Personalized Web Service Recommendation via Normal Recovery Collaborative Filtering

Huifeng Sun, Zibin Zheng, *Member, IEEE*,
Junliang Chen, and
Michael R. Lyu, *Fellow, IEEE*

Abstract—With the increasing amount of web services on the Internet, personalized web service selection and recommendation are becoming more and more important. In this paper, we present a new similarity measure for web service similarity computation and propose a novel collaborative filtering approach, called normal recovery collaborative filtering, for personalized web service recommendation. To evaluate the web service recommendation performance of our approach, we conduct large-scale real-world experiments, involving 5,825 real-world web services in 73 countries and 339 service users in 30 countries. To the best of our knowledge, our experiment is the largest scale experiment in the field of service computing, improving over the previous record by a factor of 100. The experimental results show that our approach achieves better accuracy than other competing approaches.

Index Terms—Service recommendation, collaborative filtering, recommender system, QoS

1 Introduction

Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Web service discovery that deals with functional properties has been extensively studied [25]. However, web service discovery cannot differentiate services with identical or similar functionalities. Facing large amount of functionally equivalent services, it is difficult and expensive for service users to identify the optimal one, since the user-received performance of web services is highly related to user locations. Designing effective approaches for personalized web service selection and recommendation is becoming more and more important in the field of service computing [30].

Quality-of-Service (QoS) is usually defined as a set of nonfunctional properties, such as round-trip time (RTT), price, failure-rate, and so on. QoS is usually considered when making service selection [17], [19]. The values of some web service QoS properties (e.g., RTT and failure-rate) are influenced by the communication links and usually differ from user to user. It is not practical to measure QoS information (e.g., RTT, failure-rate) of all service candidates for each user, since it is money-spending, resource-consuming, and time-consuming to conduct real-world web service invocations for evaluation purposes [31]. It will be very attractive if we can make personalized QoS value prediction for a service user using a small amount of available web service QoS values. Based on the predicted QoS values, personalized web service recommendation can be conducted for service users.

Personalized web service recommendation provides valuable information for service users. First, it assists users in making decision when selecting optimal service from a set of functionally equivalent web services. To obtain the optimal service, a user can select the top 1 recommended service immediately, or select the one from k best services after testing them one by one. Second, it helps discover good performing web services for the current user, and recommends potential users to service providers.

Extended from its previous conference version [27], this paper proposes a novel collaborative filtering (CF) approach for personalized web service recommendation that takes advantage of the small amount of available QoS information. By our CF methods, QoS values of web services for an active user can be automatically predicted by employing the QoS information from other similar users who have similar historical QoS experience on the web services to the active user.

There are several challenges when applying CF methods to service recommendation. First, the existing CF methods are usually designed for product recommendation (e.g., movie recommendation, book recommendation), and they do not take characteristics of web service QoS into consideration. Second, the scale of experiments of the existing web service recommendation approaches [5], [23], [31] is too small to verify the recommendation results. As far as we know, the largest scale experiment in the field of service computing only contains 150 users and 100 web services [31]. To attack these challenges, this paper proposes a normal recovery collaborative filtering approach and conducts large-scale experiments to advance the current state-of-the-art in service recommendation.

The remainder of this paper is organized as follows: Section 2 introduces related work. Section 3 presents our normal recovery collaborative filtering approach. Section 4 shows experiments and Section 5 concludes the paper.

2 Related Work

Collaborative filtering (CF) approaches are widely adopted for the recommender systems [20]. According to [3], collaborative filtering algorithms can be grouped into two classes: memory-based and model-based. Memory-based collaborative filtering approaches are usually classified into user-based approaches [3], [7], [9], item-based approach [6], [13], [22], and their combined approaches [15], [29]. Similarity measures have been discussed in several investigations [3], [10], [20], [22], [24]. In memory-based collaborative filtering approaches, the Pearson correlation coefficient (PCC) [20], [24] and the cosine-based approach (COS) [3], [22] are the two most popular approaches [1] to measure the similarity.

Several kinds of approaches have been proposed for the service recommendation, containing semantic-based approach [12], context-based approach [14], syntactic-based approach [2], and CF-based approach. For presenting the nonfunctional properties of the web services, the QoS models of web services have been discussed in a number of research investigations [4], [8], [16], [18], [21], [28]. In recent years, a number of tasks have been proposed in employing collaborative filtering techniques for the QoS-aware web service selection and recommendation [5], [11], [23], [25], [31], [32]. These approaches predict the QoS values for an active user based on QoS values from similar users or similar web services. Investigations in [11], [25] mention the idea of applying user-based collaborative filtering techniques for web service recommendation. However, it is not convincing enough to employ the MovieLens (a publicly available film rating data set) for conducting experiments, since film ratings are quite different from web service QoS

- *H. Sun is with the State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Haidian, Beijing 100876, China, and the Research Institute of Petroleum Exploration and Development, PetroChina, Haidian, Beijing, China. E-mail: huifengsun.zj@gmail.com.*
- *Z. Zheng and M.R. Lyu are with the Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, The Chinese University of Hong Kong Shenzhen Research Institute, and Computer Science and Engineering Department, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Email: {zbzheng, lyu}@cse.cuhk.edu.hk.*
- *J. Chen is with the State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Haidian, Beijing 100876, China. Email: chjl@bupt.edu.cn.*

Manuscript received 31 May 2011; revised 2 Jan. 2012; accepted 20 May 2012; published online 23 Nov. 2012.
For information on obtaining reprints of this article, please send e-mail to: tsc@computer.org, and reference IEEECS Log Number TSC-2011-05-0056.
Digital Object Identifier no. 10.1109/TSC.2012.31.

1939-1374/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

values. Shao et al. [23] proposed a user-based PCC method for the web service QoS value prediction and conduct experiments on 20 web services. Zheng et al. [31] combined the user-based and item-based approaches to achieve better prediction accuracy and used PCC for the similarity measure. However, as shown in our experiments, the performance of the PCC for the similarity measure is not good enough. Moreover, the experiments only include 150 users and 100 services, which is too small compared with real-world situations.

Compared with the previous work, we propose a new similarity measurement approach and a novel collaborative filtering approach, named NRCF, for the QoS-based web service recommendation. The contributions of this paper can be summarized as follows:

1. We design a new similarity measure for memory-based collaborative filtering, which takes characteristics of web service QoS into consideration and can achieve more accurate QoS value prediction results.
2. We propose a new collaborative filtering approach, which significantly improves the recommendation performance compared with the other well-known approaches.
3. We collect a large-scale web service QoS data set, which contains 339 users and 5,825 real-world web services. Based on this data set, comprehensive experiments are conducted to study the performance of our approach.

3 Normal Recovery Collaborative Filtering

In this section, we present our normal recovery collaborative filtering approach. Given a web service recommender system that contains M users and N items (web services), we obtain an $M \times N$ user-item matrix, in which entry $r_{m,n}$ denotes the QoS value (e.g., RTT values) of the web service n observed by user m. If the entry $r_{m,n}$ is empty, then $r_{m,n} = \phi$, denoting that the web service n has never been invoked by user m before.

3.1 Traditional Similarity Measures

There are two types of similarity measures, i.e., the functional similarity measure and the nonfunctional similarity measure. Input/output/operation names are usually employed to measure the functional similarity between two web services. In this paper, instead of the functional similarity, we focus on the nonfunctional similarity (QoS similarity).

In user-based collaborative filtering, the PCC can be employed to measure the similarity between two users u and v by

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

where $I = I_u \cap I_v$ is the set of items coinvoked by users u and v, $r_{u,i}$ is the QoS value of item i observed by user u, and $\bar{r}_u$ denotes the average value of user u on items in I. From the above equation, we can see that the values of the PCC are in the interval of -1 and 1.

The PCC can also be employed to measure the similarity between two items i and j by

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

where $U = U_i \cap U_j$ is the set of users who invoked both items i and j, $r_{u,i}$ is the QoS value of item i observed by user u, and $\bar{r}_i$ denotes the average value of item i observed by users in U.

In the cosine-based approach, the similarity between users or items can be measured by calculating the cosine of the angle between them:

	i_1	i_2	i_3	i_4	i_5
u_1	1	2	3	4	5
u_2	2	2	3	4	
u_3	3	2		4	
u_4	1	1	1	1	2
u_5	5	5	5	5	6

Fig. 1. Motivating example.

$$Sim(u,v) = \frac{\sum_{i\in I} r_{u,i} r_{v,i}}{\sqrt{\sum_{i\in I} r_{u,i}^2}\sqrt{\sum_{i\in I} r_{v,i}^2}}, \quad (3)$$

$$Sim(i,j) = \frac{\sum_{u\in U} r_{u,i} r_{u,j}}{\sqrt{\sum_{u\in U} r_{u,i}^2}\sqrt{\sum_{u\in U} r_{u,j}^2}}, \quad (4)$$

where $I = I_u \cap I_v$ is the set of items coinvoked by users $u$and v and $U = U_i \cap U_j$ is the set of users who invoked both items i and j. Equation (3) is for calculating the similarity between users, while (4) is for calculating the similarity between items. Similarity values calculated by (3) and (4) are in the interval of -1 to 1.

3.2 Normal Recovery Similarity Measure

As shown in Fig. 1, let us consider a simple user-item matrix that includes five users (u_1 to u_5) and five items (i_1 to i_5), where 1 and 6 are the lowest and the highest QoS values (e.g., RTT values in seconds) of the user-item matrix, respectively. When adopting PCC as shown in (1), we calculate the similarity between users and obtain the arithmetic expression as follows:

$$Sim(u_1,u_2) > Sim(u_3,u_2).$$

According to this similarity computation result, u_1 is more similar to u_2 than u_3 is. However, as shown in Fig. 1, u_1 is actually less similar to u_2 than u_3 is, because the values from $u2$ and $u3$ both range between 2 and 4, while the values from $u1$ range between 1 and 5. In this example, the similarity calculation result by the PCC is inconsistent with this fact. This contradiction is caused by the fact that the PCC does not properly handle the QoS style difference between users.

Let us consider another example. When adopting COS in (3) for computing the similarity between u_4 and u_5, we obtain

$$Sim(u_4,u_5) = 0.9701.$$

According to this computation result, we can infer that u_4 and u_5 are very similar. However, as shown in Fig. 1, u_4 and u_5 are almost opposite because the QoS values of u_4 are close to the lowest value of the user-item matrix (i.e., 1), while the QoS values of u_5 are close to the highest value in the user-item matrix (i.e., 6). In this example, the similarity calculation result is inconsistent with the fact, which is caused by the case that COS overlooks the length of vectors.

There are several shortcomings when applying the existing well-known similarity measurement approaches for the web service recommendation. The PCC does not properly handle the QoS style difference of vectors in different vector spaces, while COS only measures the angle between two vectors and neglects length difference of different vectors. To overcome these shortcomings, we propose a new similarity measure named normal recovery (NR). Our NR approach first normalizes the user QoS values to the same range, and then unifies similarity of the scaled user vectors (or item vectors) in different multidimensional vector spaces. The precondition of NR is that neither of the two vectors has the same value on all dimensions.

In our approach, to measure the similarity between users, we first normalize each row of the original user-item matrix P by the lowest and the highest QoS values of the same row, so that each row has a value range of [0, 1]. As a result, we map the original user-item matrix P into row-normal user-item matrix P_{nu}. Assume users u and v coinvoke num items, and vectors $\vec{u}$ and $\vec{v}$ are the observed value vectors of users u and v in the matrix P_{nu}, respectively. $dist(\vec{u}, \vec{v})$ calculates the euclidean distance between $\vec{u}$ and $\vec{v}$ in the num-dimensional vector space, while $dist_{max}$ calculates the maximal euclidean distance in the num-dimensional vector space where each dimension ranges from 0 to 1. Let $nr_{u,i}$ and $nr_{v,i}$ be the value of item i from users u and v in P_{nu}, respectively. Extended from our previous work [26], we propose the formula of NR to measure the similarity between two users u and v as follows:

$$Sim(u,v) = 1 - \frac{dist(\vec{u},\vec{v})}{dist_{max}} = 1 - \frac{\sqrt{\sum_{i\in I}(nr_{u,i} - nr_{v,i})^2}}{\sqrt{\sum_{k=1}^{|I|}(1-0)^2}}$$
$$= 1 - \frac{\sqrt{\sum_{i\in I}\left(\frac{r_{u,i}-r_{umin}}{r_{umax}-r_{umin}} - \frac{r_{v,i}-r_{vmin}}{r_{vmax}-r_{vmin}}\right)^2}}{\sqrt{\sum_{k=1}^{|I|}1}},$$

i.e.,

$$Sim(u,v) = 1 - \frac{\sqrt{\sum_{i\in I}\left(\frac{r_{u,i}-r_{umin}}{r_{umax}-r_{umin}} - \frac{r_{v,i}-r_{vmin}}{r_{vmax}-r_{vmin}}\right)^2}}{\sqrt{|I|}}, \tag{5}$$

where $I = I_u \cap I_v$ is the set of items coinvoked by users u and v; $|I|$ is the number of I; $r_{u,i}$ is the value of item i from user u in the original user-item matrix P; r_{umin} and r_{umax} denote the lowest and the highest values from user u in P, respectively; and r_{vmin} and r_{vmax} denote the lowest and the highest values from user v in P, respectively. In (5), $Sim(u,v) \in [0,1]$, where $Sim(u,v) = 0$ represents that two users are dissimilar and $Sim(u,v) = 1$ indicates that two users are the same. For the motivating example mentioned in Section 3.2, adopting (5) we obtain

$$Sim(u_1,u_2) < Sim(u_3,u_2),\ Sim(u_4,u_5) = 0,$$

which are consistent with the facts.

Specially, the situation when user u (or v) has the same value for all items make (5) fail to work. But this special situation exists in theory and does not exist in reality, because the users' QoS values (e.g., RTT values) of web services distributed all over the world cannot be all the same. Therefore, we ignore this special situation in reality.

To calculate the similarity between two web services, similarly, based on normalizing the items' QoS values, we map the original user-item matrix P into the column-normal user-item matrix P_{ni}, where each column is in the range of [0, 1]. The formula of NR to measure the similarity between two items i and j is as follows:

$$Sim(i,j) = 1 - \frac{\sqrt{\sum_{u\in U}\left(\frac{r_{u,i}-r_{imin}}{r_{imax}-r_{imin}} - \frac{r_{u,j}-r_{jmin}}{r_{jmax}-r_{jmin}}\right)^2}}{\sqrt{|U|}}, \tag{6}$$

where $U = U_i \cap U_j$ is the set of users who invoked both items i and j; $|U|$ is the number of U; $r_{u,i}$ is the value of item i from user u in the original matrix P; and r_{imin}, r_{imax}, r_{jmin}, and r_{jmax} denote the lowest value of item i, the highest value of item i, the lowest value of item j, and the highest value of item j in the original matrix P, respectively. In (6), $Sim(i,j) \in [0,1]$, where $Sim(i,j) = 0$ represents that two items are completely dissimilar, and $Sim(i,j) = 1$ represents that two items are the same.

Especially, the situation when item i (or j) has the same value for all the users makes (6) fail to work.But this special situation exists in theory and does not exist in reality, because the items' QoS values (e.g., RTT values) from users located all over the world cannot be all the same. Therefore, we ignore this special situation in reality.

During the similarity measure (e.g., using the PCC, COS, or NR), when a user has not invoked any web services (or a web service has not been invoked by any user), we do not calculate its similarities with other users (or web services).

3.3 Normal Recovery Collaborative Filtering

Based on our NR similarity measurement approach, we propose an innovative memory-based CF method, named normal recovery collaborative filtering (NRCF).

When predicting the unknown QoS value $\hat{r}_{u,i}$ of user u on item i, our NRCF recovers the original scale of user u or item i. In the user-based QoS value prediction, we define our user-based NRCF as follows:

$$\hat{r}_{u,i} = r_{umin} + (r_{umax} - r_{umin})\frac{\sum_{u'\in U} Sim(u,u') \times nr_{u',i}}{\sum_{u'\in U} Sim(u,u')}, \tag{7}$$

where U denotes the set of similar users to user u, who have invoked item i; $nr_{u',i}$ is the value of item i from user u' in the row-normal matrix P_{nu}; r_{umin} and r_{umax} are the lowest and the highest values from user u in the original matrix P, respectively; and $Sim(u,u')$ can be computed by (5).

In the item-based value prediction, we create item-based NRCF whose formula is given as

$$\hat{r}_{u,i} = r_{imin} + (r_{imax} - r_{imin})\frac{\sum_{i'\in I} Sim(i,i') \times nr_{u,i'}}{\sum_{i'\in I} Sim(i,i')}, \tag{8}$$

where I denotes the set of similar items to item i, which have been invoked by user u; $nr_{u,i'}$ is the value of item i' from user u in the column-normal matrix P_{ni}; r_{imin} and r_{imax} are the lowest and the highest values of item i in the original matrix P, respectively; and $Sim(i,i')$ can be computed by (6).

To make use of the information from both similar users and similar items, a parameter $\lambda(0 \le \lambda \le 1)$ is employed to determine how much does the prediction rely on user-based NRCF or item-based NRCF. Our NRCF approach makes prediction by employing the following equation:

$$\hat{r}_{u,i} = \lambda \times \left(r_{umin} + (r_{umax} - r_{umin})\frac{\sum_{u'\in U} Sim(u,u') \times nr_{u',i}}{\sum_{u'\in U} Sim(u,u')}\right)$$
$$+ (1-\lambda) \times \left(r_{imin} + (r_{imax} - r_{imin})\frac{\sum_{i'\in I} Sim(i,i') \times nr_{u,i'}}{\sum_{i'\in I} Sim(i,i')}\right). \tag{9}$$

When $\lambda = 1$, our approach only uses information of similar users to make prediction. When $\lambda = 0$, our approach uses information of similar web services for making missing value prediction. When $0 < \lambda < 1$, our approach systematically combines the user-based NRCF approach (see (7)) and the item-based NRCF approach (see (8)) to fully utilize the information of both similar users and similar web services.

3.4 NRCF for Web Service Recommendation

The predicted QoS values via our NRCF approach can be employed for the web service recommendation and selection. When the target web services are functionally equivalent, the one with the best predicted QoS performance can be recommended to the current user. By this way, personalized web service selection can be achieved without conducting the expensive and time-consuming real-world web service invocations. When the target web services have different functionalities, our NRCF approach can help make personalized QoS performance prediction of the

TABLE 1
MAE Performance Comparison of Different Similarity Measures (Smaller Value Means Better Performance)

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PCC	0.5714	0.5414	0.5146	0.4908	0.4702	0.4533	0.4404	0.4328	**0.4322**	0.4434
COS	0.6839	0.6499	0.6175	0.5867	0.5575	0.5303	0.5055	0.4839	0.4672	**0.4612**
NR	0.3484	**0.3473**	0.3495	0.3542	0.3610	0.3699	0.3807	0.3936	0.4088	0.4278

unused web services for the current users and recommend the good performing ones to the users.

In the web service recommender system, service users and web services are located in many countries that are far from each other. In addition, the network through which web services are invoked is highly dynamic. Thus, the user QoS styles and the item QoS styles are very different from each other. Our NRCF approach can adapt to different environment easily, since it considers the QoS style difference and makes use of information of both similar users and similar items for making the prediction.

4 Experiments

4.1 Web Service QoS Data Set

The experiments are conducted by employing our web service QoS data set [33], which contains 1,873,838 real-world route-trip time (RTT) records from 339 distributed users on 5,825 real-world web services. To collect the data set, we use 339 distributed planet-lab[1] computers from 30 countries to monitor the 5,825 web services in 73 countries. As far as we know, the scale of this data set is the largest in the field of service computing.

Given an $M \times N$ user-item matrix that contains T records on N items from M users, we define density of the matrix as $density = \frac{T}{M \times N}$. Based on this definition, the density of our user-item matrix is 94.9 percent.

4.2 Evaluation Metric

To evaluate the QoS value prediction accuracy, we use the well-known mean absolute error (MAE) metric. The MAE is the average absolute deviation of predictions to the ground truth values. The MAE is defined as

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{N}, \tag{10}$$

where $r_{u,i}$ denotes the actual RTT of web service i observed by user u, $\hat{r}_{u,i}$ denotes the predicted RTT of web service i for user u, and N denotes the total number of predicted RTTs. Smaller MAE values indicate better prediction accuracy.

4.3 Experimental Setup

To study the RTT value prediction accuracy of different prediction approaches, we randomly remove RTT records from the original user-item matrix to produce 10 sparse matrices. The densities of these sparse matrices range from 2 percent to 20 percent with a step of 2 percent. The reason why we use small density values is that a user usually only invokes a small number of web services in real-world. Therefore, the user-item matrix is typically very sparse.

We use five-fold cross validation in the experiments. For each one of the 10 sparse matrices, we separate its RTT records into two parts, i.e., training set (80 percent of the RTT records in the matrix) and test set (the remaining 20 percent of the RTT records in the matrix). The separating process is repeated five times, and for each time we predict RTTs in the test set based on the RTTs in the corresponding training set. The advantage of this method is that all RTT records are used for training and validation, and each RTT record is used for validation exactly once. For obtaining a reliable evaluation, we repeat the five-fold cross-validation process five times and report the average MAE of 25 folds (5 folds $\times$ 5 times).

1. http://www.planet-lab.org

4.4 Performance Comparison of Similarity Measures

To show the effectiveness of our NR similarity measure (see (5) and (6)), we compare it with two other well-known similarity measures, i.e., PCC (see (1) and (2)) and COS (see (3) and (4)), which have been introduced in Section 3.1. We combine NR, PCC, and COS with the formula shown in (9) for the missing RTT value prediction, respectively.

In this experiment, we set $density = 0.1$, neighborhood size $k = 30$, and vary the value of λ from 0.1 to 1 with a step value of 0.1. Table 1 shows the prediction accuracy of different similarity measures. From Table 1, we can see the following:

1. Our NR approach outperforms all the competing approaches consistently under different λ values.
2. The worst case of NR (0.4278) is superior to the best ones of the competing approaches (0.4322 for PCC and 0.4612 for COS).
3. The best MAE performances of PCC, COS, and NR are 0.4322, 0.4612, and 0.3473, respectively. Compared with PCC and COS, our NR approach significantly improves the prediction accuracy (24.45 percent and 32.80 percent better than PCC and COS, respectively).

4.5 Performance Comparison of Prediction Approaches

To show the effectiveness of our NRCF approach, we compare it with five well-known prediction approaches:

- *User-mean (UMEAN).* UMEAN predicts the missing RTT values by employing the average RTT value of other web services invoked by the same user.
- *Item-mean (IMEAN).* IMEAN predicts the missing RTT values by employing the average RTT value of the web service observed by other users.
- *User-based CF adopting PCC (UPCC).* UPCC employs the information of similar users for making missing value prediction [11], [23], [25]. Equation (1) is employed by UPCC when making a prediction.
- *Item-based CF adopting PCC (IPCC).* IPCC employs the information of similar web services for making prediction [13], [20]. Equation (2) is employed by IPCC when making a prediction.
- *WSRec.* WSRec is a memory-based CF approach, which systematically combines the UPCC and IPCC approaches [31].

In the six prediction approaches, the missing value predictions of UPCC, IPCC, WSRec, and NRCF are influenced by the neighborhood size. In this experiment, we vary the neighborhood size from 10 to 50 with a step value of 10. Since there are no parameters for the six approaches except NRCF and WSRec, we set $\lambda = 0.1$ for NRCF, $confidence\ weight = 0.1$ for WSRec. (In WSRec, a parameter called *confidence weight* is employed to determine how much the prediction relies on the user-based method or item-based method, and it reaches the best performance

TABLE 2
MAE Performance Comparison of Different Prediction Approaches (Smaller Value Means Better Performance)

(a) Number of Neighbors: 10

Density	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
UMEAN	0.8800	0.8727	0.8676	0.8781	0.8704	0.8764	0.8705	0.8802	0.8736	0.8729
IMEAN	0.6812	0.6829	0.6656	0.6796	0.6761	0.6772	0.6748	0.6750	0.6775	0.6740
UPCC	0.8400	0.8058	0.7728	0.7566	0.7242	0.7051	0.6790	0.6651	0.6412	0.6210
IPCC	0.6718	0.6869	0.6771	0.6880	0.6746	0.6680	0.6512	0.6452	0.6405	0.6214
WSRec	0.6148	0.5976	0.5667	0.5661	0.5473	0.5341	0.5199	0.5075	0.4971	0.4815
NRCF	**0.3714**	**0.3970**	**0.3814**	**0.3876**	**0.3755**	**0.3703**	**0.3713**	**0.3696**	**0.3613**	**0.3472**
	39.59%	33.57%	32.70%	31.53%	31.39%	30.67%	28.58%	27.17%	27.32%	27.89%

(b) Number of Neighbors: 20

Density	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
UMEAN	0.8919	0.8830	0.8761	0.8733	0.8748	0.8821	0.8713	0.8762	0.8736	0.8752
IMEAN	0.6977	0.6767	0.6803	0.6783	0.6779	0.6799	0.6756	0.6774	0.6729	0.6759
UPCC	0.8528	0.8158	0.7840	0.7568	0.7311	0.7128	0.6849	0.6679	0.6469	0.6294
IPCC	0.6839	0.6711	0.6781	0.6649	0.6507	0.6345	0.6108	0.6068	0.5926	0.5849
WSRec	0.6320	0.5929	0.5810	0.5662	0.5509	0.5399	0.5232	0.5120	0.4973	0.4871
NRCF	**0.3859**	**0.3852**	**0.3744**	**0.3666**	**0.3598**	**0.3615**	**0.3528**	**0.3477**	**0.3388**	**0.3325**
	38.94%	35.03%	35.56%	35.25%	34.69%	33.04%	32.57%	32.09%	31.87%	31.74%

(c) Number of Neighbors: 30

Density	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
UMEAN	0.8767	0.8707	0.8778	0.8710	0.8766	0.8754	0.8854	0.8770	0.8796	0.8759
IMEAN	0.6982	0.6821	0.6816	0.6741	0.6752	0.6780	0.6868	0.6782	0.6801	0.6769
UPCC	0.8389	0.8089	0.7896	0.7592	0.7391	0.7168	0.7034	0.6772	0.6609	0.6410
IPCC	0.6832	0.6719	0.6652	0.6423	0.6266	0.6106	0.5986	0.5797	0.5732	0.5617
WSRec	0.6348	0.5983	0.5838	0.5640	0.5502	0.5395	0.5341	0.5144	0.5037	0.4907
NRCF	**0.3774**	**0.3736**	**0.3693**	**0.3568**	**0.3491**	**0.3460**	**0.3502**	**0.3358**	**0.3316**	**0.3251**
	40.55%	37.56%	36.74%	36.74%	36.55%	35.87%	34.43%	34.72%	34.17%	33.77%

(d) Number of Neighbors: 40

Density	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
UMEAN	0.8831	0.8717	0.8801	0.8748	0.8717	0.8794	0.8792	0.8697	0.8760	0.8744
IMEAN	0.6865	0.6750	0.6874	0.6750	0.6744	0.6777	0.6768	0.6749	0.6767	0.6732
UPCC	0.8468	0.8107	0.7946	0.7650	0.7397	0.7244	0.7057	0.6784	0.6658	0.5342
IPCC	0.6686	0.6601	0.6598	0.6314	0.6143	0.5983	0.5854	0.5708	0.5587	0.4725
WSRec	0.6238	0.5931	0.5894	0.5661	0.5512	0.5410	0.5293	0.5141	0.5039	0.3974
NRCF	**0.3786**	**0.3638**	**0.3616**	**0.3473**	**0.3434**	**0.3428**	**0.3403**	**0.3336**	**0.3274**	**0.2870**
	39.31%	38.66%	38.65%	38.65%	37.70%	36.64%	35.71%	35.11%	35.03%	27.78%

(e) Number of Neighbors: 50

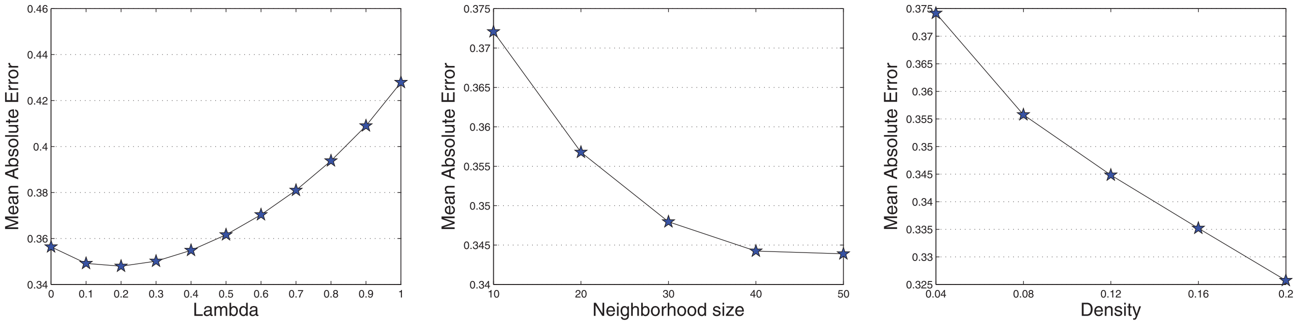
Density	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
UMEAN	0.8727	0.8886	0.8716	0.8808	0.8666	0.8718	0.8792	0.8760	0.8807	0.8700
IMEAN	0.6844	0.6903	0.6767	0.6792	0.6761	0.6717	0.6777	0.6746	0.6804	0.6716
UPCC	0.8376	0.8287	0.7885	0.7735	0.7406	0.7232	0.7099	0.6896	0.6770	0.5412
IPCC	0.6665	0.6718	0.6437	0.6293	0.6084	0.5859	0.5760	0.5626	0.5575	0.4536
WSRec	0.6241	0.6101	0.5825	0.5702	0.5554	0.5385	0.5306	0.5164	0.5098	0.3985
NRCF	**0.3677**	**0.3682**	**0.3504**	**0.3461**	**0.3411**	**0.3383**	**0.3399**	**0.3326**	**0.3301**	**0.2924**
	41.08%	39.65%	39.85%	39.30%	38.58%	37.18%	35.94%	35.59%	35.25%	26.62%

when *confidence weight* is 0.1 according to our experiments, in which we vary *confidence weight* from 0 to 1 with a step of 0.1.)

Table 2 shows the experimental results of all evaluated approaches in terms of prediction accuracy measured by MAE. In Table 2, there are five subtables corresponding to the neighborhood size of 10, 20, 30, 40, and 50. In each subtable, we present the MAE values of different matrix densities (i.e., we vary the user-item matrix density from 0.02 to 0.2 with a step of 0.02). The last row of each subtable shows the improvement of NRCF over the best result of all the competing approaches.

As shown in Table 2, we can see the following:

1. Our NRCF significantly outperforms all the competing approaches consistently, with an improvement of 27.17 percent to 41.08 percent better than the best results of other competing approaches.
2. With the increase of the user-item matrix density from 0.02 to 0.20, the improvement percentages of our NRCF decline in general. When the user-item matrix becomes sparser, there are more vectors with less covalued dimensions, which leads to NRCF's better improvement compared with the competing approaches.
3. For each subtable, the range between the lowest MAE of NRCF and the highest MAE of NRCF is small, i.e., NRCF keeps a smooth MAE performance under different user-item matrix densities, since NRCF considers dimension-number difference of vectors.



(a) Impact of λ (b) Impact of k (c) Impact of Matrix Density

Fig. 2. Impact of parameters.

These experimental results indicate two aspects of NRCF: On one hand, the more sparse the user-item matrix is, the better performance improvement NRCF achieves compared with other prediction approaches. On the other hand, NRCF is not sensitive to the increase of user-item matrix density, indicating that our NRCF approach can provide good prediction performance even with limited available QoS values. Since in real web service recommender systems, the user-item matrix is usually very sparse, our NRCF can provide more stable and accurate QoS value prediction services compared with other approaches.

4.6 Impact of λ

The parameter λ balances the information from similar users and similar items. If $\lambda = 0$, we only extract information from items for making the prediction, and if $\lambda = 1$, we only employ information of similar users. In the other cases, we fuse information from users and items to make the prediction. To study the impact of the parameter λ on the prediction results, in this experiment, we vary λ from 0.1 to 1 with a step of 0.1. We set $k = 30$ and $density = 0.1$. Fig. 2a shows the impact of λ on the prediction results.

As shown in Fig. 2a, the MAE value first declines slightly and then rises sharply. It reaches bottom when $\lambda = 0.2$. This experimental result indicates that by systematically fusing the information of similar users and similar web services (i.e., set $\lambda = 0.2$), the prediction accuracy can be enhanced.

4.7 Impact of Neighborhood Size k

The neighborhood size k plays an important role in our NRCF approach, which determines how many users or items are employed to predict missing values. To study the impact of parameter k, in this experiment, we vary the value of k from 10 to 50 with a step of 10. We set $\lambda = 0.2$ and $density = 0.1$. Fig. 2b shows the impact of neighborhood size k on the prediction results of our NRCF approach.

As shown in Fig. 2b, the curve declines with the increase of value k, indicating that our NCRF achieves the better performance when more neighbors are employed to make the prediction.

4.8 Impact of User-Item Matrix Density

The performance of NRCF is also influenced by the user-item matrix density. The user-item matrix density means how many records in the matrix can be employed to predict the missing values. To study the impact of the matrix density, in this experiment, we vary $density$ from 0.04 to 0.2 with a step of 0.04. We set $\lambda = 0.2$ and $k = 30$. Fig. 2c displays the impact of the user-item matrix density on the prediction accuracy of our NRCF approach.

As shown in Fig. 2c, the curve declines with the increase of $density$. This experimental result indicates that our NRCF achieves more accurate prediction accuracy with the increase of the user-item matrix density, since a denser matrix provides more information for making the missing value prediction.

5 Conclusion

In this paper, we propose a normal recovery collaborative filtering approach (named NRCF) to address the problem of personalized web service recommendation. Our NRCF approach investigates the characteristics of web service QoS values and proposes a new similarity measure, which finds similar users (or web services) more accurately and leads to better QoS value prediction accuracy. Moreover, by systematically fusing the information of similar users and similar web services, our NRCF approach can achieve better prediction accuracy. Comprehensive experiments are conducted on a large-scale real-world web service QoS data set. The experimental results show that our method significantly improves the QoS value prediction accuracy compared with other existing approaches.

The Internet environment is highly dynamic and the QoS value of web services may change over time. By taking advantages of the latest advanced technologies in machine learning, we will design an online version of our algorithm to effectively handle this dynamic QoS changing problem in our future work. Furthermore, we also plan to investigate additional QoS properties of web services and apply our NRCF approach to other application domains, such as service recommendation for the field of cloud computing in which everything can be seen as a service.

Acknowledgments

The authors thank the reviewers for their extensive and informative comments for the improvement of this manuscript. The work described in this paper was fully supported by the 973 Program of the National Basic Research Program of China (Grant No. 2011CB302704), the National Natural Science Foundation of China (Grant No. 61132001, 61001118), the Shenzhen Basic Research Program (Project Nos. JCYJ20120619153834216), and the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK 415311).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, pp. 734-749, June 2005.

[2] M.B. Blake and M.F. Nowlan, "A Web Service Recommender System Using Enhanced Syntactical Matching," *Proc. IEEE Fifth Int'l Conf. Web Services (ICWS '07)*, pp. 575-582, 2007.

[3] J. Breese et al., "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI '98)*, pp. 43-52, 1998.

[4] V. Cardellini, E. Casalicchio, V. Grassi, and F.L. Presti, "Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes," *Proc. IEEE Fifth Int'l Conf. Web Services (ICWS '07),* pp. 743-750, 2007.

[5] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-Aware Web Service Recommendation and Visualization," *IEEE Trans. Service Computing,* vol. 6, no. 1, pp. 35-47, Jan. 2012.

[6] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation," *ACM Trans. Information System,* vol. 22, no. 1, pp. 143-177, 2004.

[7] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," *Proc. 22nd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '99),* pp. 230-237, 1999.

[8] M.C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "QoS Aggregation for Web Service Composition Using Workflow Patterns," *Proc. IEEE Eighth Int'l Conf. Enterprise Distributed Object Computing,* pp. 149-159, 2004.

[9] R. Jin, J.Y. Chai, and L. Si, "An Automatic Weighting Scheme for Collaborative Filtering," *Proc. 27th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '04),* pp. 337-344, 2004.

[10] R. Jin and L. Si, "A Study of Methods for Normalizing User Ratings in Collaborative Filtering," *Proc. 27th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '04),* pp. 568-569, 2004.

[11] K. Karta, "An Investigation on Personalized Collaborative Filtering for Web Service Selection," honors program thesis, Univ. of Western Australia, Brisbane, 2005.

[12] M. Klusch and P. Kapahnke, "iSeM: Approximated Reasoning for Adaptive Hybrid Selection of Semantic Services," *Proc. IEEE Fourth Int'l Conf. Semantic Computing (ICSC '10),* pp. 184-191, 2010.

[13] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing,* vol. 7, no. 1, pp. 76-80, Jan. 2003.

[14] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, "Using Context Similarity for Service Recommendation," *Proc. IEEE Fourth Int'l Conf. Semantic Computing (ICSC '10),* pp. 277-284, 2010.

[15] H. Ma, I. King, and M.R. Lyu, "Effective Missing Data Prediction for Collaborative Filtering," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07),* pp. 39-46, 2007.

[16] D.A. Menasce, "QoS Issues in Web Services," *IEEE Internet Computing,* vol. 6, no. 6, pp. 72-75, Nov. 2002.

[17] O. Moser, F. Rosenberg, and S. Dustdar, "Non-Intrusive Monitoring and Service Adaptation for WS-BPEL," *Proc. 17th Int'l Conf. World Wide Web (WWW '08),* pp. 815-824, 2008.

[18] M. Ouzzani and A. Bouguettaya, "Efficient Access to Web Services," *IEEE Internet Computing,* vol. 8, no. 2, pp. 34-44, Mar. 2004.

[19] M. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," *Comm. ACM,* vol. 46, no. 10, pp. 25-28, 2003.

[20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '94),* pp. 175-186, 1994.

[21] S. Rosario, A. Benveniste, S. Haar, and C. Jard, "Probabilistic QoS and Soft Contracts for Transaction-Based Web Services Orchestrations," *IEEE Trans. Services Computing,* vol. 1, no. 4, pp. 187-200, Oct. 2008.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l Conf. World Wide Web (WWW '01),* pp. 285-295, 2001.

[23] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized Qos Prediction for Web Services via Collaborative Filtering," *Proc. IEEE Fifth Int'l Conf. Web Services (ICWS '07),* pp. 439-446, 2007.

[24] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating Word of Mouth," *Proc. SIGCHI Conf. Human Factors in Computing Systems,* 1995.

[25] R.M. Sreenath and M.P. Singh, "Agent-Based Service Selection," *J. Web Semantics,* vol. 1, no. 3, pp. 261-279, 2003.

[26] H. Sun, Y. Peng, J. Chen, C. Liu, and Y. Sun, "A New Similarity Measure Based on Adjusted Euclidean Distance for Memory-Based Collaborative Filtering," *J. Software,* vol. 6, no. 6, pp. 993-1000, 2011.

[27] H. Sun, Z. Zheng, J. Chen, and M. Lyu, "NRCF: A Novel Collaborative Filtering Method for Service Recommendation," *Proc. IEEE Ninth Int'l Conf. Web Services (ICWS '11),* pp. 702-703, 2011.

[28] N. Thio and S. Karunasekera, "Automatic Measurement of a QoS Metric for Web Service Recommendation," *Proc. Australian Conf. Software Eng.,* pp. 202-211, 2005.

[29] J. Wang, A.P. de Vries, and M.J. Reinders, "Unifying User-Based and Item-Based Collaborative Filtering Approaches by Similarity Fusion," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06),* pp. 501-508, 2006.

[30] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing.* Springer, 2007.

[31] Z. Zheng, H. Ma, M.R. Lyu, and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommender System," *Proc. IEEE Seventh Int'l Conf. Web Services (ICWS '09),* pp. 437-444, 2009.

[32] Z. Zheng, H. Ma, M.R. Lyu, and I. King, "QoS-Aware Web Service Recommendation by Collaborative Filtering," *IEEE Trans. Service Computing,* vol. 4, no. 2, pp. 140-152, Apr.-June 2011.

[33] Z. Zheng, Y. Zhang, and M.R. Lyu, "Distributed QoS Evaluation for Real-World Web Services," *Proc. IEEE Eight Int'l Conf. Web Services (ICWS '10),* pp. 83-90, 2010.