Ge J K, Chen Z Q, Peng J, Li T F, Zhang L. Web service recommendation based on QoS prediction method. In *Proc. the 9th IEEE International Conference on Cognitive Informatics*, July 2010, pp.109-112.

[15] Zhu J M, Kang Y, Zheng Z B, Lyu M R. A clustering-based QoS prediction approach for Web service recommendation. In *Proc. the 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, April 2012, pp.93-98.

[16] Deng S G, Huang L T, Wu B, Xiong L R. Parallel optimization for data-intensive service composition. *Journal of Internet Technology*, 2013, 14(5): 817-824.

[17] Chen X, Zheng Z B, Liu X D, Huang Z C, Sun H L. Personalized QoS-aware Web service recommendation and visualization. *IEEE Transactions on Service Computing*, 2013, 6(1): 35-47.

[18] Zhang Y, Yu T. Mining trust relationships from online social networks. *Journal of Computer Science and Technology*, 2012, 27(3): 492-505.

[19] Wang S X, Zhang L, Wang S, Qiu X. A cloud-based trust model for evaluating quality of Web services. *Journal of Computer Science and Technology*, 2010, 25(6): 1130-1142.

[20] Du W, Cui G H, Liu W. An uncertainty enhanced trust evolution strategy for e-Science. *Journal of Computer Science and Technology*, 2010, 25(6): 1225-1236.

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

[22] Malik Z, Bouguettaya A. RATEWeb: Reputation assessment for trust establishment among Web services. *The VLDB Journal*, 2009, 18(4): 885-911.

[23] Nguyen H T, Zhao W L, Yang J. A trust and reputation model based on Bayesian network for web services. In *Proc. IEEE International Conference on Web Services,* July 2010, pp.251-258.

[24] Abawajy J H, Goscinski A M. A reputation-based grid information service. In *Proc. the 6th International Conference on Computational Science,* May 2006, pp.1015-1022.

[25] Huang L T, Deng S G, Li Y, Wu J, Yin J W, Li G X. A trust evaluation mechanism for collaboration of data-intensive services in cloud. *Applied Mathematics & Information Sciences*, 2013, 7(1L): 121-129.

[26] Paradesi S, Doshi P, Swaika S. Integrating behavioral trust in Web service compositions. In *Proc. IEEE International Conference on Web Services,* July 2009, pp.453-460.

[27] Rich E. User modeling via stereotypes. *Cognitive Science*, 1979, 3(4): 329-354.

[28] Shao L S, Zhang J, Wei Y, Zhao J F, Xie B, Mei H. Personalized QoS prediction for web services via collaborative filtering. In *Proc. IEEE International Conference on Web Services,* July 2007, pp.439-446.

[29] Zheng Z B, Ma H, Lyu M R, King I. WSREc: A collaborative filtering based Web service recommender system. In *Proc. IEEE International Conference on Web Services,* July 2009, pp.437-444.

[30] Rong W G, Liu K C, Liang L. Personalized Web service ranking via user group combining association rule. In *Proc. IEEE International Conference on Web Services,* July 2009, pp.445-452.

[31] Miller B N, Albert I, Lam S K, Konstan J A, Riedl J. MovieLens unplugged: Experiences with an occasionally connected recommender system. In *Proc. the 8th International Conference on Intelligent User Interfaces*, January 2003, pp.263-266.

[32] McLaughlin M R, Herlocker J L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proc. the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2004, pp.329-336.

[33] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th International Conference on World Wide Web*, May 2001, pp.285-295.

[34] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, Article ID 421425.

[35] Sun H F, Chen J L, Yu G, Liu C C, Peng Y, Chen G, Cheng B. JacUOD: A new similarity measurement for collaborative filtering. *Journal of Computer Science and Technology*, 2012, 27(6): 1252-1260.

[36] Hu D, Zhao J L, Cheng J. Reputation management in an open source developer social network: An empirical study on determinants of positive evaluations. *Decision Support Systems*, 2012, 53(3): 526-533.

[37] Huang Z, Zeng D D, Chen H. Analyzing consumer-product graphs: Empirical findings and applications in recommender systems. *Management Science*, 2007, 53(7): 1146-1164.

[38] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proc. ACM Conference on Computer Supported Cooperative Work*, October 1994, pp.175-186.

[39] Zhang Y L, Zheng Z B, Lyu M R. WSPred: A time-aware personalized QoS prediction framework for Web services. In *Proc. the 22nd IEEE International Symposium on Software Reliability Engineering*, Nov. 29-Dec. 2, 2011, pp.210-219.

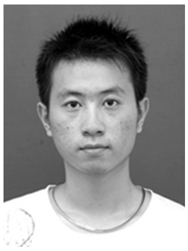
[40] Jiang F, Wang Z J. Pagerank-based collaborative filtering recommendation. In *Proc. the 1st Int. Conf. Information Computing and Applications*, October 2010, pp.597-604.

[41] Göksedef M, Gündüz-Ögüdücü S. Integration of the Pagerank algorithm into Web recommendation system. In *Proc. IADIS European Conference on Data Mining*, July 2008, pp.19-28.

[42] Peserico E, Pretto L. Score and rank convergence of HITS. In *Proc. the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2009, pp.770-771.

[43] Golbeck J, Hendler J. FilmTrust: Movie recommendations using trust in Web-based social networks. In *Proc. the 3rd IEEE Consumer Communications and Networking Conference*, January 2006, pp.282-286.

Shui-Guang Deng received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, in 2002 and 2007, respectively. At present, he is an associate professor in the College of Computer Science and Technology at Zhejiang University. His research interests include service computing, business process management and data management. Up to now, he has published more than 30 papers in peer-refereed journals and international conference proceedings as the first author or the corresponding author. And also, he has held a number of invention patents. He is a recipient of Microsoft Fellowship Award 2005. He is a member of ACM and IEEE.



Long-Tao Huang received the Bachelor's degree in software engineering from Zhejiang University, Hangzhou, in 2010. Now he is pursuing his Ph.D. degree in computer science and technology in Zhejiang University. His research interests include service computing and cloud computing.

Jian Wu received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, in 2004. He is currently an associate professor in Zhejiang University. His research interests include service computing, middleware techniques, data mining, and intelligent transportation.

Zhao-Hui Wu received a Diploma in computer science from Zhejiang University in 1988 and was a member of Sino-Germany Jointly Train Ph.D. Program from 1991 to 1993. He is a senior member of the IEEE. Currently, he is a professor in the College of Computer Science and Technology, Zhejiang University. His research interests cover the range of distributed artificial intelligence, grid computing, biometrics, embedded system, etc.