QoS-Aware Web Service Recommendation with Reinforced Collaborative Filtering

Guobing Zou[1,2], Ming Jiang[1,2], Sen Niu[3], Hao Wu[1], Shengye Pang[1], and Yanglan Gan[4](✉)

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
guobingzou@gmail.com, jimmyme520@gmail.com
[2] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China
[3] College of Computer and Information, Shanghai Polytechnic University, Shanghai, China
[4] School of Computer Science and Technology, Donghua University, Shanghai, China
ylgan@dhu.edu.com

Abstract. With the overwhelming increase of web services on the Internet, how to accurately perform QoS prediction has played a key role in service recommendation. Recently, three kinds of approaches have been presented on service QoS prediction based on collaborative filtering (CF), including user-intensive, service-intensive and their combination. However, the deficiency of current approaches is that all of the services invoked by target user (or all of the users who invoked target service) are applied to calculate average QoS, without the reduction to those dissimilar with target service (or target user). In this paper, we propose a reinforced collaborative filtering approach, where both similar users and services are integrally considered into a singleton CF. The experiments are conducted on a large-scale dataset called WS-DREAM, involving 5,825 real-world Web services in 73 countries and 339 service users in 30 countries. The experimental results demonstrate that our approach for QoS prediction outperforms the competing approaches.

Keywords: Service-oriented computing · Service recommendation
QoS · Collaborative filtering

1 Introduction

With the rapid development of network technology and increasing demands on service-oriented application integration, more and more software developers publish their softwares as web services on the Internet. It accelerates the interoperable machine-to-machine interaction and greatly promotes the advancements on service discovery, optimum selection, automated composition and recommendation. However, as the overwhelming explosion on the number of the registered

C. Pahl et al. (Eds.): ICSOC 2018, LNCS 11236, pp. 430–445, 2018.
https://doi.org/10.1007/978-3-030-03596-9_31

services, many of them merge with the same or similar functionality in a service repository. That tends to be a labor-intensive challenging task for service requesters to choose their desired services from a large-scale service repository. Quality of Service (QoS) as a non-functionality criterion has been widely applied as a key factor to differentiate those functionally equivalent web services. In many cases, however, a target user has not invoked a target service. That is, there are very few historical QoS invocations, leading to difficulty in recommending appropriate services to a target user. In this application scenario, how to designing an effective approach of unknown QoS prediction for service recommendation has become a critical research issue to be addressed.

Applying the techniques of collaborative filtering in recommender system, correlative research efforts [1–7,9,10,12–14] have been made on service QoS prediction. They can be grouped into model-based and memory-based approaches. Matrix factorization (MF) [4–6,12,14] as the typical model-based technique has been used for missing QoS prediction, where the original user-service QoS matrix with sparsity is converted into two low dimensional matrices. Memory-based collaborative filtering approaches for QoS prediction are usually classified into user-based [2], service-based [7], and their linear combination with a confidence weight [13]. They leverage similar users (or services) for a target user (a target service) to predict unknown QoS.

Although these existing approaches can assist and facilitate QoS prediction of web services, the deficiency is that they still cannot reach the accuracy satisfying service requesters' demands. More specifically, the traditional collaborative filtering algorithm consists of two major components, including average QoS calculation and deviation migration calculation. It is observed that the existing approaches either directly employ all the services that a target user has invoked, or all the users who have invoked a target service, when performing average QoS calculation. That is, they did not consider the discrepancies between all of the services invoked by a target user and a target service. Simultaneously, the differences between all of the users who invoked a target service and a target user have also not been considered. That kind of collaborative filtering algorithm decreases the purity on average QoS calculation and affects the accuracy of QoS prediction. To partially solve this issue, enhanced collaborative filtering algorithms have been proposed with the help of external context information, such as a user's geographical location. However, it is difficult to obtain context information in real-world applications. Therefore, how to design an effective collaborative filtering algorithm for QoS prediction without any external heuristic information support has become a challenging research issue.

To handle above issue, we proposed a novel reinforced collaborative filtering algorithm (RECF) for QoS prediction. When performing user-based RECF, the implicit context information from the service side is obtained by a ratio-based similarity calculation method [10] in order to eliminate those dissimilar services with the target one. It has been integrally taken into account for calculating average QoS. In this way, the advantage of our approach is that user-based and service-based similarity information can be integrated into a singleton collaborative filtering algorithm without additional parameters learning and estimation. Similarly, service-based RECF takes those dissimilar users with the target one as hidden factors to promote QoS prediction.

To test the performance of QoS prediction, extensive experiments are conducted on a large-scale real-world dataset called WS-DREAM, involving 5,825 real-world web services in 73 countries and 339 service users in 30 countries. We compare our approach with seven existing collaborative filtering-based methods on QoS prediction accuracy. The experimental results demonstrate that our approach can outperform those competing approaches.

The main contributions of this paper are summarized as follows.

- We propose a novel reinforced collaborative filtering framework for QoS prediction, where the similarity calculation from user-perspective and service-perspective are integrally integrated into a singleton CF method, instead of a linear combination by weighted parameters.
- We propose an approach for optimizing the average QoS calculation. By applying ratio-based similarity computation, implicit features among users or services can be discovered without any external context information.
- We design and implement a prototype system and conduct extensive experiments on a real-world dataset called WS-DREAM. The experimental results demonstrate that our approach of QoS prediction is superior to existing completing methods in terms of accuracy.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 elaborates our approach of reinforced collaborative filtering for service QoS prediction. Section 4 shows the experimental evaluation. Finally, Sect. 5 concludes the paper.

2 Related Work

According to [2], collaborative filtering algorithms for service QoS prediction can be mainly divided into two categories, including model-based and memory-based approaches. We review the advancement on collaborative filtering algorithms for QoS prediction that is highly related with our work.

In model-based collaborative filtering approaches, matrix factorization [4,5,12,14] is the typical technique for predicting missing QoS values of web services. It turns an original matrix into two low dimensional matrices to reveal pivot features that can be used to estimate an unknown QoS value, where a target user has not ever invoked a target service.

In memory-based collaborative filtering approaches, it consists of average QoS calculation step and deviation migration step. This kind of approach has three different variations, including user-based CF [2], service-based CF [7] and their linear combination with confidence weights [13]. Based on historical QoS invocation logs, neighborhood users or services can be chosen by similarity calculation in deviation migration step. Moreover, correlative research works concentrated on how to more accurately quantify the correlation between users or services. The authors in [9] investigated the QoS distribution characteristics and proposed a novel collaborative filtering approach for QoS prediction. It normalizes the QoS values to the same range and then unifies the similarity in different multi-dimensional vector spaces. The authors in [10] proposed a novel ratio-based similarity approach to measure neighborhood users and services. Compared with PCC similarity [8] and cosine similarity [7], it is more precise for predicting the unknown QoS of web services.

To further improve the prediction accuracy, several recent approaches have been proposed by adding context information as heuristic knowledge during the procedure of deviation migration. The authors in [4,12,14] took the geography location of the users or services, the provider of the services and the infrastructure information into account to extract more accurate neighborhood users or services. The authors in [1] found out a group of trusted users or services based on pareto dominance comparison. In addition, time series was used to predict the tendency information of missing QoS prediction of web services [3]. The authors in [11] proposed a hybrid approach of QoS prediction via combining the information from geography location and time series.

From the above investigation, we observe that the existing methods mainly focused on how to improve the accuracy of similarity calculation in deviation migration. However, they rarely made the reduction to those dissimilar services invoked by a target user (or dissimilar users who invoked a target service) in average QoS calculation. Although some of the recent research has borrowed external context information as heuristic knowledge to optimize the similar services for a target user (or similar services for a target user), they only apply them to deviation migration, instead of average QoS calculation. That affects the prediction accuracy of QoS prediction. The ideal way of overcoming the problem is to design an effective approach to eliminate those dissimilar services (or users) for more accurate QoS prediction of web services.

3 Reinforced Collaborative Filtering for QoS Prediction

In this section, we first formulates the problem of QoS prediction. Then, the framework of our approach is illustrated. Finally, we elaborate user-based and service-based reinforced collaborative filtering approach, respectively.

3.1 Problem Formulation

Definition 1 (Service Ecosystem). In a web service ecosystem, $M = <U, I, R>$, $U = \{u_1, u_2, \ldots\}$ is a set of users and $I = \{i_1, i_2, \ldots\}$ is a set of web services. $R = \{r_{u,i}\}_{m*n}$ is QoS matrix, where each entry $r_{u,i}$ represents the invocation QoS value when u invoked i.

For example, Table 1 illustrates a service ecosystem, where $U = \{u_1, u_2, \ldots, u_5\}$ and $I = \{i_1, i_2, \ldots, i_6\}$. $R = \{r_{u,i}\}_{5*6}$ is a QoS matrix and each entry represents the response time when a user u invokes a service i.

Table 1. Sample of user-service QoS matrix

U/I	i_1	i_2	i_3	i_4	i_5	i_6
u_1	0.0	0.228	0.237	0.0	0.0	0.0
u_2	0.453	0.0	0.0	0.649	0.0	0.0
u_3	0.0	0.0	0.0	5.782	0.239	0.25
u_4	0.249	0.0	5.294	0.366	0.0	0.285
u_5	0.0	0.288	0.0	0.0	0.328	5.93

Definition 2 (QoS Invocation Log). Given a service ecosystem $M = <U, I, R>$, a QoS invocation log is defined as 3-tuple $<u, i, r_{u,i}>$, where $u \in U$ is a user, $i \in I$ is a service, and $r_{u,i}$ is the QoS value when u invoked i.

Note that if an entry of a QoS invocation log is equal to 0, indicating that a user has not ever invoked a service. In such case, its QoS value need to be further predicted for use. The QoS prediction problem is defined as below.

Definition 3 (QoS Prediction Problem). Given a service ecosystem $M = <U, I, R>$, QoS prediction problem is defined as 3-tuple $Q = <M, u, i>$, where u is a target user, i is a target service and $r_{u,i}$ has no invocation log. The goal is to predict its QoS value $\widehat{r}_{u,i}$.

The solution to a QoS prediction problem is $<u, i, \widehat{r}_{u,i}>$. It indicates the predicted value when a target user invokes a target service. Based on a set of predicted QoS values, desired services can be recommended.

Definition 4 (Service Recommendation). Given a service ecosystem $M = <U, I, R>$, a target user u, and a set of functionally equivalent services I', it has no QoS invocation log from u to a service $i' \in I'$. Service recommendation problem is defined as 3-tuple $S = <M, u, I'>$ and the goal is to choose a subset of services from I' that the recommended services can be invoked with the best predicted QoS by u.

By predicting missing QoS values on each service in I', we have their predicted QoS values as

$$R_{u,I'} = \{<u, i'_1, \widehat{r}_{u,i'_1}>, <u, i'_2, \widehat{r}_{u,i'_2}>, \ldots\} \tag{1}$$

In terms of the ranking of predicted QoS values, a subset of web services can be recommended to a target user. Here, we mainly focus on predicting QoS value when a target user invokes a target service.

3.2 The Framework of Our Approach

Figure 1 illustrates the overall framework of our proposed approach. Given a target user and a target service, the procedure of task functionality goes through four stages, including finding similar users (or services), detecting neighbor services (or users), average QoS calculation, and deviation migration.

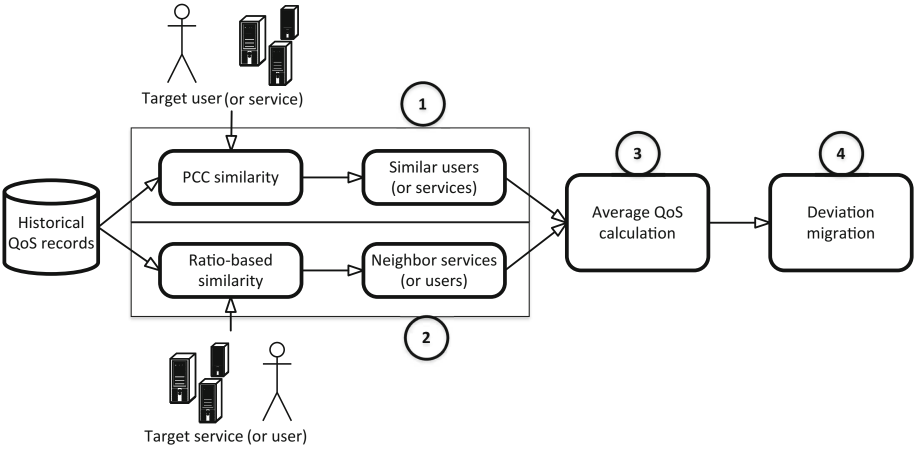

Fig. 1. The framework of our approach.

In the stage of finding similar users (or services), pearson correlation coefficient (PCC) is used to generate a group of similar users with the target user from historical QoS records (or similar services with the target service). In the stage of detecting neighbor services (or users), ratio-based similarity (RBS) is used to generate a group of neighbor services with the target service (or users with the target user). In the stage of average QoS calculation, taking all of the neighbor services (or neighbor users) as inputs, we calculate average QoS. In the stage of deviation migration, taking all of the similar users (or services) to calculate the deviation migration. Furthermore, the predicted comprehensive QoS for a target user invoking a target service can be finally calculated by integrally integrating average QoS value and deviation migration in a singleton CF.

3.3 User-Based Reinforced Collaborative Filtering

Given a target user and a target service, the procedure of user-based reinforced collaborative filtering approach for QoS prediction consists of finding similar users based on PCC, detecting neighbor services based on RBS, average QoS calculation and deviation migration.

(1) Finding Similar Users. Given a target user u, PCC [8] measures the correlation between a target user u and another user $v \in U$ in a service ecosystem.

$$Sim_{PCC}(u,v) = \frac{\sum_{i \in I_c} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_c} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_c} (r_{v,i} - \bar{r}_v)^2}} \qquad (2)$$

Where $I_c = I_u \cap I_v$ is the intersection of web services that both u and v have invoked previously, and $r_{u,i}$ is a vector of QoS values of service i observed by u. $\bar{r}_u$ and $\bar{r}_v$ represent average QoS values of different services observed by u and v, respectively.

By the PCC similarity calculation, we choose the k similar users with the highest similarity degree.

$$U_{PCC} = topSimilar(u) \tag{3}$$

As a result, U_{PCC} contains a set of similar users with target user u from U. In the deviation migration step, U_{PCC} is used to aggregate deviated QoS values.

(2) Detecting Neighbor Services. Given a target user u, it corresponds to a set of services I_u that have been invoked by u. In this step, we aim to reduce these services and only remain a subset of services that hold similar historical QoS invocation values with the target service i. Here, we filter out those services dissimilar with the target service i by the ratio-based similarity (RBS) [10].

$$Sim_{RBS}(i,j) = \frac{\sum\limits_{u \in U_c} \frac{\min(r_{u,i}, r_{u,j})}{\max(r_{u,i}, r_{u,j})}}{|U_c|} \tag{4}$$

Where U_c contains a set of common users who have invoked both i and j. $|U_c|$ is the number of users in U_c. $r_{u,i}$ and $r_{u,j}$ represent the QoS values while u invoked i and j, respectively. $\min(r_{u,i}, r_{u,j})$ and $\max(r_{u,i}, r_{u,j})$ calculate the minimum and maximum QoS values between $r_{u,i}$ and $r_{u,j}$, respectively.

Note that if the ratio-based similarity between i and j tends to be 1, it reflects that almost all the users are apt to obtain the highly close QoS values when they invoked these two services. In other words, they share implicit characteristics and provide similar invocation experiences, such as deployed in close geographical locations with the same network setting or hosted by the same service provider. However, we do not require the precisely underlying context information, since the neighborhood relationship among services can be evaluated by analyzing the historical invocation QoS values.

By setting a neighbor similarity threshold θ, we detect a subset of services that have been invoked by the target user u and share highly close invocation QoS values with the target service i.

$$I'_u = \{i' \in I_u \,|Sim_{RBS}(i', i) \geq \theta\} \tag{5}$$

After filtering out all of those dissimilar services from I_u, detected neighbor services is used in average QoS calculation step.

(3) Average QoS Calculation. Given a target user u, we make a reduction to all of the services I_u invoked by u and obtain a subset of neighbor services I'_u, where each service $i' \in I'_u$ has a high similarity degree with the target service i. Obviously, the target user could obtain similar QoS values with neighbor services when invoking the target service. Naturally, we use neighbor services to initially estimate the QoS value. For each $i' \in I'_u$, we calculate an initially predicted QoS for the target user u.

$$\hat{r}^{i'}_{u,i} = \begin{cases} r_{u,i'} * Sim_{RBS}(i,i') & \text{if } \bar{i} \leq \overline{i'} \\ r_{u,i'} / Sim_{RBS}(i,i') & \text{otherwise} \end{cases} \tag{6}$$

Where $\overline{i'}$ and $\bar{i}$ represent the average QoS value of service itself, respectively. $Sim_{RBS}(i, i')$ is the ratio-based similarity of the target service i and a neighbor service i'. By calculating the predicted QoS on each neighbor service $i' \in I'_u$, we get a group of predicted QoS values if the target user u invokes the target service i.

$$AvgQoS_u = \{\hat{r}_{u,i}^{i'_1}, \hat{r}_{u,i}^{i'_2}, \cdots, \hat{r}_{u,i}^{|I'_u|}\} \tag{7}$$

Taking the ratio-based similarity between two services as weight, we calculate the average QoS value for the target user u invoking the target service i.

$$\overline{u}_{avg} = \frac{\sum_{k=1}^{|I'_u|} Sim_{RBS}(i, i'_k) * \hat{r}_{u,i}^{i'_k}}{\sum_{k=1}^{|I'_u|} Sim_{RBS}(i, i'_k)} \tag{8}$$

The predicted average QoS value $\overline{u}_{avg}$ is still a preliminary result, as we mainly rely on the QoS values from those neighbor services similar to the target service, while similar users with the target user are not taken into consideration at this point. They are integrally integrated into the QoS prediction in deviation migration step.

(4) Deviation Migration. It calculates the QoS deviation of each similar user between the QoS obtained from invoking target service and its mean value, and then accumulates and migrates these deviation values to the average QoS for a target user.

First, based on found $U_{PCC} = \{u_1, u_2, \ldots, u_k\}$ for a target user u, we use Eqs. (4)–(7) to obtain k groups of average QoS values.

$$\begin{cases} AvgQoS_{u_1} = \{\hat{r}_{u_1,i}^{i'_1}, \hat{r}_{u_1,i}^{i'_2}, \cdots, \hat{r}_{u_1,i}^{|I'_{u_1}|}\} \\ AvgQoS_{u_2} = \{\hat{r}_{u_2,i}^{i'_1}, \hat{r}_{u_2,i}^{i'_2}, \cdots, \hat{r}_{u_2,i}^{|I'_{u_2}|}\} \\ \qquad\vdots \\ AvgQoS_{u_k} = \{\hat{r}_{u_k,i}^{i'_1}, \hat{r}_{u_k,i}^{i'_2}, \cdots, \hat{r}_{u_k,i}^{|I'_{u_k}|}\} \end{cases} \tag{9}$$

After that, we use Eq. (8) to calculate and generate a set of average QoS values for each similar user in U_{PCC}.

$$AvgQoS_{U_{pcc}} = \{\overline{u}_1, \overline{u}_2, \cdots, \overline{u}_k\} \tag{10}$$

Finally, applying collaborative filtering algorithm with $\overline{u}_{avg}$ and $Avg\ QoS_{U_{PCC}}$, we make the final QoS prediction.

$$\widehat{r}_{u,i} = \overline{u}_{avg} + \frac{\sum_{m=1}^{k} Sim_{PCC}(u, u_m) * (r_{u_m,i} - \overline{u}_m)}{\sum_{m=1}^{k} Sim_{PCC}(u, u_m)} \tag{11}$$

Where $\widehat{r}_{u,i}$ is the final predicted QoS value for a target user u when invoking a target service i.

3.4 Service-Based Reinforced Collaborative Filtering

Similar to user-based reinforced collaborative filtering, the procedure of service-based RECF approach also includes the same four steps.

(1) Finding Similar Services. Given a target service i, PCC measures the correlation between a target service i and another service in a service ecosystem. we use PCC to evaluate the correlation between two services.

$$Sim_{PCC}(i,j) = \frac{\sum_{u \in U_c}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U_c}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U_c}(r_{u,j} - \bar{r}_j)^2}} \tag{12}$$

Where $U_c = U_i \cap U_j$ is the intersection of users who have invoked both i and j previously, and $r_{u,i}$ is a vector of QoS values of service i observed by u. $\bar{r}_i$ and $\bar{r}_j$ represent average QoS values of i and j observed by a set of common users in U_c, respectively.

By the PCC similarity calculation, we choose the k similar services with the highest similarity degree.

$$I_{PCC} = topSimilar(i) \tag{13}$$

I_{PCC} consists of a set of similar services with target service i from I. In the deviation migration step, I_{PCC} is used to aggregate deviated QoS values.

(2) Detecting Neighbor Users. Given a target service i, it corresponds to a set of users U_i who have invoked i. In this step, we aim to eliminate those dissimilar users with a target user u from U_i, where they have low similar historical QoS values on their commonly invocated services. Here, we still apply the ratio-based similarity to measuring the neighborhood similarity degree.

$$Sim_{RBS}(u,v) = \frac{\sum\limits_{i \in I_c} \frac{\min(r_{u,i}, r_{v,i})}{\max(r_{u,i}, r_{v,i})}}{|I_c|} \tag{14}$$

It is observed that if the similarity degree $Sim_{RBS}(u,v) = 1$, it reflects that two users nearly received the same quality of QoS values when they invoked their commonly requested services in I_c. The implicit possibility is that these two users live in the same city and use the same network environment.

With a similarity threshold θ, we detect a subset of users who have invoked the target service i and obtain highly close invocation QoS values with the target user u.

Then, using a threshold to find out those users who are more similar to the target user u.

$$U_i' = \{u' \in U_i \,| Sim_{RBS}(u', u) \geq \theta\} \tag{15}$$

By the elimination of the dissimilar users from U_i, detected neighbor users is used in average QoS calculation step.

(3) Average QoS Calculation. Given a target service i, we make a reduction to all of the users U_i who invoked i and obtain a subset of neighbor users U_i', where each user $u' \in U_i'$ has a high similarity degree with the target user u. Similar to user-based reinforced collaborative filtering, we use neighbor users to initially estimate the QoS value. For each $u' \in U_i'$, we calculate an initially predicted QoS for the target service i.

$$\hat{r}_{u,i}^{u'} = \begin{cases} r_{u',i} * Sim_{RBS}(u,u') & \text{if } \overline{u} \leq \overline{u'} \\ r_{u',i} / Sim_{RBS}(u,u') & \text{otherwise} \end{cases} \tag{16}$$

Here, $\overline{u'}$ and $\overline{u}$ represent the average QoS value of user itself, respectively. $Sim_{RBS}(u,u')$ is the ratio-based similarity of the target user u and a neighbor user u'. By calculating the predicted QoS on each neighbor user $u' \in U_i'$, we get a group of predicted QoS values.

$$AvgQoS_i = \{\hat{r}_{u,i}^{u_1'}, \hat{r}_{u,i}^{u_2'}, \cdots, \hat{r}_{u,i}^{|U_i'|}\} \tag{17}$$

Taking the ratio-based similarity between two users as weight, we calculate the average QoS value for the target service i to be invoked the target user u.

$$\bar{i}_{avg} = \frac{\sum_{k=1}^{|U_i'|} Sim_{RBS}(u,u_k') * \hat{r}_{u,i}^{u_k'}}{\sum_{k=1}^{|U_i'|} Sim_{RBS}(u,u_k')} \tag{18}$$

In the same way, $\bar{i}_{avg}$ is still a preliminary result, as we mainly rely on the QoS values from those neighbor users similar to the target user, while similar services with the target service are not considered at this point. They are integrally integrated into the QoS prediction in deviation migration step.

(4) Deviation Migration. Based on found $I_{PCC} = \{i_1, i_2, \cdots, i_k\}$ for a target service i, we use Eqs. (14)–(18) to calculate and generate a set of average QoS values for each similar service in I_{PCC}.

$$AvgQoS_{I_{PCC}} = \{\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_k\} \tag{19}$$

Finally, applying collaborative filtering algorithm with $\bar{i}_avg$ and $AvgQo$ $S_{I_{PCC}}$, we make the final QoS prediction.

$$\widehat{r}_{u,i} = \bar{i}_{avg} + \frac{\sum_{m=1}^{k} Sim_{PCC}(i,i_m) * (r_{u,i_m} - \bar{i}_m)}{\sum_{m=1}^{k} Sim_{PCC}(i,i_m)} \tag{20}$$

Where $\widehat{r}_{u,i}$ is the final predicted QoS value for a target user u when invoking a target service i.

4 Experiments

4.1 Experimental Setup and Dataset

The experiments are conducted on a large-scale real-world dataset called WS-DREAM [15], involving 5,825 real-world Web services in 73 countries and 339 service users in 30 countries. This dataset consists of two QoS invocation matrices, one for response time and the other for throughput.

To validate the performance of our approach, we use the response time matrix to perform our experiments. We extract the matrix into 1,873,838 QoS invocation logs, after removing those invocations where a target user failed to access a target service. All the QoS invocation logs are partitioned into two parts, one for the training set and the other for the test set. During the experiments, the proportion of the number of QoS invocation logs in training set among the whole dataset is called density. The user-service QoS invocation matrix always keeps sparse in real-world applications. Thus, we conduct a series of experiments with the density varying from 0.04 to 0.32 with a step of 0.02. In order to fairly perform reliable experimental evaluation, we repeat each experiment 5 times for each density and calculate their average results.

4.2 Competing Methods

In order to show the feasibility and effectiveness of our approach, we compared with seven competing approaches, including UMEAN, IMEAN, UPCC [2], IPCC [7], WSRec [13], NRCF [9] and RACF [10].

- UMEAN. It is a user-based QoS prediction method. It averages the QoS values that the target user invoked all of the services as the predicted result.
- IMEAN. It is a service-based QoS prediction method. It averages the QoS values that all of the users invoked the target service as the prediction result.
- UPCC. It is a user-based QoS prediction method. It is required to find a set of similar users to the target user. The prediction result combines the average QoS value by UMEAN and the deviation migration based on the found similar users.
- IPCC. It is a service-based QoS prediction method. It selects the most similar services to the target service. The prediction result is composed of the average QoS value by IMEAN and the deviation migration based on the found similar services.
- WSRec. It is a QoS prediction approach by the combination of UPCC and IPCC, which utilizes a parameter to respectively weigh the importance of UPCC and IPCC.
- NRCF. It improves the accuracy of traditional collaborative filtering algorithm for QoS prediction by novel similarity computation, where it normalizes the QoS values of web services to the same range and unifies the similarity in different multi-dimensional vector spaces.
- RACF. It a QoS prediction approach based on a novel similarity computation method called ratio-based similarity (RBS). The prediction results can be calculated by the similar users or services.

4.3 Experimental Results on Accuracy of QoS Prediction

In the experiments, mean absolute error (MAE) is used as the evaluation metric that measures the average absolute deviation of the predicted QoS values to the ground truth ones. Thus, the smaller value it is, The better performance the approach has. MAE is defined as below.

$$MAE = \frac{\sum_{i=1}^{N} |r_{u,i} - \widehat{r}_{u,i}|}{N} \tag{21}$$

Where $r_{u,i}$ and $\widehat{r}_{u,i}$ represent the ground truth QoS and predicted QoS of the target user u invoking a target service i. N is the number of QoS invocation logs for test. Under different densities of QoS invocation matrix, we compare our proposed approach with the existing seven approaches on QoS prediction accuracy. The experimental results are shown in Table 2.

Table 2. MAE within the different QoS matrix densities among competing methods

Density	UNEAN	IMEAN	UPCC	IPCC	WSRec	NRCF	RACF	RECF
0.04	0.8822	0.7138	0.7107	0.7339	0.6362	0.6528	0.5838	**0.5655**
0.06	0.8766	0.7004	0.6794	0.7177	0.6395	0.5892	0.5617	**0.4837**
0.08	0.8742	0.6890	0.6294	0.7048	0.6450	0.5514	0.5157	**0.4453**
0.10	0.8752	0.6887	0.6063	0.7000	0.6394	0.5312	0.4937	**0.4332**
0.12	0.8763	0.6880	0.5869	0.6535	0.6042	0.5104	0.4674	**0.4197**
0.14	0.8747	0.6832	0.5662	0.6145	0.5689	0.4910	0.4499	**0.4099**
0.16	0.8753	0.6844	0.5485	0.5764	0.5381	0.4770	0.4370	**0.4027**
0.18	0.8755	0.6821	0.5459	0.5516	0.5173	0.4677	0.4281	**0.4028**
0.20	0.8737	0.6819	0.5379	0.5351	0.5024	0.4607	0.4208	**0.3946**
0.22	0.8737	0.6792	0.5293	0.5185	0.4887	0.4483	0.4149	**0.3905**
0.24	0.8738	0.6791	0.5181	0.5072	0.4767	0.4405	0.4104	**0.3843**
0.26	0.8751	0.6801	0.5177	0.4973	0.4725	0.4396	0.4060	**0.3859**
0.28	0.8755	0.6787	0.5096	0.4855	0.4613	0.4339	0.4032	**0.3766**
0.30	0.8752	0.6778	0.5084	0.4783	0.4571	0.4296	0.3997	**0.3789**
0.32	0.8749	0.6779	0.4980	0.4672	0.4459	0.4249	0.3991	**0.3692**

From the experimental results in Table 2, it is observed that the prediction accuracy on MAE among all of the competing methods decreases along with the increase of density in QoS invocation matrix. The reason is that the similarity degree can be more accurately calculated to improve the prediction result, as the density becomes larger and sufficient QoS invocation logs can be provided. Given a specific density, our proposed approach receives lower MAE than that of existing ones, indicating that RECF is superior to those state-of-the-art methods in terms of QoS prediction accuracy. The main reason is that we strategically eliminate all of the dissimilar services (or users) that boost the noisy of average QoS calculation in traditional collaborative filtering methods.

In order to further test the QoS prediction accuracy, we count the QoS invocation logs of test samples within multiple deviation intervals and analyze the performance among different approaches. QoS deviation represents the gap between the predicted QoS value and the true real QoS value, which is shown in the form of absolute value interval. In our experiments, the QoS deviation interval is set to 0.04. In other words, the test samples' QoS deviation intervals are divided as $[0.0, 0.04), [0.04, 0.08), [0.08, 0.12), \ldots, [1.96, 2.0)$. We calculate the number of samples in each QoS deviation interval as a percentage of the total test samples. Under the setting of QoS matrix density as 0.08 and 0.12, two groups of experiments have been performed and the results are illustrated in Figs. 2 and 3, respectively.

For each subgraph in Figs. 2 and 3, the value of each point on the polyline in lower part indicates the proportion of the number of test samples in that QoS deviation interval to the total number of test samples. The polyline in upper part is the cumulative value of the polyline in lower part till the point. It is observed that from the Figs. 2 and 3 more test samples are distributed in lower QoS deviation intervals when using our approach to predict QoS value, compared with other existing seven approaches. More specifically, making statistics on lower part in Fig. 2, we can find that 30% of test samples are distributed in $[0.0, 0.04)$ QoS deviation interval by our approach RECF, while UMEAN is 3%, IMEAN is 14%, UPCC is 13%, IPCC is 6%, WSRec is 7%, NRCF is 14% and RACF is 20%.

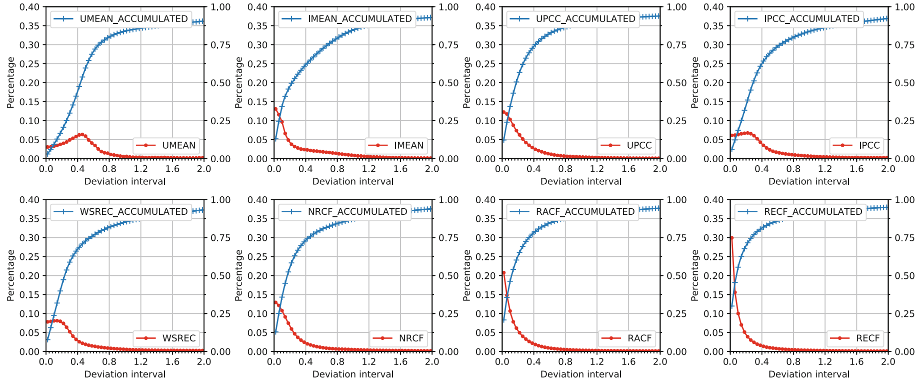

Fig. 2. The experimental results on distributions of QoS prediction among different deviation intervals (density: 0.08)

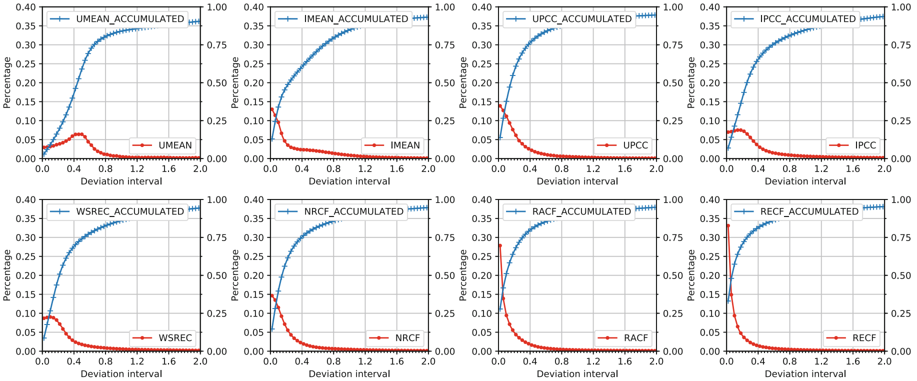

Fig. 3. The experimental results on distributions of QoS prediction among different deviation intervals (density: 0.12)

From the above experiments, we conclude that our proposed approach outperforms the existing competing ones for QoS prediction.

4.4 Impact of Parameter Tuning

In our proposed approach, there are two main parameters that affect the QoS prediction accuracy. They are the top k number of similar users (or services) in PCC similarity and neighbor similarity threshold θ in ratio-based similarity. In order to analyze the trends of QoS prediction accuracy and find out the optimal parameter value, we conduct two groups of experiments. Figure 4 illustrates that when θ keeps constant, the MAE fluctuation of our RECF is measured along with the changes of k at different densities. On the contrary, Fig. 5 shows that when k remains unchanged, the MAE variation of our RECF is measured along with the changes of θ at different densities.

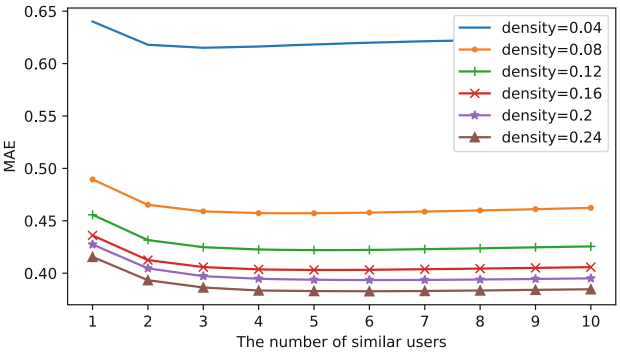

Fig. 4. The experimental results of MAE affected by the parameter (k)

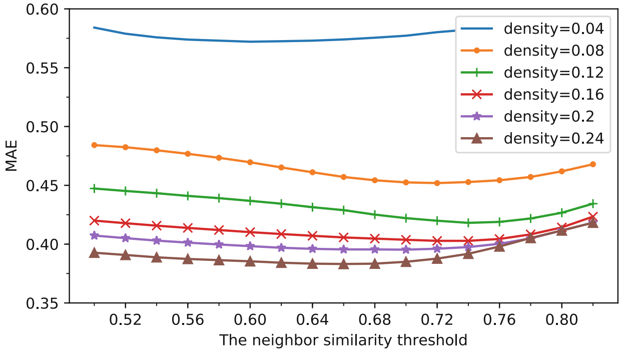

Fig. 5. The experimental results of MAE affected by the parameter (θ)

We can observe from Fig. 4, the QoS prediction accuracy achieves the best on MAE within different QoS matrix densities, when the number of similar users is set to 4. The QoS prediction accuracy is almost unchanged, as the number of similar users increases. However, the computational complexity grows with the number of increasing similar users.

From the experimental results in Fig. 5, we can find that the QoS prediction accuracy achieves the best on MAE within different QoS matrix densities when the neighbor similarity threshold in ratio-based similarity is set as 0.72.

5 Conclusion

To effectively predict missing QoS of web servies, we proposed a reinforced collaborative filtering approach that eliminates dissimilar services (or users) for improving QoS prediction accuracy. It goes through four steps: finding similar services, detecting neighbor users, average QoS calculation and deviation migration. Extensive experiments are conducted on a large-scale real-world web service QoS dataset. The results demonstrate its effectiveness competing with the existing methods. In the future work, we will apply our approach to the real-world recommender systems for modern microservices. Moreover, since collaborative filtering algorithm has high computational complexity, we will further optimize our approach making it more efficient by incremental learning.

Acknowledgements. This work was supported by National Natural Science Foundation of China (61772128), Shanghai Natural Science Foundation (18ZR1414400, 17ZR1400200) and Fundamental Research Funds for the Central Universities (16D111208).

References

1. Azadjalal, M.M., Moradi, P., Abdollahpouri, A., Jalili, M.: A trust-aware recommendation method based on Pareto dominance and confidence concepts. Knowl.-Based Syst. **116**, 130–143 (2017)
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Conference on Uncertainty in Artificial Intelligence (UAI), pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
3. Ding, S., Li, Y., Wu, D., Zhang, Y., Yang, S.: Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model. Decis. Support Syst. **107**, 103–115 (2018)
4. He, P., Zhu, J., Zheng, Z., Xu, J., Lyu, M.R.: Location-based hierarchical matrix factorization for web service recommendation. In: IEEE International Conference on Web Services (ICWS), pp. 297–304. IEEE (2014)
5. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
6. Rennie, J.D., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: International Conference on Machine Learning (ICML), pp. 713–719. ACM (2005)
7. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: International World Wide Web Conference (WWW), pp. 285–295. ACM (2001)
8. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: ACM CHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM (1995)
9. Sun, H., Zheng, Z., Chen, J., Lyu, M.R.: Personalized web service recommendation via normal recovery collaborative filtering. IEEE Trans. Serv. Comput. **6**(4), 573–579 (2013)
10. Wu, X., Cheng, B., Chen, J.: Collaborative filtering service recommendation based on a novel similarity computation method. IEEE Trans. Serv. Comput. **10**(3), 352–365 (2017)
11. Xiong, W., Wu, Z., Li, B., Gu, Q.: A learning approach to QoS prediction via multi-dimensional context. In: IEEE International Conference on Web Services (ICWS), pp. 164–171. IEEE (2017)
12. Xu, Y., Yin, J., Deng, S., Xiong, N.N., Huang, J.: Context-aware QoS prediction for web service recommendation and selection. Expert Syst. Appl. **53**, 75–86 (2016)
13. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(2), 140–152 (2011)
14. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
15. Zheng, Z., Zhang, Y., Lyu, M.R.: Distributed QoS evaluation for real-world web services. In: IEEE International Conference on Web Services (ICWS), pp. 83–90. IEEE (2010)