Eng Int Syst (2020) 1: 31–40
© 2020 CRL Publishing Ltd

Engineering
Intelligent Systems

Collaborative Prediction of Web Service Quality Based on User Preference and Service

Yang Song[1]* **and Zhanglong Nie**[2]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Software and Big Data, Changzhou College of Information Technology, Changzhou, 213164, China*

The prediction of Web service quality plays an important role in improving user services. Therefore, it has been one of the most popular topics in the field of Internet services. In traditional collaborative filtering methods, the differences in personalization and preferences of different users have been ignored. In this paper, for different types of quality of service (QoS) attributes, different extraction rules are applied to extract the user preference matrices from the original Web data, and the negative value filtering-based Top-K method is used to merge the optimization results into the collaborative prediction method. In doing so, the individualized differences have been fully exploited, and the problem of inconsistent QoS values has been resolved. The experimental results demonstrate the validity of the proposed method. Compared with previous methods, the proposed method performs better, and the results are closer to the real values.

Keywords: web service; quality of service; collaborative filtering; user preferences; Internet

1. INTRODUCTION

Web service is a low-coupling and reusable network software that is independent of programming languages and operation platforms. By displaying an application interface to the outside world for Web invocation, developers can use Web services without knowing the details of the software implementation. Starting from the concept of software as a service, the emergence of Web services has also led to the innovation of the software application model, which has shifted from the traditional software development mode to the use of Web services to achieve maximum software integration, demonstrating great development potential. The emergence of Web services, and their gradual innovation, has greatly steered the development of distributed computing in an efficient and accurate direction [1]. At the same time, the increasing popularity of service-oriented computing (SOC) [2] has brought new vitality to Web services of different functions and the seamless connection between services and commercial software. Therefore, Web service has gradually become a popular research topic.

Developers can arbitrarily publish Web services with various features. When users consider their own needs, there will be many optional services, which may lead to selection difficulties. In this case, the accurate prediction and selection of Web services are particularly important. As a performance description method for Web services, quality of service (QoS) generally uses the idea of collaborative filtering to determine the relationship between different users and their preferences. This idea was first proposed by Goldberg et al. [3], and has been applied in some e-commerce platforms such as Amazon [4], Time Network and so on. Inspired by this idea, collaborative filtering technology has gradually entered the field of Web service selection. Shao et al. [5] proposed a method for collaborative prediction utilizing the similarity of different users. Firstly, find neighbors similar to the target users, then perform collaborative prediction based on the invocation records of the neighbors. Vadivelou et al. [6] and Liu et al. [7] also adopted the collaborative filtering method. Li et al. [8] proposed a method based on

*Email: lee0919k@163.com

Table 1 Response time.

	s_1	s_2	s_3	s_4
u_1	0.1	0.2	0.1	0.2
u_2	0.1	0.2	*null*	0.2
u_3	0.1	0.2	0.1	0.4

a comprehensive consideration of different users and service groups. By introducing a weighting factor, the user preference and QoS prediction structure are combined, namely WSRec, to improve the accuracy of prediction results. In order to further improve the prediction of Web services performance, some researchers have added other factors such as geographic locations and network environments. For example, Zhang et al. [9] proposed a method combining user input and usage experience. Hu et al. [10] proposed a method considering time as a factor. Other studies [11–13] considered the influence of geographical locations in order to improve prediction accuracy, using different strategies according to the relative distances between the users and location of the services.

The optimization methods mentioned above improve the prediction accuracy to a certain extent. However, there are also some limitations, such as lack of comprehensive utilization and detailed analysis, ignoring the differences in the acceptable ranges of different users for Web services. Therefore, in this paper we associate user preferences with Web service quality (QoS) values, and propose a Web service collaborative prediction method to improve the prediction accuracy.

2. PROBLEM DESCRIPTION

Conventional collaborative filtering methods [5–8] have mostly focused on the algorithm itself while ignoring the individual differences and preferences of the users. Here, the individual preferences correspond to the acceptable ranges of QoS data for different users, which may also be referred to as the preference range [14]. For instance, two users may have experienced the same response time after invoking the same service. One user was very satisfied with the response time, while the other considered it as a timeout. This is because the two users have different preferences. Different users may have different QoS experiences due to network conditions and other factors. In general, the shorter the response time, the better the user satisfaction. Therefore, the upper limit of the preference range is defined as the lowest response time record, and the lower limit of the preference range is defined as the highest response time record. Similarly, when a user invokes the service, the higher the reliability, the better the user satisfaction. Therefore, the upper limit and the lower limit of the preference range are defined as the highest reliability record and the lowest reliability record, respectively. Users often prefer to have services with a shorter response time, lower prices and greater reliability. However, different users may have different application backgrounds, geographical locations and network conditions, so their preference ranges in terms of QoS attributes can vary accordingly.

For three users under different network conditions, the response time records after accessing four different services are taken as an example, as shown in Table 1.

In this case, user u_2's invocation information of service s_3 is missing. Therefore, we treat user u_2 as the target user, and service s_3 as the target service, to predict u_2's QoS value for response time with regard to service s_3. To calculate the similarity, the conventional collaborative filtering algorithms is intended to find the users who have visited service s_3 and have invocation records in common with user u_2. As mentioned earlier, the most commonly-used similarity calculation method is the Pearson Correlation Coefficient method. The Pearson Correlation Coefficient is used to calculate the similarity between the users u_2 and u_1, and the result is 1. The similarity between the users u_2 and u_3 is calculated in the same way, and the result is also 1. The reason is that the response time of the three users invoking all the services except the target service is exactly the same. Hence, based on the QoS values of the three services, user u_2 has the same similarity as users u_1 and u_3. Therefore, the predicted value of QoS for user u_2 invoking the service s_3 is 0.1.

However, users and servers are scattered across the world and interact through the Internet. The uncertainty of multiple factors may cause different users to obtain different QoS feedback data when invoking the same service, and the same user may experience different QoS performance when invoking different services. This results in different preference ranges of QoS values for different users. For example, given the preference ranges of response time for user u_1, u_2 and u_3 as [0.1,1], [0.08,1.5] and [0.1,3], respectively (in seconds). As described above, in the similarity calculation, if the original QoS data is directly used for prediction and the individualized differences of different users are ignored, the result will be inaccurate. The QoS preference range should be incorporated into the calculation, and different preference ranges should make different contributions to the prediction. Therefore, this paper proposes a collaborative preference prediction method (PFPre) that links the user preference range with QoS values to improve the accuracy of prediction.

3. PREFERENCE DATA EXTRACTION RULES

In the following scenario, a user accessing a service will receive K QoS feedback values for response time, availability and throughput, etc. These QoS values can be encapsulated into a K-dimensional vector. Then the access records of M users to N services can be expressed by a matrix $Matrix_{MN}$:

$$Matrix_{MN} = \begin{pmatrix} l(u_1,s_1), & l(u_1,s_2), & l(u_1,s_3), & \cdots, & l(u_1,s_N) \\ l(u_2,s_1), & l(u_2,s_2), & l(u_2,s_3), & \cdots, & l(u_2,s_N) \\ \vdots & \cdots & \vdots & & \\ l(u_M,s_1), & l(u_M,s_2), & l(u_M,s_3), & \cdots, & l(u_M,s_N) \end{pmatrix} \quad (1)$$

where $l(u_m, s_n)$ represents a $K-$dimensional vector containing K QoS values perceived by user $u_m$when accessing the service s_n. For each QoS attribute in $l(u_m, s_n)$, there is a two-dimensional matrix $M \times N$ corresponding to it. The response time matrix *Rtmatrix* can be written as:

$$Rtmatrix = \begin{pmatrix} r(u_1,s_1), & r(u_1,s_2), & r(u_1,s_3), & \cdots, & r(u_1,s_N) \\ r(u_2,s_1), & r(u_2,s_2), & r(u_2,s_3), & \cdots, & r(u_2,s_N) \\ \vdots & \cdots & \vdots & & \\ r(u_M,s_1), & r(u_M,s_2), & r(u_M,s_3), & \cdots, & r(u_M,s_N) \end{pmatrix} \quad (2)$$

where $r(u_m, s_n)$ represents the specific response time of user u_m invoking service s_n. If there is no invocation record, the QoS data has been lost, and this is expressed as $r(u_m, s_n)$ *=null*. As important indicators when measuring the quality of service, different QoS attributes would have different impacts on service quality. For example, cost-based attributes such as service price and response time should be as small as possible; on the other hand, benefit-based attributes such as service availability and reliability should be as high as possible. In order to standardize the preference range so that for each QoS attribute, the upper limit of the preference range indicates the highest user satisfaction, and the lower limit of the preference range indicates the lowest user satisfaction, the cost attributes and the benefit attributes should be considered separately. Based on the original QoS matrices, different extraction rules are applied to different types of attributes. The preference matrix formed from the extraction of cost-based attributes is defined as *PFMatrix*$^{\alpha}$, and the preference matrix formed from the extraction of benefit-based attributes is defined as *PFMatrix*$^{\beta}$.

For cost-based attributes, taking the response time as an example, the extraction rules of the data $r^{\alpha}(u_i,s_j)$ in the preference matrix *PFMatrix*$^{\alpha}$can be expressed as:

$$r^{\alpha}(u_i,s_j) = \begin{cases} 1 & , r(u_i,s_j) = Min(u_i) \\ \frac{Max(u_i)-r(u_i,s_j)}{Max(u_i)-Min(u_i)} & , Min(u_i) < r(u_i,s_j) < Max(u_i) \\ 0 & , r(u_i,s_j) = Max(u_i) \end{cases} \quad (3)$$

For benefit-based attributes, taking the reliability as an example, the extraction rules of the data $r^{\beta}(u_i,s_j)$ in the preference matrix *PFMatrix*$^{\beta}$can be expressed as:

$$r^{\beta}(u_i,s_j) = \begin{cases} 1 & , r(u_i,s_j) = Max(u_i) \\ \frac{r(u_i,s_j)-Min(u_i)}{Max(u_i)-Min(u_i)} & , Min(u_i) < r(u_i,s_j) < Max(u_i) \\ 0 & , r(u_i,s_j) = Max(u_i) \end{cases} \quad (4)$$

where *Min*(*ui*) and *Max*(*ui*) refer to the maximum and minimum QoS feedback values perceived by $u_i$for the services that the user has visited, respectively.

The values in the resulting preference matrix falls into the range of [0,1] regardless of the attribute types. The larger the value, the more satisfied is the user. Therefore, the individual user's preference in terms of the QoS data is fully considered. In addition, it can also be regarded as a special normalization process, which avoids the similarity calculation error caused by the inconsistency of the preference ranges.

4. COLLABORATIVE PREDICTION OF WEB SERVICE BASED ON USER PREFERENCES

4.1 Similarity Calculation Based on User Preferences

After the extraction of the user preference matrix, it is used to calculate the user preference-based similarity. Similarity calculation is a core component of the collaborative filtering algorithm. Firstly, similarity calculation directly relates to the sifting of similar neighbors, which is fundamental for finding high-quality neighbors. Secondly, the weighted sum of similarities of the similar neighbors is usually used in the prediction phase. Hence, similarity calculation also determines the amount of weight that is given to similar neighbors during the prediction process.

In the field of collaborative prediction, the most used similarity calculation methods are Pearson Correlation Coefficient [15], Tanimoto Coefficient [16] and Euclidean distance [17], etc. Euclidean distance is the simplest and the most straightforward similarity algorithm, and it can reflect the absolute difference of individual numerical characteristics. The value ranges in the resulting preference matrices have been standardized. Therefore, the similarity calculation can be performed according to the values of the preference data. Here, the Euclidean distance is chosen to perform similarity calculation, and the preference similarity between users $u_i$and $u_j$can be calculated as:

$$Sim_{Euc}(u_i,u_j) = \frac{1}{1+\sqrt{\sum\limits_{s \in Su_{ij}} (r^{\alpha}(u_i,s) - r^{\alpha}(u_j,s))^2}} \quad (5)$$

Where Su_{ij}=Su_i∩Su_j, Su_i and Su_j represent the service sets accessed by u_i and u_j respectively. Su_{ij} represents the overlapping portion of the services accessed by u_i and u_j, which is the intersection of two users' historical access records. $r^{\alpha}(u_i,s)$ and $r^{\alpha}(u_j,s)$ respectively represent the QoS values in the preference matrix for u_i and $u_j$after accessing each service s in the intersection set $su_{i,j}$,which is the extracted preference data. The similarity of service preferences can be calculated as:

$$Sim_{Euc}(s_i,s_j) = \frac{1}{1+\sqrt{\sum\limits_{s \in Us_{ij}} (r^{\alpha}(u,s_i) - r^{\alpha}(u,s_j))^2}} \quad (6)$$

where Us_{ij}=Us_i∩Us_j, Us_i and Us_j represent the overlapping set of the users who have invoked both the service S_i and S_j. $r^{\alpha}(u,s_i)$ and $r^{\alpha}(u,s_j)$ represent the QoS values in the preference matrix for user u after invoking the services S_i and S_j, respectively.

4.2 Negative-Value Based Similar Neighbor Filtering

In order to emphasize only the role of the preference matrix in the proposed algorithm, while ignoring any interference from other steps of the algorithm, this chapter uses the traditional Top-K algorithm to select similar neighbors. The Euclidean distance-based similarities are ranked from high to low, and the top Kmembers are selected as neighbors that will directly participate in the prediction. However, there is a loophole in this process. If the minimum similarities of the top K members are less than 0, then the weakly-correlated users would participate in the prediction, thereby increasing the prediction error. Therefore, the proposed method adds a negative-value filtering policy to the neighbour-choosing process. Users with a similarity less than 0 will be removed from the Top K members, to obtain the final set comprising the nearest neighbors.

4.3 Hybrid QoS Collaborative Prediction Based on Users and Services

Firstly, the final set of the nearest neighbors is sifted. Then the prediction of missing QoS attributes is performed based on the similarities between all users (services) and the target users (services) in this set.

Taking the cost-based attribute (e.g. response time) as an example, the collaborative prediction based on user preferences can be written as [18]:

$$\begin{aligned} P_U^\alpha(u_i, s_j) &= \bar{r}^\alpha(u_i) \\ &= \frac{\sum_{u \in Sim(u_i)} Sim_{Euc}(u_i, u)(r^\alpha(u, s_j) - \bar{r}^\alpha(u))}{\sum_{u \in Sim(u_i)} Sim_{Euc}(u_i, u)} \end{aligned} \tag{7}$$

where $\bar{r}^\alpha(u_i)$ and $\bar{r}^\alpha(u)$ are the mean values of the service feedback from users u_i and u in the preference matrix, respectively.

Similarly, the collaborative prediction based on the services can be written as:

$$\begin{aligned} P_S^\alpha(u_i, s_j) &= \bar{r}^\alpha(s_j) \\ &= \frac{\sum_{s \in Sim(s_j)} Sim_{Euc}(s_j, s)(r^\alpha(u_i, s) - \bar{r}^\alpha(s))}{\sum_{u \in Sim(s_j)} Sim_{Euc}(s_j, s)} \end{aligned} \tag{8}$$

where $\bar{r}^\alpha(s_j)$ and $\bar{r}^\alpha(s)$ are the mean values of the user feedback for using the services s_j and s in the preference matrix, respectively. $Sim(s_j)$ is the set of similar neighbors for service s_j. the data sparsity problem during the collaborative sifting process can be effectively alleviated by incorporating the user-based and the service-based collaborative filtering algorithms so as to improve the accuracy of QoS predictions. Therefore, this section also adopts hybrid QoS collaborative prediction based on users and services.

The final prediction based on user preferences can be written as:

$$P^\alpha(r(u_i, s_j)) = w_u \times P_U^\alpha(u_i, s_j) + w_s \times P_S^\alpha(u_i, s_j) \tag{9}$$

Where $P_U^\alpha(u_i, s_j)$ is the prediction result of user-preference based collaborative filtering, $P_S^\alpha(u_i, s_j)$ is the prediction result of service based collaborative filtering.

In the same way as (9), benefit-based attributes such as reliability can also be calculated.

4.4 Reduction Calculation of QoS Prediction Values

Firstly, the user preference information is extracted from the original user-service QoS attribute matrix, and the data is normalized in the range of [0, 1], and then collaborative prediction is carried out. By considering the individual differences among various users which are inevitable in real-world scenarios, we avoid the problem of users with different preference ranges being treated equally in the prediction, and the prediction error due to the inconsistent fluctuation ranges of the QoS values has been lowered.

It is worth mentioning that both the user-based and the service-based prediction values given by (6) and (7) are the results calculated with the user preference matrix. When analyzing the accuracy of the algorithm, they cannot be directly compared with the real values, but should be reduced to the predicted values under the original matrix, and then compared with the real values.

To normalize the preference ranges, the original matrix data is extracted by using (3) and (4), and map it into a user preference matrix. Hence, the reduction calculation can be seen as the process of using the known image (i.e. the predicted values under the preference matrix) to find the inverse image (i.e. the predicted values under the original matrix) according to the corresponding map (preference extraction rules). Therefore, for cost-based attributes such as response time, the reduction rules of the QoS prediction values calculated by the collaborative filtering algorithm can be written as:

$$\begin{aligned} Pre^\alpha(r(u_i, s_j)) = Max(u_i) &- P^\alpha(r(u_i, s_j)) \\ &\times (Max(u_i) - Min(u_i)) \end{aligned} \tag{10}$$

Similarly, for benefit-based attributes, the reduction rules of the QoS prediction values calculated by the collaborative filtering algorithm can be expressed as:

$$\begin{aligned} Pre^\beta(r(u_i, s_j)) = Min(u_i) &+ P^\beta(r(u_i, s_j)) \\ &\times (Max(u_i) - Min(u_i)) \end{aligned} \tag{11}$$

Where $P^\alpha(r(u_i, s_j))$ and $P^\beta(r(u_i, s_j))$ are the predicted results of the cost-based attributes and benefit-based attributes, respectively. $Pre^\alpha(r(u_i, s_j))$ and $Pre^\beta(r(u_i, s_j))$ are the prediction values of the cost-based attributes and benefit-based attributes after the reduction calculation. $Min(u_i)$ is the minimum QoS feedback value of user u_i accessing the services, $Max(u_i)$ is the maximum QoS feedback value of user u_i accessing the services.

Algorithm 1 Collaborative prediction algorithm based on user preference ranges

```
**Input:**user-service matrix $RTMatrix_{MN}$, target user $u_i$, target service $s_j$, prediction tuning parameter $\lambda$, the number of the nearest neighbors $neighbor\_k$;
**Output:**QoS prediction values $Pre(r(u_i, s_j))$.
(01) $Sim[u_\alpha] \leftarrow$ □, $Sims_i$ □□; $N(u_\alpha)$ □□; $N(s_i)$ □□;
     $Pre^{\alpha}_{U}(r(u_i, s_j)) \leftarrow 0$;     // store the user-based collaborative prediction results.
     $Pre^{\alpha}_{S}(r(u_i, s_j)) \leftarrow 0$;     // store the service-based collaborative prediction results.
     $Pre^{\alpha}(r(u_i, s_j)) \leftarrow 0$;     // store the hybrid collaborative prediction results based on users and services
     $Pre(r(u_i, s_j)) \leftarrow 0$;     // store the reduction calculation results of the user preference-based prediction, i.e. the final prediction values.
(02) **for each** $r(u_m, s_n) \in RTMatrix_{MN}$ **do**
(03) $PFMatrix^{\alpha} \leftarrow preference(r(u_m, s_n))$;
          // extract user preference information by using the extraction rules.
(04) **end for**
(05) **for each** $u_i \in U$ **do**
(06) $sim(u_a,\ u_i) \leftarrow EucSimilarity(PFMatrix^{\alpha}, S(u_a,\ u_i))$;
          // calculate the preference similarities among users based on the preference matrix.
(07)**end for**
(08) $N(u_a) \leftarrow Top - K(sim(u_a),\ neighbor\_k)$;
          // sifting out the set of similar neighbors of users by using the negative-value filtering based Top-K algorithm.
(09) **for each** $(r^{\alpha}(u_i,\ s_j) \leftarrow -1) \in PFMatrix^{\alpha}$ **do**
(10) $Pre^{\alpha}_{U}(r(u_i,\ s_j)) \leftarrow Predict(r^{\alpha}(u_i,\ s_j),\ N(u_a), sim(u_a))$;
(11) **end for**
(12) **for** each $s_i \in S$ **do**
(13) $sim(s_a,\ s_j) \leftarrow EucSimilarity(PFMatrix^{\alpha}, U(s_a,\ s_i))$;
          //calculate the preference similarities among services by using the Euclidean method based on the preference matrix.
(14) **end for**
(15) $N(s_a) \leftarrow Top - K(sim(s_a), neighbor\_k)$;
          // sifting out the set of similar neighbors of services by using the negative-value filtering based Top-K algorithm.
(16) **for each** $(r^{\alpha}(u_i,\ s_j) \leftarrow -1) \in PFMatrix^{\alpha}$ **do**
(17) $Pre^{\alpha}_{S}(r(u_i,\ s_j)) \leftarrow Predict(r^{\alpha}(u_i,\ s_j),\ N(s_a), sim(s_a))$;
(18) **end for**
(19) $Pre(r(u_i,\ s_j)) \leftarrow MixPredict(Pre^{\alpha}_{U}(r(u_i,\ s_j)), Pre^{\alpha}_{S}(r(u_i,\ s_j)))$;
          //hybrid collaborative prediction based on users and services.
(20) $Pre(r(u_i,\ s_j)) \leftarrow ReductionCalculation\ (Pre^{\alpha}(r(u_i,\ s_j)))$;
          //reduction calculation of the prediction results
(21) **return** $Pre(r(u_i,\ s_j))$
**End**
```

4.5 Collaborative Prediction Algorithm Based on User Preferences

Algorithm 1 demonstrates the collaborative prediction algorithm of Web services based on user preferences.

Algorithm 1 can be divided into four parts: Part 1 contains line 1, which is used to initialize the variables and define the variables needed in the algorithm; Part 2 contains line 2 to line 4, which is used to extract the user preference matrix; Part 3 contains line 5 to line 18, which is used to calculate the preference similarities between the users (services), and to sift out similar neighbors by using the negative value filtering based Top-K algorithm, obtaining the set of nearest neighbors; Part 4 contains line 19 to line 21, which is used to calculate the hybrid collaborative prediction results based on users and services, and perform reduction calculation on the obtained prediction values to obtain the results based on original data.

5. EXPERIMENTS AND DISCUSSIONS

5.1 Experiment Preparation

1) Dataset. The experimental dataset was the QoSDataset2 from the publicly released WS-DREAM [19, 20], and response time was chosen as the QoS attribute in the experiments.
2) Performance Metrics. The Mean Absolute Error and Normalized Mean Absolute Error which are most commonly used in the rating prediction filed were chosen to evaluate the accuracy of the proposed algorithm.

5.2 Comparison of the Prediction Methods

To evaluate the effectiveness and accuracy of the proposed user preference-based Web service collaborative prediction

Table 2 Comparison of accuracy of PFPre and other prediction methods in terms of MAE (the smaller the MAE value, the higher the prediction accuracy).

datasets	method	MAE				
		d=10%	d=20%	d=30%	d=40%	d=50%
100 × 100	UPCC	0.3570	0.3167	0.2992	0.2914	0.2900
	IPCC	0.3526	0.3005	0.2664	0.2595	0.2438
	WSRec	0.3428	0.2962	0.2646	0.2582	0.2366
	PFPre	**0.2788**	**0.2738**	**0.2345**	**0.2232**	**0.2150**
100 × 150	UPCC	0.4753	0.4107	0.3805	0.3685	0.3668
	IPCC	0.4616	0.3985	0.3578	0.3419	0.3349
	WSRec	0.4403	0.3933	0.3522	0.34	0.3292
	PFPre	**0.3931**	**0.374**	**0.332**	**0.3066**	**0.2934**
150 × 100	UPCC	0.4801	0.4022	0.3828	0.3791	0.3697
	IPCC	0.4415	0.3707	0.3485	0.3417	0.331
	WSRec	0.4331	0.3699	0.3423	0.3319	0.3197
	PFPre	**0.3348**	**0.3116**	**0.2754**	**0.2485**	**0.2255**

Table 3 Comparison of accuracy of PFPre and other prediction methods in terms of NMAE (the smaller the NMAE value, the higher the prediction accuracy).

datasets	method	MAE				
		d=10%	d=20%	d=30%	d=40%	d=50%
100 × 100	UPCC	0.6416	0.5683	0.5273	0.5201	0.5182
	IPCC	0.6336	0.5392	0.4694	0.4632	0.4395
	WSRec	0.6160	0.5314	0.4663	0.4608	0.4264
	PFPre	**0.5010**	**0.4913**	**0.4132**	**0.3983**	**0.3876**
100 × 150	UPCC	0.7158	0.6157	0.5707	0.5613	0.5598
	IPCC	0.6952	0.5975	0.5367	0.5208	0.5111
	WSRec	0.6630	0.5896	0.5282	0.5179	0.5025
	PFPre	**0.5919**	**0.5608**	**0.498**	**0.467**	**0.4477**
150 × 100	UPCC	0.7330	0.6209	0.5834	0.5827	0.5711
	IPCC	0.6742	0.5723	0.5312	0.5252	0.5113
	WSRec	0.6613	0.5711	0.5217	0.5101	0.494
	PFPre	**0.5112**	**0.4811**	**0.4198**	**0.382**	**0.3484**

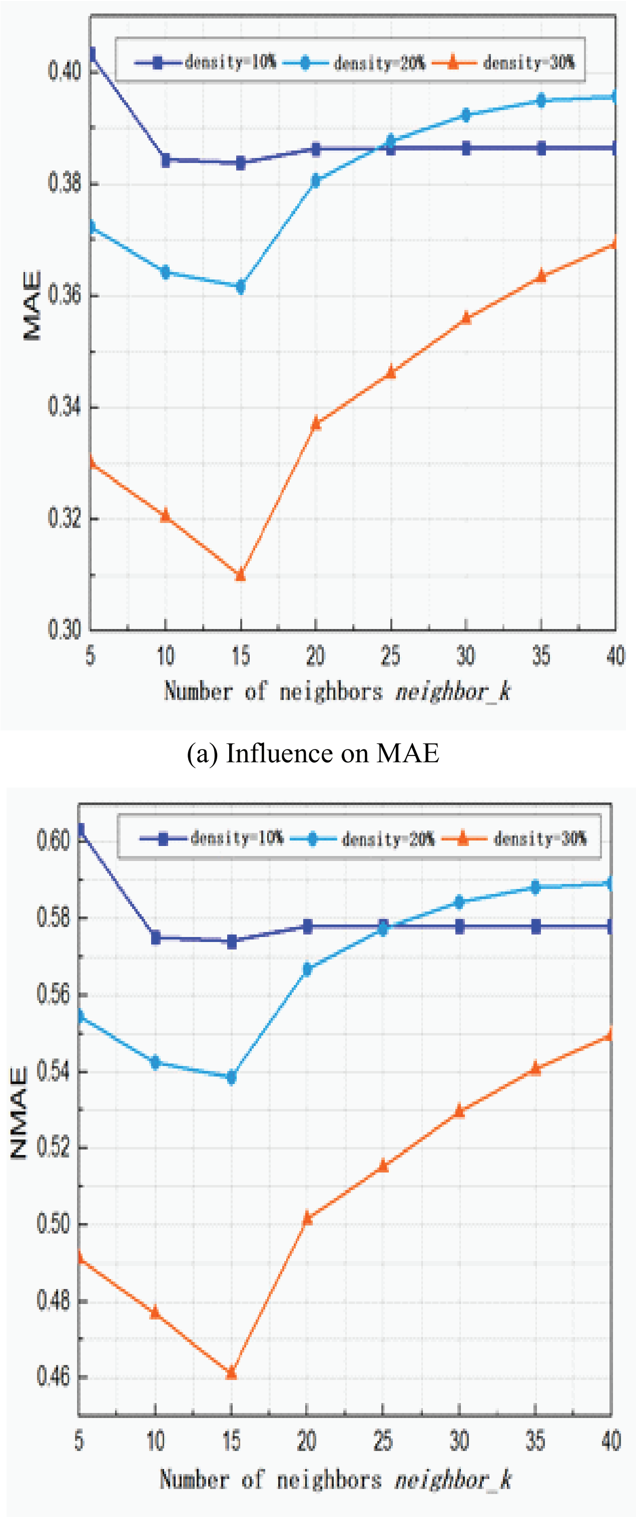
method PFPre, in this paper we have compared the performance of PFPre with the most commonly-used prediction methods, UPCC, IPCC and WSPre. In order to perform prediction, UPCC uses the Pearson Correlation Coefficient to calculate the similarity between users and find similar neighbors for the users, while IPCC uses the Pearson Correlation Coefficient to calculate the similarity between the services and find similar neighbors for the services. WSPre uses the weighted prediction results of UPCC and IPCC as the final prediction results.

Experimental parameters were set for the configuration of PFPre . In the experiment, the density of the dataset started from 10% and ended at 50% with an increment of 10%, $neighbor\_k = 15$, $\lambda = 0.3$. The settings for the parameters $neighbor\_k$ and λ will be discussed in detail in Section 5.3 and 5.4. To assess the adaptability of the proposed model, three user-service response time matrices with different sizes and structures of $size = 100 \times 100$, $size = 100 \times 150$, $size = 150 \times 100$ were constructed by randomly extracting a certain number of users and services. The mean absolute error MAE and the normalized mean absolute error NMAE were used to evaluate the accuracy of the algorithms. The experiment results are shown in Table 2 and Table 3.

It can be observed from Table 2 and Table 3 that: (1) as the density of the matrix sparsity increases, the MAE values of the four methods decrease, indicating that as the data becomes more dense, the prediction accuracy will be increased; (2) compared with other algorithms, the proposed PFPre method exhibits lower MAE and NMAE values with smaller errors under the same conditions, indicating that the proposed PFPre algorithm outperforms other traditional algorithms in terms of prediction accuracy.

5.3 Parameter Tuning for the Top-K Algorithm

During the process of sifting similar neighbors, the parameter $neighbor\_k$ in the Top-K algorithm controls the size of the nearest neighbor set. If the $neighbor\_k$ is too small, there will not be a sufficient number of neighbors, resulting in a lower prediction accuracy. If the $neighbor\_k$ is too big, some neighbors with weak correlation will be placed into the nearest neighbor set [21], resulting in similar neighbors making less contribution to the prediction. Therefore, it is necessary to find an appropriate value for the parameter $neighbor\_k$ to improve the prediction performance. To evaluate the influences that are imposed on the results by the changing $neighbor\_k$, we constructed a 100×150 user-service matrix by randomly extracting a certain amounts of users and services, where



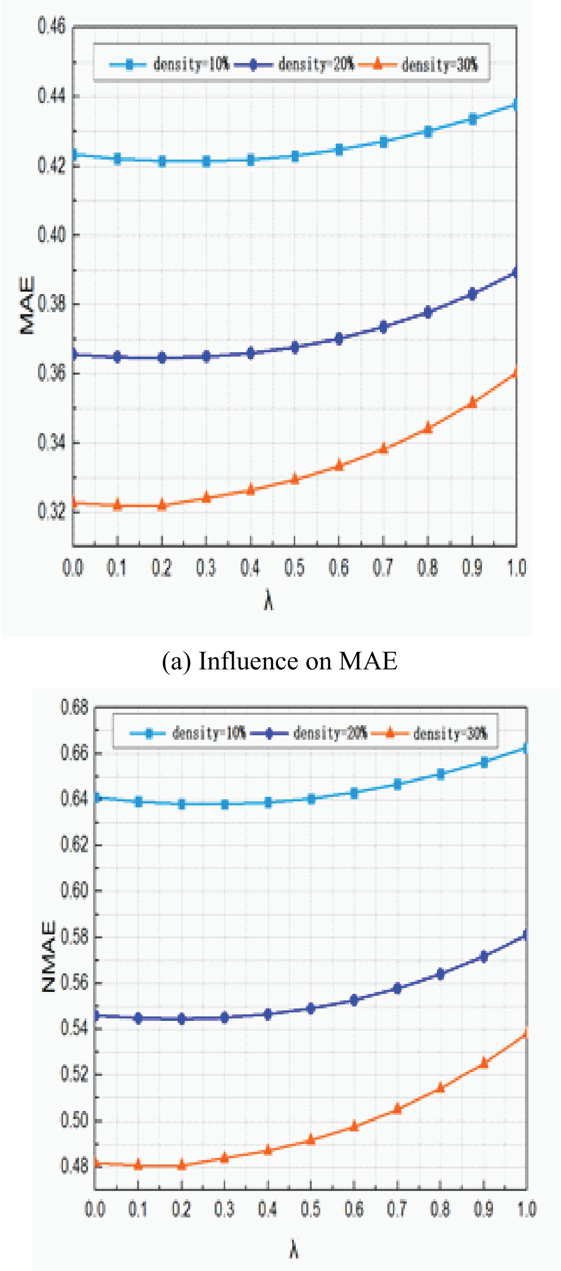
(a) Influence on MAE

(b) Influence on NMAE

Figure 1 The influence of different values of the parameter *neighbor_k* in the Top-K algorithm on the prediction accuracy

$\lambda = 0.3$, the value of *neighbor_k* was increased from 5 to 40 with an increment of 5, the data density was within the range of 10% to 30%. The results are shown in Fig. 1a and Fig. 1b.

It can be seen from the figures that when the value of *neighbor_k* is 15, the minimum error has been achieved under all the scenarios with different densities, indicating the optimal prediction performance. Hence, the value of *neighbor_k* will be set to 15 for the rest of the experiments. When the sparsity density is 10%, the prediction results remain basically the same after the value of *neighbor_k* becomes greater than 25. The reason is that the negative value filtering strategy has been used during the sifting of similar neighbors. Under the scenarios with little density, the number of neighbors



(a) Influence on MAE

(b) Influence on NMAE

Figure 2 The influence of different values of the parameter λ on the prediction accuracy.

to be chosen is relatively small, and the neighbors with a similarity less than 0 have been excluded. Hence, after the value of *neighbor_k* has been increased to a certain degree, the variations in the final nearest neighbor set will be small, leading to a very small fluctuation in the prediction results.

5.4 Parameter Tuning for the Hybrid Prediction

The parameter λ is responsible for adjusting the proportion of user-based and service-based predictions in the PFPre training model. We constructed a 100 × 150 user-service matrix by randomly extracting a certain amounts of users and services, where *neighbor_k* was set to 15, the value of λ was in the range of 0 and 1 with an increment of 0.1, and the predictions were performed under scenarios of various densities: 10%, 20% and 30%.

It can be observed that the promising prediction results were achieved under different scenarios when the value of λ was set to 0.3. The influence of different values of the parameter λ on the prediction accuracy is shown in Figure 2.

5.5 Comparison of Recommendation Instances

To further verify the proposed user preference-based QoS collaborative prediction method and its practical significance, this section will introduce the experiments performed using specific instances. To analyze the advantage of disadvantage of each algorithm, different methods were used to recommend

Table 4 Recommendation results (in seconds).

No.	Address	Actual Value	PFPre prediction value (sorted)	WSRec predicted value (sorted)	PFPre error	WSRec error
86	http://forums.genom-e.com/_vti_bin/ BusinessDataCatalog.asmx?wsdl	0.038	0.3298	1.5114	0.2918	1.4734
87	http://forums.genom-e.com/ _vti_bin/Sharepoint EmailWS.asmx?wsdl	0.039	0.3366	2.1730	0.2976	2.1340
67	http://www.taet.com.au/Web +Service+Agent.nsf/Random DirectoryWSDL	0.047	0.3408	2.1960	0.2938	2.1490
100	http://www.arikan.at/axis/services/ SOAPMonitorService?wsdl	0.087	0.3677	1.4537	0.2807	1.3667
38	http://services1.pharmx.com.au/ order/supplierlist.asmx?WSDL	0.121	0.4257	1.5987	0.3047	1.4777
45	http://api.godo.com.au/soap/v1_00/ product.cfc?WSDL	0.121	0.5277	1.5559	0.4067	1.4349
95	http://www.arikan.at/axis/services/ CountryService?wsd	0.122	0.4521	1.5524	0.3301	1.4304
62	http://www.stratapay.com.au/ ecommservices.asmx?WSDL	0.142	0.6063	1.7462	0.4643	1.6042
58	http://www.myboot.com.au/webservice/ shared/galleries/resources.asmx?WSDL	0.145	0.6010	1.7015	0.456	1.5565
59	http://www.myboot.com.au/webservice/ myboot/suburb/picker.asmx?WSDL	0.145	0.5959	1.5525	0.4509	1.4075

high performance services for the same user and the recommendation results were compared. Take the QoS attribute of response time as an example, where the density of the dataset is set to 30%. For PFPre, $\lambda = 0.3$, $neighbor\_k = 15$. For conventional prediction method WSRec, $\lambda = 0.3$. The two algorithms were performed to give a recommendation to the same user in the dataset based on service performance (in this case it means the response time). Usually, the user prefers to have high performance services than are reliable and give prompt responses. Subsequently, the actual values of the response time were ranked in ascending order in order to compare the difference in the recommendation instances obtained from each algorithm. The top 10 Web services were chosen for the comparison. The results are shown in Table 4.

It can be seen from Table 4 that, compared with WSRec, the proposed PFPre performs better in terms of prediction error. For the ranking lists based on the actual values of the response time in ascending order, the ranking list of the predicted values obtained from the proposed PFPre method exhibits more consistency with the ranking lists of the actual values. Therefore, the proposed method is suitable for QoS-based Web service recommendation platforms.

6. SUMMARY

This paper explains the prediction error in the conventional collaborative filtering methods that results from not knowing the range of differences in user preferences. Therefore, this paper introduces a definition of the user preference range. For different types of QoS attributes, different extraction rules were used to extract user preference matrices from the original QoS data. Based on this, the Euclidean similarity was used, instead of the conventional Pearson similarity, to perform calculations. During the process of sifting similar neighbors, the negative-value based Top-K method was used, and all the optimized results were incorporated into the final collaborative prediction method. Finally, the reduction calculation was performed on the prediction results. The main advantage of the proposed algorithm is that it can fully extract the individual differences among various users, and overcome the problem of inconsistent value ranges of QoS attributes, thereby avoiding the prediction errors caused by directly using the QoS data.

ACKNOWLEDGMENT

This article is supported by Jiangsu Province Higher Vocational Education High Level Backbone Major (Computer Application Technology) Project.

REFERENCES

1. Papazogolou M P, Traverso P, Dustdar S, et al. Service-oriented Computing: A Research Roadmap. International Journal of Cooperative Information Systems, 2008, 17(2): 223–255.
2. Valilai O F, Houshmand M. A Collaborative and Integrated Platform to Support Distributed Manufacturing System Using a Service-oriented Approach Based on Cloud Computing Paradigm. Robotics and Computer-Integrated Manufacturing, 2013, 29(1): 110–127.

3. Goldberg D, Nichols D, Oki B M, et al. Using Collaborative Filtering to Weave Information Tapestry. Communications of the ACM 1992, 35(2): 61–70.
4. Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In: Proceedings of IEEE Internet Computing, Washington, D.C., USA: IEEE Press, 2003, pp. 76–80.
5. Zhou Z, Wang B, Jie G, et al. QoS-Aware Web Service Recommendation Using Collaborative Filtering with PGraph. In: IEEE International Conference on Web Services, 2015, 140–152.
6. Vadivelou G. Collaborative filtering-based web service recommender system using users' satisfaction on QoS attributes. In: International Conference on Inventive Computation Technologies. 2017, pp. 23–31.
7. Liu J, Tang M, Zheng Z, et al. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation. IEEE Transactions on Services Computing, 2016, 9(5): 686–699.
8. Li S, Wen J, Luo F, et al. A New QoS-Aware Web Service Recommendation System Based on Contextual Feature Recognition at Server-Side. IEEE Transactions on Network & Service Management, 2017, 14(2): 1–12.
9. Zhang L, Zhang B, Liu Y, et al. A Web Service QoS Prediction Approach Based on Collaborative Filtering. In: Proceedings of IEEE Asia-Pacific Services Computing Conference. Washington, D.C., USA: IEEE Press, 2010, pp. 725–731.
10. Hu Y, Peng Q, Hu X. A Time-Aware and Data Sparsity Tolerant Approach for Web Service Recommendation. In: IEEE International Conference on Web Services. Anchorage, AK, USA: IEEE Computer Society, 2014, pp. 33–40.
11. Hu Y, Peng Q, Hu X, et al. Web Service Recommendation Based on Time Series Forecasting and Collaborative Filtering. In: IEEE International Conference on Web Services. New York, NY, USA: IEEE, 2015, pp. 233–240.
12. Wang Y, Qiang H, Zhang X, et al. Efficient QoS-Aware Service Recommendation for Multi-Tenant Service-Based Systems in Cloud. IEEE Transactions on Services Computing, 2017, PP(99): 1–1.
13. Yu C, Huang L. A Web Service Qos Prediction Approach Based on Time- and Location-Aware Collaborative Filtering. Service Oriented Computing & Applications, 2016, 10(2): 135–149.
14. Kumar S, Sarkar M, Vora M. Dynamic Management of QoS for Multimedia Multicasting. In: International Wireless Communications & Mobile Computing Conference. 2008, pp. 27–38.
15. Benesty J, Chen J, Huang Y, et al. Pearson Correlation Coefficient. Noise Reduction in Speech Processing. 2009.
16. Freire F, Ellen, Borges B, et al. Development and Validation of a Simple Method for Routine Analysis of Ractopamine Hydrochloride in Raw Material and Feed Additives by HPLC. Journal of AOAC International, 2009, 92(3): 102–111.
17. Hao W, Yue K, Bo L, et al. Collaborative QoS Prediction with Context-Sensitive Matrix Factorization. Future Generation Computer Systems, 2018, 82(C): 57–68.
18. Jeong B, Lee J, Cho H. Improving Memory-Based Collaborative Filtering via Similarity Updating and Prediction Modulation. Information Sciences, 2010, 180(5): 602–612.
19. Zheng Z, Zhang Y, Lyu M L. Investigating QoS of Real-World Web Services. IEEE Transactions on Service Computing, 2014, 7(1): 32–39.
20. Zheng Z, Zhangy Y, Lyu M L. Distributed QoS Evaluation for Real-World Web Services. In: Proceedings of the 8^{th} International Conference on Web Services. Miami, Florida, USA, 2010: 83–90.
21. Bao Y, Du X, Ishii N. Improving Performance of the k-Nearest Neighbor Classifier by Combining Feature Selection with Feature Weighting. Transactions of the Japanese Society for Artificial Intelligence, 2002, 17(3): 209–216.