

## Research Article

# Trusted Service Scheduling and Optimization Strategy Design of Service Recommendation

**Xiaona Xia, Jiguo Yu, Shanshan Zhang, and Shu Wu**

*School of Information Science and Engineering, Qufu Normal University, Rizhao, Shandong 276826, China*

Correspondence should be addressed to Xiaona Xia; [xiagn@sina.com](mailto:xiagn@sina.com)

Received 12 August 2017; Accepted 9 November 2017; Published 5 December 2017

Academic Editor: Chang Liu

Copyright © 2017 Xiaona Xia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

More and more Web services raise the demands of personalized service recommendation; there exist some recommendation technologies, which improve the qualities of service recommendation by using service ranking and collaborative filtering. However, privacy and security are also important issues in service scheduling process; social relationships have been the key factors of interpersonal communication; service selection based on user preferences has become an inevitable trend. Starting from user demand preferences, this paper analyzes social topology and service demand information and obtains trusted social relationships; then we construct the fusion model of service historical preferences and potential ones; according to social service recommendation demands, TSRSR algorithm has completed designing. Through experiments, TSRSR algorithm is much better than the others, which can effectively improve potential preferences' learning. Furthermore, the research results of this paper have more significance to study the security and privacy of service recommendation.

## 1. Introduction

Web service recommendation is an important research topic in today's world with more and more Internet resources. In recent years, social networking platform and electronic transaction process have made new breakthroughs; in particular, cloud computing and big data applications are more deeply and fully implemented, which blooms the diversified demands of service recommendation process [1–3]. From the mass services, we choose the suitable candidate services and find the potential demand preferences so as to meet the functional and nonfunctional requirements. In many cases, the users do not know in advance the potential preferences of their own and do not know the multisources to recommend services. Therefore, it is very important to find and recommend effective services [4].

At present, the recommendation system is applied in various fields, and the system performances are mainly affected by recommendation algorithms [5]. There are two kinds of traditional recommendation algorithms: influence ranking and collaborative filtering. With recommended technology based on the influence ranking recommended services with more audiences to the demanded users [6], they were

needed to survey and analyze the relevant influences, such as reputation, HITS (Hyperlink-induced Topic Search) [7], and PageRank [8]. Collaborative filtering was widely used in service recommendation and helped the demand users to find similar users. Secondly, similar users recommended services with better evaluation parameters to demand users that considered two aspects, user-based and item-based [9]. Although the recommendation technology has not been fully applied, it achieved good results, while there are still some problems about its research and demonstration: (1) the existing recommendation technology does not take into account the trusted user social relationships; (2) cold start problem still exists.

This paper pays attention to seeking the effective trusted friend relationships and builds service demand conditions to drive effective service scheduling and strategy optimizing. So our work is as follows. The first part of this paper is an introduction, the second part is the related work, the third defines the trusted service location and incentive rules, the fourth argues trust relationships based calling mechanism, the fifth designs the corresponding novel algorithms, the sixth part expands the experiment implementation and simulation evaluation, and the last summarizes the whole paper.

In the process, we will analyze the histories of services and capture the potential demands of users. Secondly, normally users will tend to consider the recommendation of his trusted users, based on the demand characteristics; the recommendation process could calculate trusted friend relationships and recommended conditions. The user feature vector is constructed to calculate the similarities between users and other users. Meanwhile, we will select Top- $K$  users for the current demands, calculate the ranking of candidate services, and recommend the ranking of Top- $N$  service. In the experiments, we decide to select WSDream-QoS Dataset, which consists of 98701 users, 74012 potential credible demand preference relationships, 105199 use evaluation or feedback remarks, and a total of 352048 services. We will find that the proposed schemes are better than other algorithms and get better performances with accuracy and reliability.

## 2. Related Work

There are more and more service resources on the network; how to find suitable Web services for the users is becoming more and more difficult. The traditional service discovery and selection methods rely heavily on explicit user demand description; this approach for large specification of services and the increasing demands is inefficient, which is not useful for potential demands. Therefore, service recommendation strategies and algorithm optimization have great influences on the accuracy and reliability of service recommendation. There have been a lot of research results related to service recommendation [10, 11].

Since Rich proposed collaborative filtering (CF), a large number of scholars and industrial applications have been fully focusing on its effectiveness [12]. Shao et al. proposed a user-based CF algorithm; Pearson correlation coefficient was used to calculate the similarities between users [13]. Users with similar historical services have the same potential demand preferences. For a demand user, by considering the characteristics of the similar demand preferences, we can add or inspire the current incomplete demand preferences. According to the prediction of Web services, [14] designed a hybrid collaborative filtering algorithm, which realized the combination of Pearson correlation coefficient and user parameters [14]. However, the design and improvement of these algorithms cannot solve the cold start problem. If a user does not have the experience of services, these methods are inefficient for the service recommendation of new users.

To improve the accuracy of service recommendation, a lot of improved algorithms are constructed. In [10], probability decomposition achieved social service demands of classified features. To find similar users for one user, it was no longer necessary to consider all the users; we can mine initial similar users. But this approach did not take into account the geographical location. Reference [15] considered the influences of personalized services. Reference [16] suggested learning the overall experiences in recommending services, environmental factors, and user input data. Reference [17] designed an efficient public auditing protocol with novel dynamic structure for cloud data that could effectively improve the qualities of cloud service recommendation.

TABLE 1: The analytical data of Figure 1.

Users invoking time	Service 1	Service 2	Service 3	Service 4	Service 5
$U1$	6		3		7
$U2$		2		4	
$U3$		3	7		
$U4$				10	
$U5$	4				5

Reference [18] proposed a self-adaptive ABC algorithm based on the global best candidate, which would have a good reference significance for global demands. With the help of semantic ontology WordNet, [19] built a user interest model for individual user by analyzing the demand histories. Reference [20] proposed a novel  $K^+$ -isomorphism method to achieve  $K$ -anonymization state among subgraphs or detected communities in a given social network. Reference [21] designed a recommended method of social network service based on the consideration of the user's historical service trigger behavior, user preferences, and trusted relationship.

But the user's trusted relationship is based on the existing relationships [4, 5]; starting from the social attributes of users, the cold start problem is not solved; for users and services with not many historical data, adaptive recommendation is still difficult [7, 22]. In this paper, the trusted service relationship is settled on the basic situation of services; self-organization of trusted relationships is captured and measured for the service users.

## 3. Trusted Service Location and Incentive Rules

To illustrate the trusted relationship between services and demand preferences, to calculate the influences of service recommendation, we give an example, in Figure 1; there are 5 services, as well as their historical record information trigger.

Table 1 shows statistics of the call records of Figure 1. Service 1 is called, respectively, by  $U1$  and  $U5$  for 6 times and 4 times,  $U2$  and  $U3$  have called Service 2, and  $U1$  and  $U3$  call service 3.  $U4$  and  $U2$  call service 4,  $U1$  and  $U5$  call service 5, and the historical information is as shown in Table 1. For example, it is necessary to recommend potential services to  $U2$ , which need to calculate the service calling similarity between  $U2$  and other four users.  $U3$  and  $U4$  have service calling similarity with  $U2$ ;  $U4$  has only called Service 4; in addition to Service 2,  $U3$  also calls service 3; the potential service for  $U2$  is firstly to consider Service 3. Furthermore,  $U3$  and  $U1$  have the service calling similarity; the historical records of  $U1$  include Service 1, Service 3, and Service 5; the calling frequencies were, respectively, 6, 3, and 7; the potential service recommendation for  $U2$  would consider Service 5, Service 1, and Service 3. Based on the similarity of service calling, we can capture similar users and then select the candidate services by the user's service calling records.

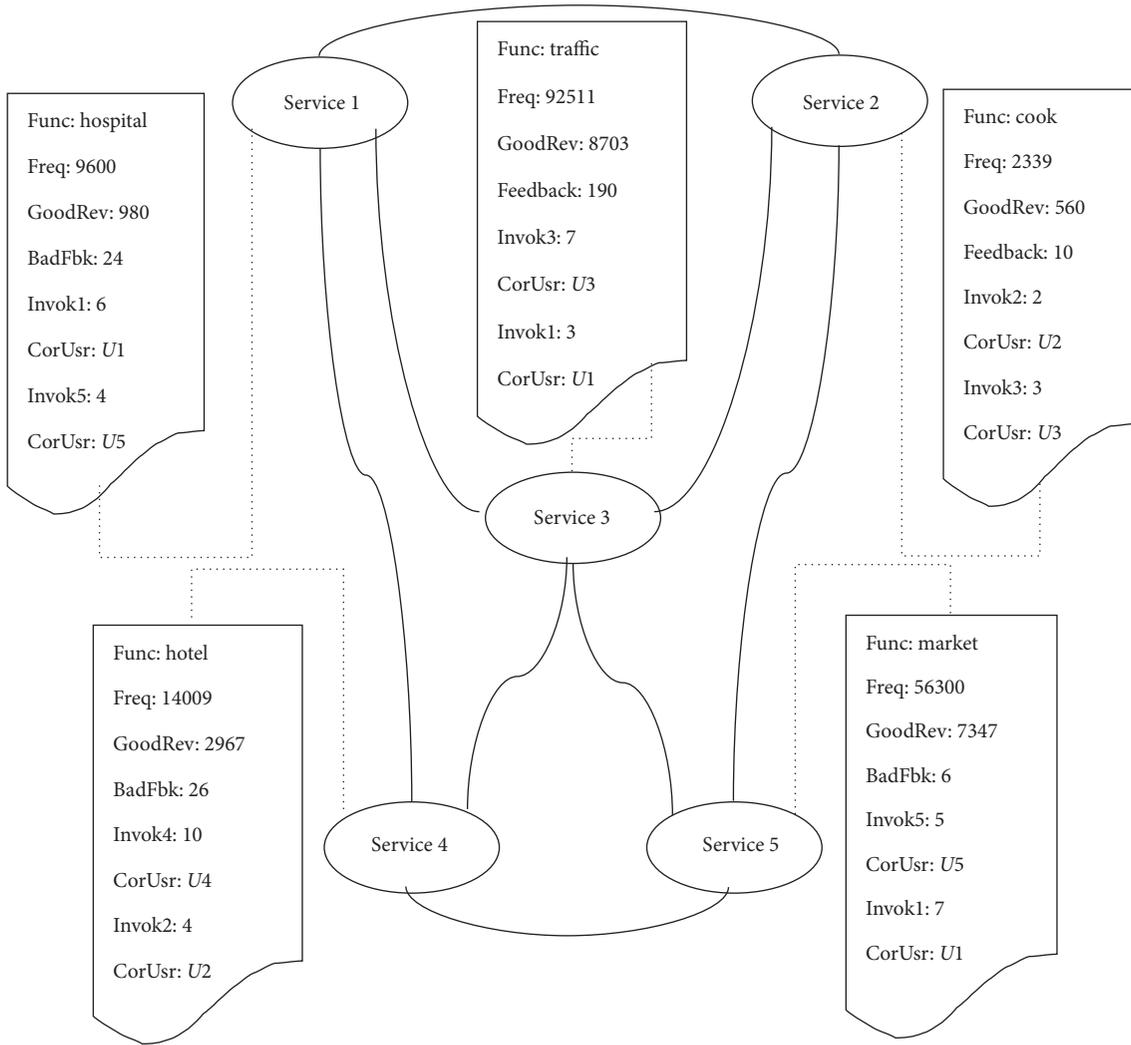


FIGURE 1: Example of service call.

New users are needed to refer two indicators, one is the recommended service calling ranking, and the other is user ranking; select the Top- $K$  users and services of candidate users; then complete sorting. For new services, we consider three indicators (one is user ranking about the overall frequency of service calling) and complete Top- $M$  user ranking (another is service calling frequency), through the service calling condition of users; we achieve the candidate users' Top- $N$  ranking.

According to the internal relationship of the trusted service calling, it is not sufficient to establish the service relationships among users. In the above example, this is more important for the relevance of the services and the evaluation of service users. In the popularity of social networks, people like sharing their social experiences. Based on the users' social relationships, we will define the calling and optimizing strategies of candidate services, not only from the users' social relationships while taking into account the user history service calling behaviors and user relationships.

#### 4. Trust Relationships Based Calling Mechanism

About the service recommendation framework based on social relationships, the whole business flow is shown in Figure 2. First of all, according to the histories of service calling and the trust relationships among users, the service preferences of current active users would be learned. Second, the similarities between the current users and other users are calculated, and other users with similar preferences are selected to serve the neighboring users. Third, according to the service calling of neighboring users, the ranking of the candidate services could be calculated. Finally, higher ranking services are recommended to the current active users.

4.1. *Related Definitions.* Before starting the work of this paper, here we show several related definitions.

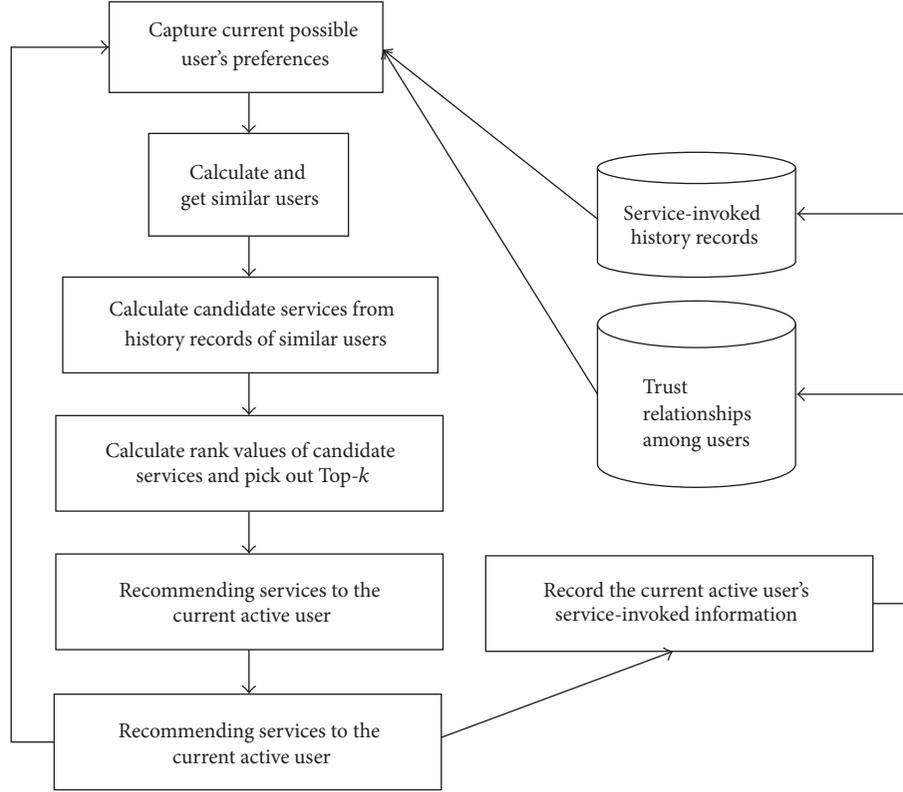


FIGURE 2: Service recommendation based on user social relationship.

**Definition 1** (service history record). In one service recommendation system, there are  $m$  users and  $n$  services, which are represented as  $U = u_1, u_2, \dots, u_m$  and  $S = s_1, s_2, \dots, s_m$ . When one user is calling a service, according to the demand target, he completes the classification of services, so  $u_i$  about a service calling historical records can be represented as  $hs_i = (s_{i1}, \text{category}_j, \text{rate}_{i1})$ ; the user's service historical records are shown as  $HS_i = (ls_{i1}, ls_{i2}, \dots, ls_{ik})$ .  $s_{i1}$  means that  $u_i$  has called  $s_1$ ;  $\text{category}_j$  represents the functional category of service;  $\text{rate}_{i1}$  represents the evaluation rate of  $u_i$  about  $s_1$ .

**Definition 2** (trusted relationship of users). There are two types of trust relationships between users: explicit and implicit. Explicit trusted relationships mean that the user can clearly recognize the trusted relationship between users, because stable cooperative relationships in a certain period of time and social relationships could be trusted and expressed as  $u_1 \propto u_2$ , which means  $u_1$  trusts  $u_2$ ; there is mutual trust between  $u_1$  and  $u_2$ , expressed as  $u_1 \infty u_2$ . For the implicit trusted relationships, through mining service calling similarities or the historical calling records, we determine that  $u_1$  provides a good reference for  $u_2$ . The combination of implicit trust and explicit trust can be used to characterize the trusted relationships among users.

**Definition 3** (user service preference vector). The service preference vector of  $u_i$  is used to represent the user's preference classification for service calling, which is represented

as  $pV_i = \langle \text{category}_1, \text{category}_2, \dots, \text{category}_p \rangle$ , and  $p$  is the number of classification.

**4.2. Construct Service Preference Vector.** Based on the above, we can initialize the service preference vector, that is, to initialize the classification of services. Given  $u_i$ , modeling  $pV_i$  includes the following steps: first, search all historical calling services, obtain the historical calling records  $ls_q$  of  $s_q$ , learn  $ls_q$ , and get category of  $s_q$ , until realizing all category of  $LS_i$ . Second, category of  $pV_i$  is increased by 1, iteratively. Third, if there is no user calling information, but trusted service preference vector and  $pV_i$  are needed to merge together, the calculation formula is expressed as

$$pV_i = pV_i + \mu \sum_{k \in U_i} pV_k. \quad (1)$$

$\mu$  ( $0 < \mu \leq 1$ ) is a balancing factor used to balance the influences of the user's service preference vector;  $U_i$  is a collection of other users trusted by  $u_i$ .

**4.3. Similarity between Users.** About the key step to calculate the demand preferences between users, we use Euclidean distance to measure the similarity; the calculation formula is

$$\text{sim}(u_i, u_j) = \sqrt{\sum_{k=1}^p (pV_{ik} - pV_{jk})^2}. \quad (2)$$

Euclidean distance is a common distance definition function that can approach the real distance between points in  $n$ -dimensional space.  $u_i$  represents the current demand user,  $u_j$  are the other trusted users of  $u_i$ ,  $u_j \in U_i$ .

**4.4. Ranking of Candidate Services.** After the calculation of the similarity of the demand preferences between the current user and other users, it is necessary to calculate the ranking of the candidate services; the corresponding services' scores are needed for calculating In order to calculate the similarity of demand preferences, we first retrieve the history service calling records of each similar user and select the services that the current user has not yet called. Second, using formula (3) calculates the ranking:

$$rs_{s_k} = \sum_{j \in \text{set}_{s_k}} \left( \text{sim}(u_i, u_j) \times \text{rate}_{u_j \rightarrow s_k} \right). \quad (3)$$

$s_k \in ((\bigcup_{j \in T_{u_i}} HS_j) - HS_i)$ ,  $T_{u_i}$  is the trusted friend set of the current user,  $\text{set}_{s_k}$  is the user set that calls  $s_k$ , and  $\text{rate}_{u_j \rightarrow s_k}$  is the evaluation rate of  $s_k$  by  $u_j$ . Third, the classification of the candidate services is completed according to the demand preference category of services, and the service with high ranking is recommended to the current user.

## 5. Recommended Optimization Algorithm

By learning the history records of services, we can obtain the trusted user social relationships and establish sensitive service preferences. Because different users have different preferences, which have different tolerance, we need to consider the social relationship intensity of users.

In this part, we integrate the trusted social relationships and the service recommendation algorithm by introducing the trusted social relationships into the recommendation process. To verify the effectiveness of the proposed algorithm, recall, precision,  $F$ -Measure, and ranking are as the evaluation parameters, and we select UserCF, ItemCF, Hits, Reputation, and PageRank as the compared algorithms.

We select WSDream-WSDL Dataset that comes from a famous business platform and could provide reliable consumer goods records, valuable user opinions, suggestions, and profound product evaluation and personalized recommendation. In the experiment, we capture the relevant records of the data set, including the users' trusted social relationships, user comments, and feedback.

Trusted social relationships of other users are constructed based on the trusted records of the active users, including service reviews and rankings, which are captured through the continuous search and demonstration. According to the trusted social network, the service platform predicts which records are useful. Therefore, users can easily find the services they need on the service platform. About WSDream-WSDL Dataset, we crawled 100809 services and 788904 relationships, in the establishment of the topological relationships, service is corresponding nodes, and the relationships are corresponding edges.

Users review services that are purchased and used, so the users can provide more valuable service records that are

TABLE 2: Two types of service resources.

	Recommended	Unrecommended
Comment	Correct Hits (CH)	inCorrect unHits (iCuH)
Uncomment	inCorrect Hits (iCH)	Correct unHits (CuH)

generally divided into four types: very useful, useful, and basically useful, useless, given four grades: 3, 2, 1, and 0.

**5.1. Metrics.** Service recommendation is to produce a ranking list of Top- $k$ ; recommendation results are based on service hit rate and the ranking; the metrics include recall rate, precision,  $F$ -Measure, and ranking.

(1) *Recall Rate.* Service recommendation involves two types of service resources: commented and not commented; these two types are also involved in the activities of the recommendation of the two categories: recommended and not recommended (that is shown in Table 2). Recall rate (RR) means that the correct hit ratio of services is commended by users and expressed as

$$RR = \frac{\text{value (CH)}}{\text{value (CH)} + \text{value (iCuH)}}. \quad (4)$$

(2) *Precision.* Precision is the percentage of correct hit service and all predictable candidate services, including new candidate services and services that had the historical comments that cannot care whether or not the services are recommended. Precision rate (PR) is expressed as

$$PR = \frac{\text{value (CH)}}{\text{value (CH)} + \text{value (iCH)}}. \quad (5)$$

(3) *F-Measure.*  $F$ -Measure is the measure accuracy of data statistics. It considers recall rate and precision rate; that is to say, it is the weight average of recall rate and precision, the best value is 1, and the worst is 0. Its formula is shown as

$$F\text{-Measure} = \frac{2 \times RR \times PR}{RR + PR}. \quad (6)$$

Recall that precision and  $F$ -Measure are used to measure the correlation and coverage of recommending services to potential users; recall and precision are indicators of potential competition,  $F$ -Measure is the harmonic mean of recall and precision and can provide the performance of a single index measure to balance the defects of recall and precision.

(4) *Ranking.* There are a number of ways to evaluate the ranking of service recommendation, which is used to evaluate the recommended weight of the candidate services. For one service demand user, there are two service lists that have similar recommended set and the same recommended performances (such as recall, precision, and  $F$ -Measure); if the subscript of recommended service is smaller than others in another list, we will recommend the service with smaller

subscript, because smaller subscript means the high ranking. The ranking is calculated by the following formula:

$$\begin{aligned} \text{Rank} &= 100 \\ &\times \frac{\sum_c \text{Rank}_c}{\int_c \text{Rank}_c^{\max}} \left( \text{Rank} = \sum_j \frac{q_{cj}}{2(j-1)/(h-1)} \right). \end{aligned} \quad (7)$$

$j$  is the subscript of ranking,  $h$  is a balance coefficient,  $c$  is the service demand user,  $q_{cj} = \{1, \text{If } j \text{ is in candidate service set}; 0, \text{Otherwise}\}$ , and  $RS_c^{\max}$  is the max ranking of the recommended services list for  $c$ .

## 6. Experiment Implementation and Simulation Evaluation

In the experiment, user comments and feedback of service set are divided into training set and validation set according to time stamp. The time stamp is based on the rule that the training set scale is equal to the scale of the validation set. So two data sets are tested, respectively. The balance coefficient is introduced into the calculation process of recall, precision,  $F$ -Measure, and ranking, in order to test the effect of service calling. We define this balance coefficient for arithmetic distance which is 0.2; if Top- $K$  of service recommendation is set to 10, the number of users with similar services is 50; the experimental result is shown in Figure 3; the abscissa stands for  $\mu$ ; we can see that the four algorithm performance metrics will increase with the balance coefficient increased, when the  $\mu$  reaches 1, four performance parameters basically reach the maximum, and the historical service calling will affect the parameters, the more the historical service calling is, the better the parameters will be. Our service recommendation optimization algorithm is based on the historical preferences and potential preferences, so in the following experiments,  $\mu$  will be set as 1.

**6.1. Trusted Service Recommendation Algorithm.** Based on trusted service calling, adaptive recommendation is effectively realized. Here we give the relevant definitions and algorithm process; our algorithm is named TSRSR (trusted service relationship based service recommendation). First, the parameters of the algorithm are given, as shown in Notations.

Through several iterations, TSRSR algorithm gets the recommendation results, TSRSR gets a target user  $u_0$  in the first iteration, the  $k$ th iteration, and implementation process will arrive at a node  $u$ . If  $u$  has the ranking of  $s$ , the ranking of  $s$  for  $u$  is final result; in other cases, the iterative process has the following characteristics.

(1) TSRSR algorithm will end at  $u$  whose probability is  $\varphi_{u,s,k}$ , based on  $F_u(s_i)$ ; it selects  $s_i$  from  $RS_u$ . The ranking of  $s_i$  for  $u$  is the iterative result.  $\varphi_{u,s,k}$  is calculated by

$$\varphi_{u,s,k} = \max_{s_i \in RS_u} \text{sim } S(s_i, s) \times \frac{1}{1 + e^{-k/2}}. \quad (8)$$

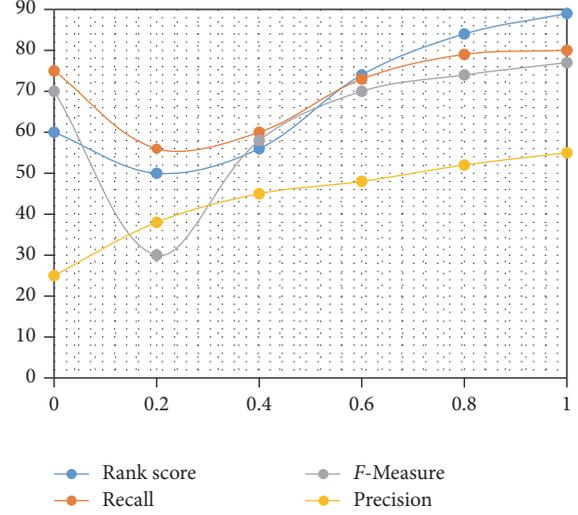


FIGURE 3: Effect of  $\mu$  on performance parameters.

$\text{sim } S(s_i, s)$  is the similarity formula of  $s_i$  and  $s$ . The Sigmoid function of  $k$  can provide values between 0 and 1. The similarity of two services is calculated as

$$\text{sim } S(s_i, s_j) = \frac{s_i \cdot s_j}{|s_i| \cdot |s_j|}. \quad (9)$$

The probability formula that  $s_i$  is selected and recommended to  $u$  is represented as

$$F_u(s_i) = \frac{\text{sim } S(s, s_i)}{\sum_{s_j \in RS_u} \text{sim } S(s, s_j)}. \quad (10)$$

(2) On the contrary, based on the probability of  $1 - \varphi_{u,s,k}$ , TSRSR algorithm will carry on the corresponding iterations. A user is selected from the user set as the target node  $v$  for the next iteration, and the user is directly trusted by the current user  $u$ . In a trusted relationship, whether or not  $v$  is selected, similar demand users can be calculated according to the formula

$$E_u(v) = \frac{\text{tr}(u, v)}{\sum_{x \in TU_u} \text{tr}(u, x)}. \quad (11)$$

Among them,  $\text{tr}(u, v)$  is the trust correlation degree between users [8]; the trust correlation degree ensures that each iteration of the algorithm can select other trusted users whose demands are similar to the current user.

The implementation process segment of TSRSR algorithm is described as Algorithm 1.

Figure 4 is an implementation result of TSRSR algorithm; there are seven user nodes in the graph; the weight of each edge represents  $E_u(v)$ . Assume  $s_3$  is recommended to  $u_1$ . In the first step, if  $E_{u_1}(u_3)$  is relatively large,  $u_3$  may be selected as the target user node. If the ranking of  $s_3$  is  $r$  for  $u_3$ ,  $r$  is the result of the algorithm implementation. Otherwise, as long as the conditions of the algorithm are not satisfied, it would

```

Input:  $U$  //User set
       $S$  //Service set
       $RM$  //Rating Matrix
       $wUSN$  //weighted User Social Network
       $u_0$  //Target User
       $ts$  // to be recommended target service

Output:  $pR$  // predicted Rating
(1) Set  $k = 1$ ; // Number of iterations for service selection
(2) Set  $u = u_0 //u_0$  as the initial node
(3) Set max – iteration = 7 //The maximum number of iterations is set 7
(4) Set  $pR = 0$ 
(5) While ( $k \leq \text{max – iteration}$ ){
(6)  $v = \text{selectUser}(u)$  //According to  $E_u(v)$ ,  $v$  in  $TU_u$  is selected as the next the next iterative target
(7) If ( $v$  has rated  $s$ ){
(8)    $r = r_{u,s}$ 
(9)   Return  $r$ 
(10) }
(11) Else {
(12)   If ( $\text{random}(0, 1) < \varphi_{u,s,k} \parallel k == \text{max – iteration}$ )
(13)     // Stop at current node
(14)     {
(15)        $s_i = \text{selectService}(u)$  //According to  $F_u(s_i)$ ,  $s_i$  is selected from  $RS_u$ 
(16)        $r = r_{u,s_i}$ 
(17)       Return  $r$ 
(18)     }
(19)   Else  $k + +$ 
(20) }
(21) }
(22) Return  $r$ 

```

ALGORITHM 1: TSRSR (trusted service relationship based service recommendation).

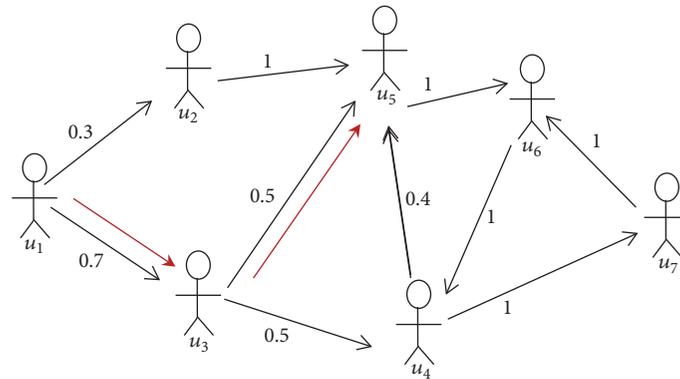


FIGURE 4: Example of TSRSR algorithm implementation.

continue to iterate. In the second step,  $u_5$  is selected; the process of selection will retrieve whether or not the ranking of  $s_3$  is denoted by  $u_5$ . If the iterations end, but  $u_5$  has no complete ranking of  $s_3$ , highest similar services are selected from the service ranking top of  $u_5$ . The probability of each service is obtained according to  $F_u(s_i)$ , and the ranking of the selected service will be the output of TSRSR.

**6.2. Algorithm Performance Comparison.** In order to measure the performances of TSRSR, in this section, we will select five compared approximate service recommendation algorithms,

respectively, UserCF, ItemCF, Hits, Reputation, and PageRank, and the scale of Top-K is set to 5, 10, 15, and 30. Figure 5 shows the experimental simulation results; we can see that the TSRSR is superior to other algorithms, and the ranking verification shows that the TSRSR is more conducive to the current demand users to trigger the recommended services.

Figures 5–8 show the results of the five algorithms on the performances, from which we can get the performances of TSRSR. At the same time, Figure 8 indicates that the social trust model can be triggered and invoked earlier by the current demand users.

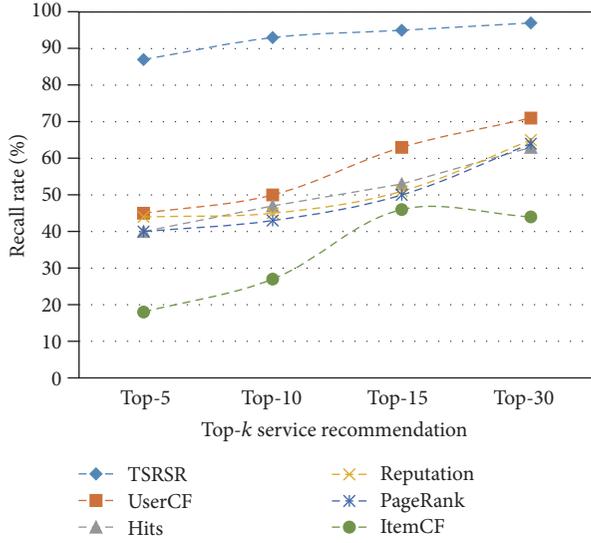


FIGURE 5: Recall rate.

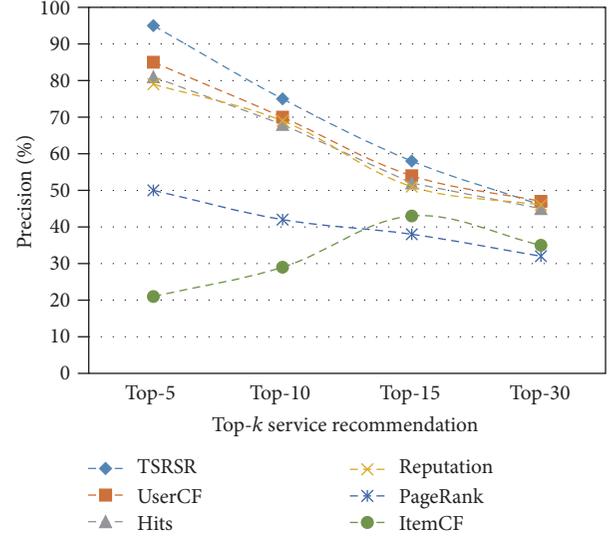


FIGURE 7: Precision.

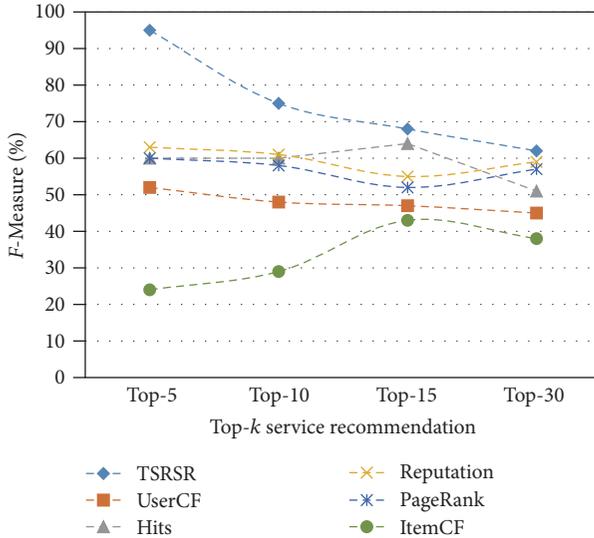
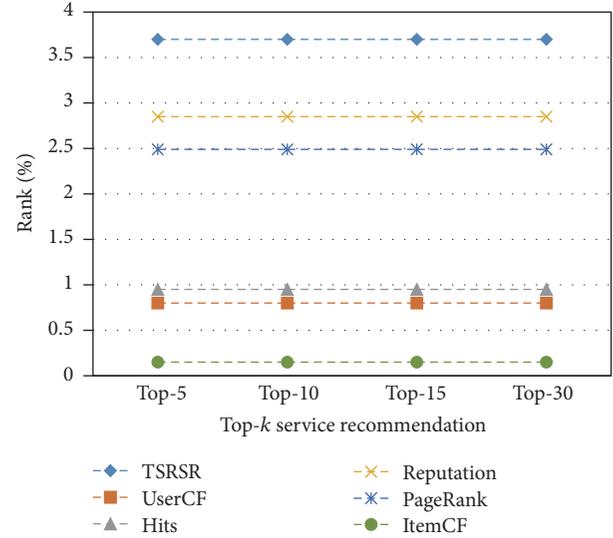
FIGURE 6:  $F$ -Measure.

FIGURE 8: Rank score.

## 7. Conclusion

This paper proposes a recommendation algorithm TSRSR; before algorithm is constructed, the trusted service relationships are modeled and calling rules were designed, which realized the integration of historical preferences and potential preferences that is used in WSDream-QoS Dataset with a large scale; then we complete the comparison of approximate algorithms. Through a series of experiments, TSRSR in service recommendation achieved good performances and had obvious advantages in the recall rate, precision,  $F$ -Measure, and ranking.

In the following work, we will focus on the following questions: (1) integration of explicit and implicit trust social relationship model; (2) construction of the autonomous learning model of service user preferences and application.

## Notations

$r_{u,s}$ :	Ranking of $s$ for $u$
$P_{u,s}$ :	Potential recommended ranking of $s$ for $u$
$\varphi_{u,s,k}$ :	Probability of $s$ in Top- $K$ of recommended list for $u$
$TU_u$ :	Other user sets trusted by $u$
$E_u(v)$ :	Probability of $v$ as the candidate approximate user in $TU_u$ for $u$
$\text{sim } S(s_i, s_j)$ :	Similarity between $s_i$ and $s_j$
$RS_u$ :	Historical service set recommended to $u$
$F_u(s_i)$ :	Recommended probability of $s_i$ in $RS_u$ .

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors thank Lianyong Qi (Qufu Normal University) for helpful comments on earlier drafts of this manuscript. This work is supported by the National Natural Science Foundation of China (61672321, 61373027, and 61402258) and the Natural Science Foundation of Shandong Province of China (ZR2016FM45, ZR2012FQ011).

## References

- [1] A. Bouguettaya, "Meta-Path Based Service Recommendation in Heterogeneous Information Networks," in *Proceedings of the 14th International Conference on Service-Oriented Computing, ICSOC 2016*, vol. 9936, article 371, Springer, Banff, Canada, October 2016.
- [2] Q. Xie, S. Zhao, Z. Zheng, J. Zhu, and M. R. Lyu, "Asymmetric correlation regularized matrix factorization for web service recommendation," in *Proceedings of the 23rd IEEE International Conference on Web Services, ICWS 2016*, pp. 204–211, July 2016.
- [3] F. Aznoli and N. J. Navimipour, "Cloud services recommendation: Reviewing the recent advances and suggesting the future research directions," *Journal of Network and Computer Applications*, vol. 77, pp. 73–86, 2017.
- [4] S. Rendle, C. Freudenthaler, Z. Gantner et al., "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25 Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, AUAI Press, 2009.
- [5] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: a survey," *Procedia Computer Science*, vol. 49, pp. 136–146, 2015.
- [6] X. Zhou, J. He, G. Huang, and Y. Zhang, "SVD-based incremental approaches for recommender systems," *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 717–733, 2015.
- [7] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geoFM: A ranking based geographical factorization method for point of interest recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2015*, pp. 433–442, August 2015.
- [8] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank citation ranking: bringing order to the web," Tech. Rep. 1999-66, Stanford InfoLab, 1999.
- [9] S. Berkovsky and J. Freyne, "Web personalization and recommender systems," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2015*, pp. 2307-2308, August 2015.
- [10] L. Lü, M. Medo, C. H. Yeung, Y. Zhang, Z. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [11] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.
- [12] M. Nilashi, O. B. Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3879–3900, 2014.
- [13] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for web services via collaborative filtering," in *Proceedings of the 5th IEEE International Conference on Web Services (ICWS '07)*, pp. 439–446, IEEE, July 2007.
- [14] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: a collaborative filtering based web service recommender system," in *Proceedings of the 7th IEEE International Conference on Web Services (ICWS '09)*, pp. 437–444, July 2009.
- [15] X.-W. Meng, X. Hu, L.-C. Wang, and Y.-J. Zhang, "Mobile recommender systems and their applications," *Ruan Jian Xue Bao/Journal of Software*, vol. 24, no. 1, pp. 91–108, 2013.
- [16] D. Bernardes, M. Diaby, R. Fournier, F. Fogelman-Soulié, and E. Viennet, "A social formalism and survey for recommender systems," *ACM SIGKDD Explorations Newsletter*, vol. 16, no. 2, pp. 20–37, 2015.
- [17] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.
- [18] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Computing*, pp. 1–18, 2017.
- [19] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.
- [20] H. Rong, T. Ma, M. Tang, and J. Cao, "A novel subgraph K+-isomorphism method in social network based on graph similarity detection," *Soft Computing*, pp. 1–19, 2017.
- [21] S. Deng, L. Huang, Y. Yin, and W. Tang, "Trust-based service recommendation in social network," *Applied Mathematics & Information Sciences*, vol. 9, no. 3, pp. 1567–1574, 2015.
- [22] S. Pongnumkul and K. Motohashi, "Random walk-based recommendation with restart using social information and bayesian transition matrices," *International Journal of Computer Applications*, vol. 114, no. 9, pp. 32–38, 2015.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

