2011 IEEE International Conference on Web Services

An Effective Web Service Recommendation Method based on Personalized Collaborative Filtering

Yechun Jiang, Jianxun Liu, Mingdong Tang
Department of Computer Science and Engineering,
Hunan University of Science and Technology
Xiangtan, 411201, China
{yechunjiang, ljx529}@gmail.com
tangmingdong@ict.ac.cn

Xiaoqing (Frank) Liu
Department of Computer Science
Missouri University of Science & Technology
Rolla, MO 65401, USA
fliu@mst.edu

Abstract—**Collaborative filtering is one of widely used Web service recommendation techniques. There have been several methods of Web service selection and recommendation based on collaborative filtering, but seldom have they considered personalized influence of users and services. In this paper, we present an effective personalized collaborative filtering method for Web service recommendation. A key component of Web service recommendation techniques is computation of similarity measurement of Web services. Different from the Pearson Correlation Coefficient (PCC) similarity measurement, we take into account the personalized influence of services when computing similarity measurement between users and personalized influence of services. Based on the similarity measurement model of Web services, we develop an effective Personalized Hybrid Collaborative Filtering (PHCF) technique by integrating personalized user-based algorithm and personalized item-based algorithm. We conduct series of experiments based on real Web service QoS dataset WSRec [11] which contains more than 1.5 millions test results of 150 service users in different countries on 100 publicly available Web services located all over the world. Experimental results show that the method improves accuracy of recommendation of Web services significantly.**

Keywords-Web service recommendation; Web service; Web service similarity measurement; personalized collaborative filtering; personalization

I. INTRODUCTION

Recently, recommendation systems are attracting a lot of attention since it helps users to deal with information overloading on the Web, Recommendation algorithms have been used to recommend books and CDs at Amazon.com, movies at Netflix.com, and news at VERSIFI Technologies (formerly AdaptiveInfo.com) [16].

Web service is a software system designed to support interoperable machine-to-machine interaction over a network [1]. The increasing number of Web services call for effective methods for their selection and recommendation, which is one of key problems in the field of service computing [2].

User preferences and Web service properties should be considered in service selection, especially the non-functional service properties, also known as Quality of Service (QoS). QoS is a set of properties including response time, price, reputation, correctness, etc. Many researchers propose that QoS should be a key factor in the success of building critical service-oriented applications [3, 4].

Collaborative Filtering (CF) [5] is one of popular recommendation algorithms. Breese et al. [6] introduced a classification of CF algorithms that divides them into two broad categories: memory-based algorithms and model-based algorithms. The memory-based collaborative filtering method has two kinds of approaches: user-based approaches [7] and item-based approaches [8, 9]. The user-based approach, which recommends to an active user items collected by other users sharing similar tastes; and item-based, which recommends to an active user those items similar to the ones the active user preferred in the past.

A few works have been done to apply CF to Web service recommendation. Shao et al. [10] proposed a user-based CF algorithm to predict QoS values. Zheng, Ma, Lyu and King [11] proposed a hybrid user-based and item-based CF algorithm to recommend Web services. Two research groups [12, 13] apply the CF, and they use MovieLens [14] data for experimental analysis. However, since neither of these algorithms is based on personalized CF algorithm and they did not take personal preference into account, their accuracy is not very high.

In this paper, we proposed a personalized hybrid collaborative filtering method by considering the personalization of service items and the personalization of service users. The user-based algorithm using PCC, called as personalized UPCC considers personalization of services and it performs better than the traditional UPCC. The item-based algorithm using PCC, called as personalized IPCC, performs better than the traditional IPCC, as shown later. We integrate the personalized UPCC and the personalized IPCC to develop a personalized hybrid collaborative filtering technique. Experiments in Section 5 show its significant improvement of recommendation accuracy over existing methods.

The rest of this paper is organized as follows. Section II introduces the background and related works. Section III proposes the QoS-based Web service recommendation overview by using personalized hybrid collaborative filtering algorithm. Section IV presents a personalized hybrid collaborative filtering algorithm. The experiments and results are discussed in Section V. Finally, Section VI concludes the paper.

978-0-7695-4463-2/11 $26.00 © 2011 IEEE
DOI 10.1109/ICWS.2011.38

II. Background And Related Work

In this section, we first will briefly introduce the technique of collaborative filtering. We will then discuss specifically the user-based collaborative filtering and the item-based collaborative filtering. Finally, we present the personalized recommendation.

A. Collaborative Filtering

Collaborative filtering, firstly proposed by Rich [15], is one of solid recommendation approaches. It has been widely used in commercial recommendation systems. The basic idea of CF is to characterize relationships between consumers and products based on their previous interactions [18]. Formally, a CF domain consists of a set of n users $\{u_1, u_2,\ldots,u_n\}$, a set of m items $\{i_1,i_2,\ldots,i_m\}$, and users' ratings on items, which is often denoted by a user-item matrix as depicted in Table I. Entry $r_{x,y}(1 \leqq x \leqq n,\ 1 \leqq y \leqq m)$ in this matrix represents user x's rating on item y. The 0 means the user does not rate the corresponding item.

TABLE I. User-Item Matrix

	i_1	i_2	…	i_m
u_1	0	1	…	0
u_2	3	0	…	0
…	…	…	…	…
u_n	3	4	…	1

B. User-based Collaborative Filtering

In general, the user-based collaborative filtering algorithm attempts to discover a group of users who are similar to a target user. We can achieve this objective by processing the above user-item matrix. By computing their similarity measurement based on a similarity measurement model, we can derive a user similarity matrix as is shown in Table II. The user similarity measurement is in the interval of [-1, 1], with a larger value indicating that the two users are more similar. A rating value is usually predicted for all missing items in the target user's profile.

TABLE II. User Similarity Matrix

	u_1	u_2	…	u_n
u_1	1	0.85	…	0.36
u_2	0.14	1	…	-0.87
…	…	…	…	…
u_n	0.58	0.67	…	1

C. Item-based Collaborative Filtering

The item-based algorithm differs from the user-based algorithm only in that it computes item similarity measurements instead of user similarities. The item similarity matrix is shown in Table III. A rating value is also predicted for all missing items in the target user's profile.

TABLE III. Item Similarity Matrix

	i_1	i_2	…	i_m
i_1	1	0.85	…	0.36
i_2	0.14	1	…	-0.87
…	…	…	…	…
i_m	0.58	0.67	…	1

D. Personalized Recommendation

Personalize recommendation has been studied in several recommendation systems. Liu et al. [16] studied effects of degree of correlation between users and objects on collaborative filtering. They found out that objects with different correlation degrees have different contributions to the similarity measurements and recommendation accuracy could be remarkably improved by incorporating degrees of correlation between users and objects.

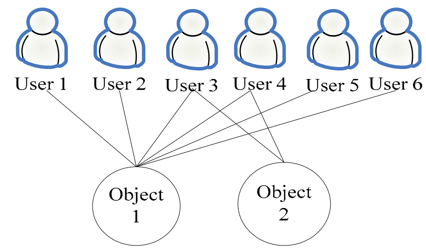


Figure 1. Illustration of user-object relation

We can see from Fig. 1 that if User 3 and User 4 both have selected Object 1 and Object 2, that means they have a similar taste for the Object 1 and Object 2. Provided that Object 1 is very popular (the degree of Object 1 is very large), the taste (favor for Object 1) is very ordinary and it does not mean that User 3 and User 4 are very similar. Therefore, the contribution of Object 1 to the similarity of User 3 and User 4 should be small. On the other hand, provided that Object 2 is very unpopular (the degree of Object 2 is very small), this taste is special. Therefore the contribution of the Object 2 to the similarity between User 3 and User 4 should be large. In other words, it is not very meaningful if two users both select a popular object, while if a very unpopular object is simultaneously selected by two users, there must be some common tastes shared by these two users.

Also, personalized recommendation exists in social network. For example, if People 1 and People 2 both have Friend 1 and Friend 2, they are similar because of the common friends they have. Provided that Friend 1 is very popular (a star or a public person), it does not mean that People 1 and People 2 are very similar, the contribution of Friend 1 to the similarity of People 1 and People 2 should be small. On the contrary, provided that People 2 is an ordinary person (a classmate or a roommate), so this friend is a special one, the contribution of Friend 2 to the similarity of People 1 and People 2 should be large. Therefore, personalization should play an important role in computing similarity measurements in a recommendation system.

III. QoS-based Web Service Recommendation Overview

Historical QoS record of Web services is an important factor in Web service recommendation. Web service QoS information helps making accurate Web service recommendation. Fig. 2 shows an approach overview of the personalized hybrid Web services recommendation. In this approach overview, historical QoS record of active users will be built. An Input Handler deals with the input data of the active user. The QoS record is used to analyze personalized service items and users; Users' similarity measurement will be computed based on the historical QoS record, the input data, and personalized services from other users, and it can be used to find similar users. With the historical QoS record, the input data, and the personalization of users, we can find similar services effectively. In the case of missing data, missing QoS values will be predicted from similar users or similar services. The Recommender weights the two predicted values to recommend optimal Web services to the active user.

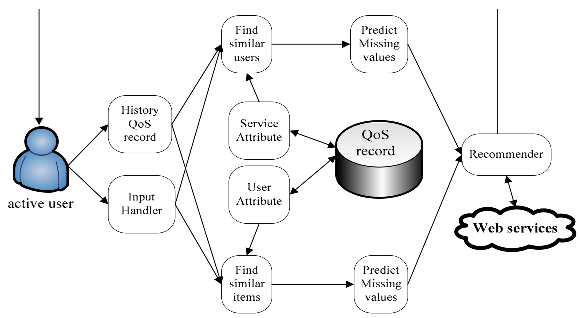


Figure 2. Web services recommendation overview

IV. Personalized Hybrid Collaborative Filtering Approach For Web Service Recommendation

A. Problem Definition

The objective of our method is to recommend Web services with optimal QoS values to an active user based on the personalization of users and services.

Formally, a Web service recommendation system is made up of M service users $\{u_1,u_2,...,u_m\}$ and N Web services items $\{i_1,i_2,...,i_n\}$. The relationship between services users and Web service items can be denoted by a user-item matrix $M \times N$. Every entry in this matrix $r_{m,n}$ represents a vector of QoS values (e.g., response-time, failure-rate, etc.), which are obtained by the service user m on the Web service item n. If user m did never invoke the Web service item n, $r_{m,n}$ =0.

B. Personalized Similarity Measure

How to determine the similarity between users? Pearson Correlation Coefficient (PCC) is used in many recommendation systems to compute the similarity. User-based collaborative filtering adopts PCC to compute similarity between two users a and u. Its computational formula is as Formula (1):

$$sim(a,u) = \frac{\sum_{i \in I}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i \in I}(r_{a,i}-\bar{r}_a)^2}\sqrt{\sum_{i \in I}(r_{u,i}-\bar{r}_u)^2}} \quad (1)$$

where $sim(a,u)$ denotes degree of similarity between user a and user u, I is the set of Web service items that are invoked by user a and user u. $r_{a,i}$ and $r_{u,i}$ denotes QoS values which were produced when user a and user u invoke service item i respectively. $\bar{r}_a$ and $\bar{r}_u$ represent an average QoS value of user a and user u respectively. It can be seen from the Formula (1) that $sim(a,u)$ is in the interval of [-1, 1]. The larger a value is, the more similar two users are.

As similar as the User-based collaborative filtering, Item-based collaborative filtering adopts PCC to compute similarity between Web service item i and j. The computational formula is as Formula (2):

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i}-\bar{r}_i)^2}\sqrt{(r_{u,j}-\bar{r}_j)^2}} \quad (2)$$

where $sim(i,j)$ denotes degree of similarity between service item i and j, U is a set of users that invoke service item i and j, $r_{u,i}$ and $r_{u,j}$ denote the QoS values which were produced when user u invokes service item i and j respectively. $\bar{r}_i$ and $\bar{r}_j$ represent an average QoS value of service item i and j respectively.

Not only because PCC sometimes overestimates the similarities of service users who are actually not similar but have similar QoS experience on a few co-invoked Web service items [15], but also the traditional user-based collaborative filtering does not take into account influence of personalization of service items. Similar to degree of influence [16] in recommendation system, degree of similarity between two users should be measured relative to personalization of service items that the two users co-invoked. However, when degree of similarity of two users is computed using Formula (1), different personalization of service items have the same contribution to the degree of similarity, as shown in Fig. 3.

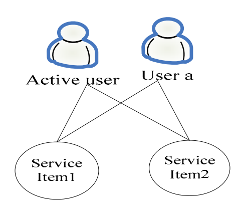


Figure 3. User-item relation

In Fig. 3, we suppose that the Active user and User a both invoked Service Item 1 and Service Item 2 before. It indicates they have similar taste for the Service Item 1 and Service Item 2. Provided that Service Item 1 is very common, and the standard deviation of QoS metrics of Service Item 1 is very small, which indicates that Service Item 1 is almost the same to all users. However, it does not mean the Active User and User a are very similar. Therefore the contribution of Service Item 1 to the degree of similarity of the Active User and User *a* should be small. On the other hand, provided that Service Item 2 is a user-sensitive service, and the standard deviation of QoS metrics of Service Item 2 is very large, which indicates that the QoS values are different from user to user. If two users have similar QoS values, these two users are really similar and therefore the contribution of the Service Item 2 to their degree of similarity should be large. To put it in another way, when we compute the similarity between two users, we shouldn't ignore the personalization of service items. Therefore, we develop a model for computing personalized degree of similarity as Formula (3):

$$Sim'(a,u)=\frac{\sum_{i\in I}\frac{|I|di^{\lambda_1}}{\sum_{i\in I}di^{\lambda_1}}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2}\sqrt{\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2}} \qquad (3)$$

where $I=I_a\cap I_u$, $|I|$ is the number of Web service items that are invoked by both of the two users, and di denotes the standard deviation of service item i. The parameter λ_1 is a tunable parameter, which determines degree of the personalization of service items. If $\lambda_1=0$, it does not take the personalization of service items into consideration and it is the same as the traditional user degree of similarity. Due to different dataset may have different personalization degree of service items, the parameter λ_1 makes the personalized method more adaptable to different environments.

Similar to the personalized user degree of similarity, suppose that the Service Item i and Service Item j both are invoked by Service User 1 and Service User 2 before. It indicates that they are similar because of the common Service Users that invoked them. Provided that Service User 1 is very common and the standard deviation of Service user 1 of QoS values is very small, which indicates that Service User 1 is almost the same to all Service Items. However it does not mean that the Service Item i and Service Item j are very similar. Therefore the contribution of Service User 1 to the similarity of the Service Item i and Service Item j should be small. On the contrary, provided that Service User 2 is a item-sensitive service and the standard deviation of Service User 2 is very large, which indicates that the QoS values are different from item to item, If two items have similar QoS values, which indicates that the two items are really similar, the contribution of the Service User 2 to their degree of similarity should be large. To put it in another way, when we compute degree of similarity between two service items, we shouldn't ignore the personalization of service users. Therefore, we develop a model of the personalized degree of similarity between item i and item j as Formula (4):

$$Sim'(i,j)=\frac{\sum_{u\in U}\frac{|U|du^{\lambda_2}}{\sum_{u\in U}du^{\lambda_2}}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_{u\in U}(r_{u,j}-\bar{r}_j)^2}} \qquad (4)$$

where $U=U_i\cap U_j$, $|U|$ is number of service users that invoked both service item i and service item j ,and du denotes standard deviation of service user u. The parameter λ_2 is a tunable parameter, which determines the degree of the personalization of service users. If $\lambda_2=0$, it does not take the personalization of service users into consideration and is the same as the traditional service item degree of similarity. Since different dataset may have different personalization degree of service users, the parameter λ_2 is used to make the personalized method more adaptable to different environments too.

C. Similar Neighbor Selection and Missing Value prediction

After computation of the user's degree of similarity and service item's degree of similarity, two matrices are obtained: one is a user similarity matrix as shown in Table II, and another is an item similarity matrix as shown in Table III. Two sets of neighbors can be defined. Since *Top-K* algorithm is widely used in recommendation systems and it has a high accuracy; we adopt the *Top-K* algorithm to define a set of similar service users $S(u)$ and a set of similar service items $S(i)$.

The user-based method uses $S(u)$ to predict missing QoS values for the active user by Formula (5):

$$P_u(r_{u,i})=\bar{u}+\frac{\sum_{u_a\in S(u)}Sim'(u_a,u)(r_{u_a,i}-\bar{u}_a)}{\sum_{u_a\in S(u)}Sim'(u_a,u)} \qquad (5)$$

where $\bar{u}$ denotes average QoS values of the active user, and $\bar{u}_a$ denotes the average QoS values of user *a*.

Similarly, the item-based method uses $S(i)$ to predict the missing QoS values for the active user by Formula (6):

$$P_i(r_{u,i})=\bar{i}+\frac{\sum_{ik\in S(i)}Sim'(i_k,i)(r_{i_k,i}-\bar{i}_k)}{\sum_{ik\in S(i)}Sim'(i_k,i)} \qquad (6)$$

where $\bar{i}$ denotes average QoS values of service item i, and i_k denotes the average QoS values of service item i_k.

Since these two predicted values may have different prediction performance, we use a tunable parameter μ to balance these two predicted values. We combine the two methods by using Formula (7):

$$P(r_{u,i})=\mu P_u(r_{u,i})+(1-\mu)P_i(r_{u,i}) \qquad (7)$$

By the above mechanism, the parameter μ determines how much the hybrid method relies on the personalized UPCC prediction or the personalized IPCC prediction. The parameter μ makes the method feasible to different environment.

D. *Web Service Recommendation*

After predicted values of the QoS are computed, we use them for Web service recommendation. When an active user requests a Web service with specific description, the one with the optimal predicted QoS value will be recommended to him/her.

V. EXPERIMENTS

Since a key component of Web service recommendation is the QoS prediction, we use the QoS prediction accuracy to measure the recommendation quality. We conduct a set of experiments to evaluate and validate our new method. Particularly, we address the following questions:

1) How does the personalized parameters λ_1 and λ_2 affect performance of prediction? The parameter λ_1 determines the degree of the personalization of service items, and the parameter λ_2 determines the degree of the personalization of service users. Experiments are carried out to examine the influence of λ_1 and λ_2.

2) How does the parameter μ influence the prediction accuracy? The parameter μ determines how much the hybrid method relies on the personalized UPCC prediction or the personalized IPCC prediction which have direct influence on the prediction accuracy. Experiments are carried out to check the impact of μ.

3) How does our approach compare with other CF-based Web service recommendation methods? We compare our approach with other existing works [6, 18, 11].

A. *Dataset*

We adopt a dataset provided by ZiBin Zheng [17] for our experiments. The dataset contains 150 files, where each file includes 10,000 Web service invocations on 100 Web services by a service user. There are totally more than 1.5 millions Web service invocations. Each line in the file is a Web service invocation result, where the Table IV provides several samples of the results. In this table, the ClientIP denotes different users, WSID represents different service items. The other four items are QoS attributes.

TABLE IV. EXAMPLES OF WEB SERVICE INVOCATIONS

ClientIP	WSID	Responsetime (ms)	DataSize	HTTP Code	HTTP Message
35.9.27.26	8451	2736	582	200	OK
35.9.27.26	8460	804	14419	200	OK
35.9.27.26	8953	20176	2624	-1	Timeout

B. *Data Processing*

Since QoS prediction is computed for a single attribute, we can not use the dataset shown in Table IV directly for our experiments. In this paper, we only take the prediction of RTT and Failure-rate as examples to conduct the experiments. The data processing procedure is: 1) Extract the Response time from data set as the RTT values and extract the HTTP Message in order to calculate Failure-rate; 2) When a user invoked a service item more than 100 times, compute an average value as an entry of the RTT matrix; 3) If HTTP Message is successfully invoked, we compute the number of failures and compute the failure rate as the entry of the failure-rate matrix. After data processing, we obtain two 150×100 user-item matrices. Table V shows the RTT matrix, Table VI shows the Failure-rate matrix.

TABLE V. RTT MATRIX AFTER PRE-PROCESSING

	s_1	s_2	…	s_{100}
u_1	71	4997	…	4300
u_2	545	343	…	53
…	…	…	…	…
u_{150}	65	448	…	33

TABLE VI. FAILURE-RATE MATRIX AFTER PRE-PROCESSING

	s_1	s_2	…	s_{100}
u_1	0	0	…	0
u_2	0	0	…	0
…	…	…	…	…
u_{150}	0	0.0638	…	0

C. *Evaluation Metric*

Mean Absolute Error (MAE) is often used in collaborative filtering methods to measure the prediction quality. MAE is defined as Formula (8):

$$MAE = \frac{\sum_{i,j}\left|r_{i,j} - \hat{r}_{i,j}\right|}{N} \tag{8}$$

where $r_{i,j}$ denotes actual QoS values of Web service j observed by service user i, and $\hat{r}_{i,j}$ represents the predicted QoS values, and N denotes the number of predicted values. Because different QoS properties of Web services have distinct value ranges, as same as [6], we use the Normalized Mean Absolute Error (NMAE) metric to measure the prediction quality of our personalize hybrid collaborative filtering method. NMAE is defined as Formula (9):

$$NMAE = \frac{MAE}{\sum_{i,j} r_{i,j} / N} \tag{9}$$

And smaller NMAE values represent higher prediction accuracy.

D. *Performance Evaluation*

We compare our method with other well-known prediction methods: user-based algorithm using PCC (UPCC) [6] , item-based algorithm using PCC (IPCC) [18], and hybrid algorithm WSRec [11] to evaluate its prediction performance. Formula (1) and Formula (2) are used for computing the UPCC and IPCC respectively.

The 150 service users are divided into two groups: one as training users and the rest as active users. The RTT matrix is divided into the RTT-training-matrix and the RTT-test-matrix, and so is the failure rate matrix. In order to simulate the real world situation, we randomly remove some entries of the training matrices to develop a set of sparse matrices with density ranging from 10% to 30%. For the test users, number of RTT values given by them varies from 10, 20 to 30, denoted as *Given 10*, *Given 20*, and *Given 30*, respectively. The removed entries of the test matrices are used to estimate the prediction quality. We set *TopK*=10, $\lambda_1 = 0.8$, $\lambda_2 = 0.8$, $\mu = 0.2$. Each experiment is performed 100 times and their average values are computed.

The NMAE performance comparison of different methods with the 10% and 30% density training matrices are shown in Table VII and Table VIII respectively. In both tables, UPCC denotes the traditional user-base algorithm using PCC, PU denotes the personalized UPCC by considering the effect of personalization of the service items. IPCC denotes the traditional item-based algorithm using PCC, PI denotes the personalized IPCC by considering the personalization of service users. WSRec [11] is a hybrid algorithm without taking the personalization of service items and the service users into consideration. The PHCF denotes our personalized hybrid collaborative filtering algorithm. It can be seen from the two tables that the PU has smaller NMAE values, which indicates better prediction performance. That is to say by adopting the personalization of service items to compute the service user's degree of similarity, the prediction accuracy is improved. The PI is similar to the PU. It has smaller NMAE values than IPCC. The PHCF achieves smaller NMAE values than WSRec [11], which means better performance. It also can be concluded from these results that the prediction accuracy can be improved by collecting more training data and giving the active user more Web service QoS data. In both tables, the NMAE performance of the failure-rate is worse than the RTT. That is because the failure-rate training matrix contains a lot of zero values. Under all the different experimental settings, PHCF consistently beats other methods.

TABLE VII. NMAE PERFORMANCE COMPARISON (TRAINING MATRIX DENSITY=10%)

	Training Users=100						**Training Users =140**					
QoS	**RTT**			**Failure-rate**			**RTT**			**Failure-rate**		
Given	**10**	**20**	**30**	**10**	**20**	**30**	**10**	**20**	**30**	**10**	**20**	**30**
UPCC	0.961	0.905	0.884	1.667	1.415	1.213	0.907	0.815	0.745	1.652	1.412	1.200
PU	0.810	0.751	0.688	1.452	1.330	1.306	0.800	0.748	0.637	1.310	1.254	1.214
IPCC	0.534	0.501	0.486	0.950	0.915	0.887	0.415	0.381	0.354	0.748	0.714	0.702
PI	0.459	0.421	0.406	0.842	0.806	0.782	0.401	0.368	0.337	0.719	0.690	0.662
WSRec	0.463	0.434	0.412	0.710	0.684	0.653	0.365	0.341	0.320	0.578	0.562	0.531
PHCF	0.415	0.403	0.386	0.653	0.621	0.603	0.340	0.321	0.300	0.527	0.516	0.500

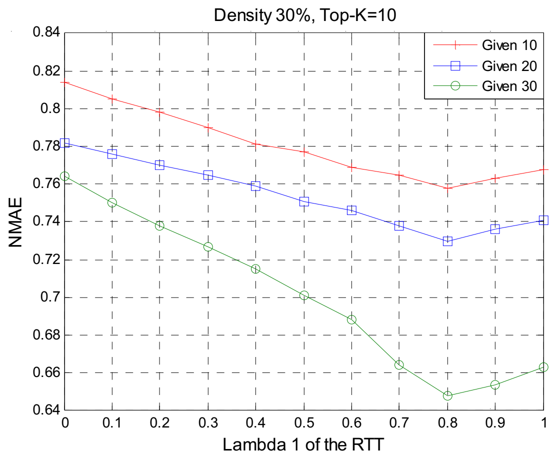
TABLE VIII. NMAE PERFORMANCE COMPARISON (TRAINING MATRIX DENSITY=30%)

	Training Users =100						**Training Users =140**					
QoS	**RTT**			**Failure-rate**			**RTT**			**Failure-rate**		
Given	**10**	**20**	**30**	**10**	**20**	**30**	**10**	**20**	**30**	**10**	**20**	**30**
UPCC	0.814	0.782	0.764	1.601	1.401	1.180	0.810	0.792	0.706	1.552	1.387	1.014
PU	0.758	0.730	0.648	1.306	1.259	1.156	0.753	0.731	0.625	1.157	1.011	0.988
IPCC	0.506	0.484	0.461	0.807	0.785	0.756	0.384	0.363	0.341	0.716	0.705	0.698
PI	0.426	0.403	0.389	0.756	0.731	0.700	0.350	0.335	0.321	0.554	0.514	0.506
WSRec	0.361	0.346	0.315	0.617	0.596	0.561	0.330	0.309	0.263	0.506	0.493	0.431
PHCF	0.330	0.314	0.296	0.548	0.516	0.500	0.302	0.274	0.238	0.496	0.452	0398

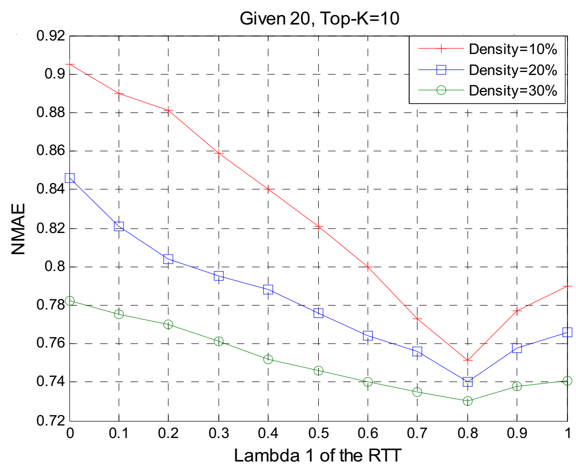
E. *Impact of λ_1 and λ_2*

The parameter λ_1 is a tunable parameter. It determines the degree of the personalization of service items. If $\lambda_1 = 0$, it does not take the personalization of service items into consideration and is the same as traditional user's degree of similarity. Different datasets may have different personalization degrees of service items. The parameter λ_1 makes the personalized method more adaptable to different environment. Likely, λ_2 determines the degree of the personalization of service users. If $\lambda_2 = 0$, it does not take the personalization of service users into consideration and is the same as the traditional service item's degree of similarity. Different dataset may have different personalization degree of service items. The parameter λ_2 makes the personalized method more adaptable to different environment.

To study the influence of λ_1 and λ_2, we set *Top-K*=10 and training users=140. We vary the value of λ_1 and λ_2 from 0 to 1 with a step value of 0.1 respectively. Fig. 4(a) shows the results of *Given 10*, *Given 20* and *Given 30* with Density=30% of RTT. The experimental results of failure-rate follow the same trend of RTT. Fig. 4(b) shows the results of Density 10%, Density 20% and Density 30% with *Given = 20* of RTT. The experimental results of failure-rate also follow the same trend of RTT. The impact of λ_2 is similar to that of λ_1, as shown in Fig. 5.

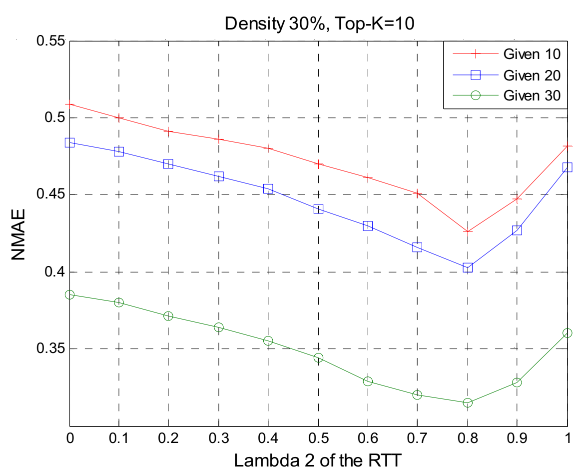


(a)

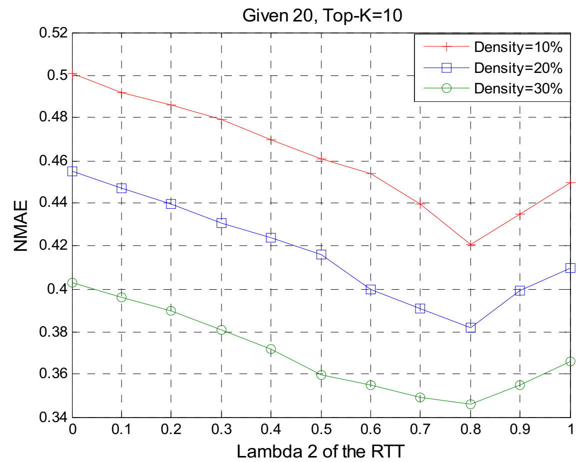


(b)

Figure 4. Impact of λ_1



(a)



(b)

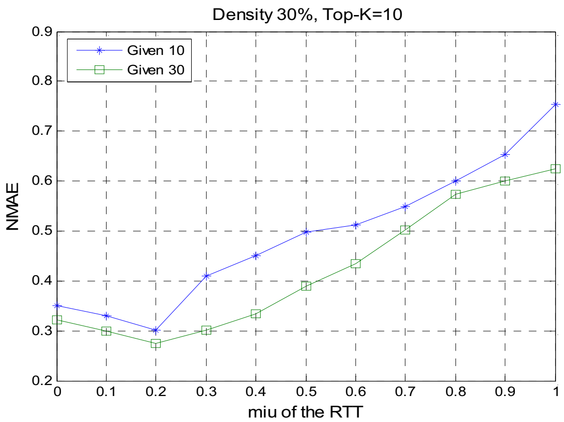
Figure 5. Impact of λ_2

It can be concluded from Fig. 4 and Fig. 5 that the value of λ_1 and λ_2 affect the accuracy significantly. Suitable λ_1 and λ_2 values provide better prediction performance. We obtain the best prediction performance at $\lambda_1 = 0.8$ and $\lambda_2 = 0.8$. It can also seen that the optimal values of λ_1 and λ_2 are not affected by the training matrix density and the *Given* numbers.

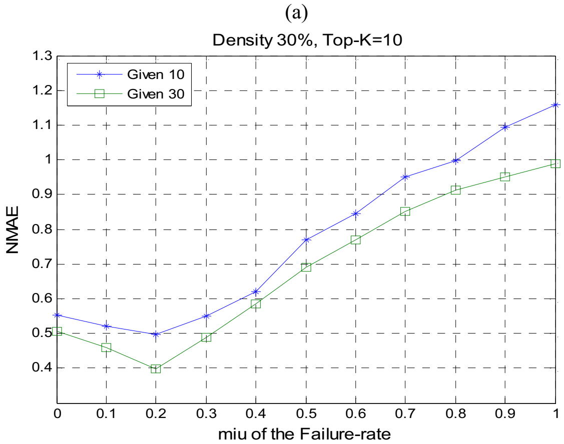
F. *Impact of* μ

Different datasets may have different data correlation characteristics. The parameter μ determines how much the hybrid method relies on the personalized UPCC prediction or the personalized IPCC prediction. The parameter μ makes the method feasible to different environment. If $\mu = 0$, we use only the personalized IPCC method. If $\mu = 1$, we use only the personalized UPCC method. In other cases, we combine the personalized UPCC and the personalized IPCC methods based on the value of μ to predict the missing value for active users.

To study the influence of μ on our personalized hybrid collaborative filtering method, we set *Top-K*=10 and *training users*=140. The value of μ varies from 0 to 1 with a step value of 0.1. Fig. 6(a) shows the results of *Given 10* and *Given 30* with Density 30% of RTT. Fig. 6(b) shows the results of *Given 10* and *Given 30* with Density 30% of Failure-rate. The experimental results of failure-rate follow the same trend of RTT. It can be concluded from Fig. 5 that the value of μ affects the recommendation results significantly, and suitable μ values, which represents proper combinations of the personalized UPCC method and the personalized IPCC method, provide better prediction performance. We achieve the best prediction performance at $\mu = 0.2$.



(a)



(b)

Figure 6. Impact of μ

VI. CONCLUSION

In this paper, we present a personalized hybrid collaborative filtering method for Web service recommendation. We take into account the influence of personalization of Web service items when computing degree of similarity between users. Similarly, we take into account the influence of personalization of service users when computing degree of similarity between service items. A personalized hybrid collaborative filtering (PHCF) is then based on the computational similarity models. According to the experiment results, the personalized UPCC method outperforms the traditional UPCC method, the personalized IPCC method outperforms the traditional IPCC approach, and the personalized hybrid method, which is an integration of the personalized UPCC and the personalized IPCC, outperforms the state-of-the-art Web service recommendation method WSRec [11]. It has achieved a significant improvement of performance over existing methods.

ACKNOWLEDGMENT

The authors would like to thank Zibin Zheng for the dataset he provided. The work was supported by National Natural Science Foundation of China under grant No.90818004, Program for New Century Excellent Talents in University under grant No.NCET-10-0140, and Foundation of Hunan Educational Committee under grant No.09K085.

REFERENCES

[1] H. Haas, and A. Brown, “Web services glossary,” W3C Working Group Note 11, http://www.w3.org/TR/ws-gloss/.

[2] L.J. Zhang, J. Zhang, and H. Cai, Services computing, Tsinghua University Press, 2007.

[3] O. Moser, F. Rosenberg, and S. Dustdar, “Non-intrusive monitoring and service adaptation for WS-BPEL,” Proc. 17th International Conference on World Wide Web (WWW), 2008,pp. 815–824.

[4] M. P. Papazoglou and D. Georgakopoulos, “Service-Oriented computing,” Communications of the ACM, 2003, pp. 46(10):24–28.

[5] G. Adomavicius, A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” IEEE Transactions on Knowledge and Data Engineering, 2005, pp: 734 – 749.

[6] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” Proc. of the UAI, 1998, pp.43-52.

[7] R. Jin, J. Y. Chai, and L. Si, “An automatic weighting scheme for collaborative filtering” Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004.doi: 10.1145/1008992.1009051.

[8] V. Cardellini, E. Casalicchio, V. Grassi, and F. L. Presti, “Flow-based service selection for Web service composition supporting multiple QoS classes,” Proc. 7th International Conference on Web Services ICWS, 2007, pp. 743–750.

[9] R. M. Sreenath and M. P. Singh. Agent-based service selection. Journal on Web Semantics, 2003. pp. 261–279.

[10] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, “Personalized QoS prediction for Web services via collaborative filtering,” Proc. 5th International Conference on Web Services (ICWS 2007), 2007, pp. 439-446.

[11] Z.B. Zheng, H. Ma, M.R. Lyu, and I. King, “WSRec: a collaborative filtering based Web service recommendation system,” Proc. 7th International Conference on Web Services (ICWS 2009), 2009, pp. 437-444.

[12] R. M. Sreenath and M. P. Singh, “Agent-based service selection,” Journal of Web Semantics, 2003, pp. 261–279.

[13] W. Rong, K. Liu, and L. Liang, “Personalized Web service ranking via user group combining association rule,” Proc. 7th International Conference on Web Services (ICWS 2009), 2009, pp.445-452.

[14] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, “MovieLens unplugged: experiences with an occasionally connected recommender system,” Proc. of ACM 2003 International Conference on Intelligent User Interfaces (IUI'03), 2003, pp.263-266.

[15] E. Rich, “User modeling via stereotypes,” Cognitive Science, vol.3, No.4, 1979.

[16] R. Liu, C.X. Jia, T. Zhou, D, Sun, B.H. Wang, “Personal recommendation via modified collaborative filtering,” Physics and Society 388(4): 2009, pp.462-468, doi : 10.1016/j.physa.2008.10.010

[17] Z.B. Zheng, M.R. Lyu, "Collaborative Reliability Prediction for Service-Oriented Systems,” Proc. of the 32nd ACM/IEEE International Conference on Software Engineering, vol.1, 2010, pp. 35 – 44.doi: 10.1145/1806799.1806809.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: An open architecture for collaborative filtering of netnews,” Proc. of ACM 1994 Conference on Computer Supported Cooperative Work, 1994, pp.175-186, doi : 10.1145/192844.192905