Predicting QoS for Cloud Services through Prefilling-Based Matrix Factorization

Chengying Mao* Zhuang Zhao

School of Software and IoT Engineering,
Jiangxi University of Finance and Economics, 330013 Nanchang, China
* *Corresponding Email: maochy@yeah.net*

***Abstract*—Quality of service (QoS) is an important indicator that users need to focus on when choosing services from the cloud center to build a service-based system. However, it is often difficult for users to collect the QoS records of all services under consideration in real circumstances. In this paper, we proposed an adjusted matrix factorization (MF) model named PFMF to predict the missing QoS values so as to make a scientific decision on service selection. Different from the existing MF models, in our PFMF method, the training is performed on the prefilled QoS matrix rather than the original sparse matrix. The prefilling of matrix is implemented by improving the PCC-based CF method. The key improvement is to consider the fluctuation degree of QoS records when finding the neighbor users or services. To validate the prediction performance of the proposed method, the comparison experiments are conducted on a widely-used large-scale dataset. The experimental results show that our PFMF method outperforms both the MF-based methods and PCC-based CF methods. In addition, it also shows good stability.**

***Index Terms*—cloud service, QoS prediction, prefilling, matrix factorization, collaborative filtering**

I. Introduction

With the rapid growth of network communication technology, cloud computing has developed into a widely-used paradigm. Accordingly, service-oriented architecture (SOA) has also become a mainstream software development model. More and more software systems are gradually changing from traditional independent development mode to the way of service composition [1]. This approach can greatly improve the efficiency of software development, but it also introduces a new challenge. That is, how to achieve satisfactory service selection [2] for different requesters from many functionally-equivalent services in some cloud centers?

For service requesters, when their functional requirements are met, non-functional requirements such as QoS will become the important reference indicators for them to choose services [3]. Unfortunately, in reality, it is often difficult for a user to have the QoS monitoring records of all services. That is to say, in the historical service invocation records, a user only has QoS values for a very limited number of services. As a result, the user may not be aware of some services that meet functional requirements. To settle this problem, the *collaborative filtering* (CF) technology in recommendation systems has been introduced to predict the missing QoS data before service selection [4].

Generally speaking, collaborative filtering prediction methods can be classified into two major categories: memory-based and model-based [5], [6]. In the memory-based CF methods, the most common way is to adopt some similarity metrics, such as Pearson correlation coefficient (PCC), to select top-*k* neighbor users (or services) from the historical records, and then QoS values of these users (or services) are taken into account for predicting the missing values of the current active user (or service) [7]. To avoid the relatively high overhead of similarity computation, Slope One [8], [9], as a lightweight CF method, has been proposed to solve the QoS prediction problem. However, the above memory-based prediction methods are usually vulnerable when the historical data set is sparse. As a popular model-based method, the *matrix factorization* (MF) [10] can alleviate the data sparsity problem with the user-service QoS matrix to some extent. Thereupon, MF and its variants have been widely used to conduct the QoS prediction and the service recommendation.

In Zheng et al.'s PCC-based QoS prediction methods [11]–[13], they proposed a strategy to tackle the data sparsity problem. That is, the empty items in the user-service QoS matrix are initially filled with estimated values. And then, the selection of similar users (or services) are performed on the filled QoS matrix rather then the original sparse matrix. Their study confirms that the filling can bring benefits to more accurately identify neighbor users (or services), thus effectively improving the prediction effect. However, at the current stage, almost all MF methods directly conduct the decomposition on the original sparse QoS matrix, and then perform the prediction by the decomposed matrices. Is the filling of the original QoS matrix beneficial for creating a more accurate MF model? To investigate this research question, we try to combine the PCC-based CF method and the MF method together to build a hybrid prediction algorithm.

In our solution, the empty items in the original user-service QoS matrix are filled with the values which are predicted by an improved PCC-based CF method. In this improved version of CF method, the stability of service invocation QoS records is taken into account. Here, the users with similar stability of service invocation records are considered to be more similar. At the same time, the local average is used to replace the global average in the original PCC formula. After the sparse QoS matrix is filled by the above method, the MF-based prediction model is performed on it. Obviously, the importance of the original values and the filled values in the MF should be different. To this end, we set a smaller weight for the filled value in the MF model. To validate the prediction effect of our prefilling-based MF method, the intensive experiments are conducted for the comparison with other state-of-the-art QoS prediction algorithms. The advantage of our method is confirmed by the experimental results.

II. Background

A. QoS Prediction for Cloud Services

When a user is ready to use some cloud services to build an application, for a given function point, there may be many functionally equivalent services in the cloud service registry for selection. Typically, the user picks the one with the best QoS performance from these candidate cloud services to build the service-based application [14]. Unfortunately, in the user's invocation records of these services, the number of the services invoked by the user is usually very limited. That is to say, the user-service QoS matrix is often a sparse matrix due to the absence of most records. In this case, for a specific user (also called an *active user*), in order to make a scientific service selection decision, it is necessary to use limited QoS information to predict the missing QoS values of the services that the active user has not invoked. This is the *QoS prediction problem*.

Here, we take the example shown in Table I to illustrate the QoS prediction problem. In this simple example, the service invocation records are about response time, and are collected from five users for five cloud services. In this record matrix, each user only has partial (two or three services) QoS values, the rest is unavailable and marked by "–". Suppose user u_5 is the current active user, and the five services are functionally equivalent cloud services that the active user is interested in. However, among these five services, only three services have the response time records. To conduct the service selection, the QoS values of the other two services (s_2 and s_4) need to be inferred. Therefore, it is necessary to refer to the QoS values of other users, especially on the services s_2 and s_4, to provide the estimated values for them.

TABLE I
An Example of User-Service Invocation Records w.r.t. Response Time (seconds)

User / Service	s_1	s_2	s_3	s_4	s_5
u_1	0.2280	–	–	0.2220	0.5270
u_2	–	0.2730	0.2510	–	0.4270
u_3	0.3660	–	0.3570	0.3580	–
u_4	–	0.2330	–	0.2190	–
u_5	0.2270	–	0.2200	–	0.3660

B. PCC-Based Collaborative Filtering

To predict the QoS for a given active user, the rational way is as follows: to estimate the missing record of the active user on a service by referring to the QoS values of the similar users on that service. Accordingly, the primary task in the QoS prediction problem is to filter out some most similar users for the given active user. The previous studies tell us that the Pearson correlation coefficient is an effective metric to measure the similarity between two users in the context of service invocation. Given a user-service QoS matrix like Table I, the similarity between two given users u and v can be calculated as follows [11], [15].

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

where $I = I_u \cap I_v$ is the subset of services which both user u and v have invoked previously, $r_{u,i}$ is a QoS value of service s_i observed by user u, and $\bar{r}_u$ and $\bar{r}_v$ represent average QoS values of different services observed by user u and v, respectively.

Based on the above similarities, for a specific service s_i, the top-k users whose QoS values on service s_i are not empty can be found to predict the missing QoS for the active user. The prediction method based on two users' PCC similarity is referred as UPCC [16]. Similarly, the prediction based on the similarity computation between two services is referred as IPCC [17]. Furthermore, the prediction method by combining the results of UPCC and IPCC is usually denoted as UIPCC [11].

C. Matrix Factorization

Matrix factorization (MF) [10] is a popular class of collaborative filtering algorithms in recent years. Its underlying idea is to represent the relationship between users and items (i.e., services in this paper) in a low-dimensional latent space. It is an approach reducing a user-item matrix into its constituent parts without significantly losing the latent features underlying the interactions between users and items. That is, the original matrix can be obtained when the two (or more) decomposed matrices multiply together. The main purpose is to perform complex matrix operations on the decomposed matrices rather than on the original one.

Let R of size $m \times n$ be the user-item matrix, and d is the number of latent features that determine how a user rates an item. In general, $d \ll \min(m,n)$. The task of MF is find two low-dimensional matrices P (a $m \times d$ matrix) and Q (a $n \times d$ matrix) such that their product approximates R:

$$R \approx P \times Q^T = \hat{R} \quad (2)$$

Nowadays, there are quite a few ways to obtain P and Q with some values such that the *difference* (also called the *error*) between their product and the original R is minimum. In the service QoS prediction problem, the user-service QoS matrix is sparse, so the MF model is trained by the available data in the matrix. When a stable MF model is obtained, it is used to predict missing QoS records in the matrix.

Non-negative matrix factorization (NMF) [18] is a typical MF with a non-negativity constraint, that is, all three matrices (R, P, and Q) have no negative elements. The existing work has shown that the NMF model can produce relatively favourable results for the QoS prediction problem. As a consequence, in this study, we use NMF as the base model for the further improvement by prefilling the empty records in the user-service QoS matrix.

III. Prefilling-Based Matrix Factorization

A. The Overall Framework

The technology roadmap of prefilling-based MF prediction for the QoSs of cloud services is illustrated in Fig. 1. The whole framework consists of the following two stages: The first stage is to prefill the original sparse user-service matrix, the second stage is to train an adjusted NMF model on the prefilled matrix, and then conduct the QoS prediction based on the trained model.

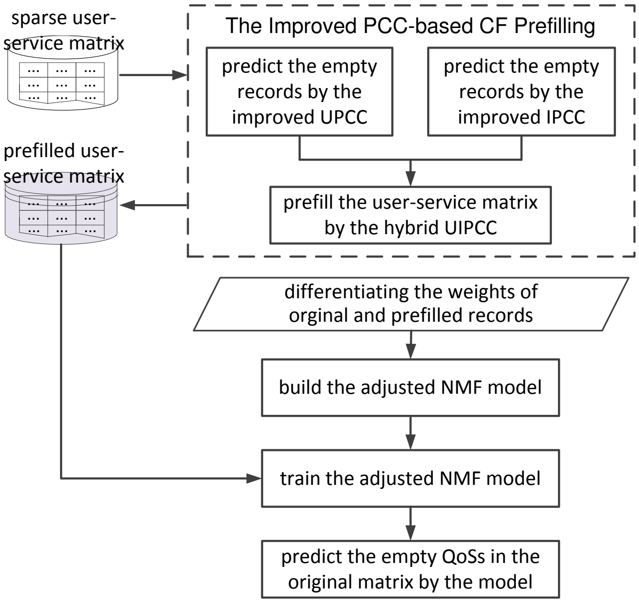

sparse user-service matrix
The Improved PCC-based CF Prefilling
predict the empty records by the improved UPCC
predict the empty records by the improved IPCC
prefilled user-service matrix
prefill the user-service matrix by the hybrid UIPCC
differentiating the weights of orginal and prefilled records
build the adjusted NMF model
train the adjusted NMF model
predict the empty QoSs in the original matrix by the model

Fig. 1. The technology roadmap of Prefilling-Based MF for QoS prediction

At the first stage, the prefilling is performed through improving the PCC-based CF method. Here, the similarity for finding the top-k users (or services) is adapting by considering the consistency of changes in the QoS records of two users (services). Meanwhile, similar to the treatment in [19], the global QoS average of neighbor users is replaced by the local average in the prediction step of PCC-based CF. At the second stage, the original QoS records are assigned to a high weight in the NMF model. By contrast, the prefilled QoSs can only be used as a preliminary reference, so the corresponding weight can only be set to a smaller value, such as 0.1. After the adjusted NMF model is built, it is used for predicting the missing QoS values in the original user-service matrix.

B. The Prefilling by the Improved PCC-based CF

When computing the similarity between two users or services, the PCC-based CF approach focuses on the degree of closeness of the QoS values between two records, and the Slope One method takes into account the relative change trend of QoS values. In this study, the fluctuation degree of QoS values is also considered as an indicator of the similarity between two users or services. That is, the users (services) that are stable (or unstable) in terms of QoS are likely to be similar.

Based on the above instinctive judgment, we improve the well-known PCC-based similarity calculation method by incorporating the fluctuation indicator. In this study, we use the basic standard deviation to account for the volatility of QoS values and adjust the similarity calculation between two users (services) as follows.

$$Sim'(u,v) = \frac{\min(\sigma_u, \sigma_v)}{\max(\sigma_u, \sigma_v)} \cdot Sim(u,v) \tag{3}$$

σ_u and σ_v are the standard deviations of QoS values of the services invoked by user u and v, respectively.

Based on the above similarity calculation, the similar users (or neighbors) of a given user can be identified. To predict a missing QoS value $r_{u,i}$ in the user-service matrix, the set $T(u)$ of top-k similar users whose QoSs on service s_i are not empty can be selected by the their similarities to user u. Then, the neighbors of user u, $N(u)$, can be found by the following equation:

$$N(u) = \{v | v \in T(u), Sim'(u,v) > 0, u \neq v\} \tag{4}$$

Subsequently, we can get the prefilling QoS value for $r_{u,i}$ from the user perspective as follows.

$$\begin{aligned} pf_{\text{user}}(r_{u,i}) =& \overline{u}_{\text{local}}(u,i) + \\ & \frac{\sum_{v \in N(u)} Sim'(u,v) \cdot (r_{v,i} - \overline{u}_{\text{local}}(v,i))}{\sum_{v \in N(u)} Sim'(u,v)} \end{aligned} \tag{5}$$

where $\overline{u}_{\text{local}}(u,i)$ means the weighted average of the QoS values of top-k neighbor services of s_i with respect to user u, similarly, $\overline{u}_{\text{local}}(v,i)$ is the local QoS average about user v.

In the similar way, we can also compute the prefilling QoS value for $r_{u,i}$ from the service perspective, and denote it as $pf_{\text{service}}(r_{u,i})$. Thus, the final prefilling value for the empty element $r_{u,i}$ in the original user-service QoS matrix R is calculated by aggregating the above $pf_{\text{user}}(r_{u,i})$ and $pf_{\text{service}}(r_{u,i})$, that is,

$$pf(r_{u,i}) = \lambda \cdot pf_{\text{user}}(r_{u,i}) + (1-\lambda) \cdot pf_{\text{service}}(r_{u,i}) \tag{6}$$

Here, a parameter λ is employed to determine how much does the final prefilling value rely on user-based CF using PCC or item-based CF using PCC. In our experiments, λ is set to 0.1. Based on the above treatment, the prefilled QoS matrix can be obtained and is denoted as R_{pf}.

C. The Adjusted NMF Model

The existing MF-based QoS prediction methods usually train the MF model by using the original sparse QoS matrix. In this study, we adopt the non-negative MF (NMF) [18] as a representative to construct a *prefilling-based MF* method (PFMF for short). Different from the existing work, our solution uses the prefilled user-service matrix to train the NMF model. In the prefilled matrix R_{pf}, the empty element is estimated by the improved PCC-based CF shown in the last subsection.

Obviously, the elements in R_{pf} can be classified into two categories: the original QoSs and the prefilled QoSs. While training the NMF model by the QoS values, the above two types of elements should play different roles. That is to say, the original QoS value should be regarded as the key reference object, and the prefilled value can only be used as a moderate reference. Based on this general idea, the basic NMF model is adjusted in the following way.

$$\begin{aligned}\min_{P,Q} \mathcal{L}(R_{\text{pf}}, P, Q) =& \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{\text{pf}}(i,j) - P_i Q_j^T)^2 \\ &+ \frac{\lambda_P}{2}\|P\|_F^2 + \frac{\lambda_Q}{2}\|Q\|_F^2, \\ s.t.\ P \geq 0, Q \geq 0 & \qquad (7)\end{aligned}$$

where I is an indicator matrix and I_{ij} is the element of the i-th row and the j-th column in it. $R_{\text{pf}}(i,j)$ represents the element on the i-th row and the j-th column in the matrix R_{pf}. $\|\cdot\|_F$ is the Frobenius norm, λ_P and λ_Q are two regularization coefficients. In this model, the multiplicative update rule [20] is also used to learn the low-dimensional latent matrix P and Q.

In the basic NMF, the model is trained by the original sparse QoS matrix R, so the indicator I is a 0-1 matrix. When the element r_{ij} in R is not empty, $I_{ij} = 1$. Otherwise, $I_{ij} = 0$. In our adjusted NMF method, the model is trained by the prefilled QoS matrix. Obviously, when the r_{ij} is an element that existed before, the corresponding I_{ij} is naturally still 1. As mentioned earlier, the prefilled elements in R_{pf} also have reference value for model training, but their value is obviously lower than that of existing elements. Therefore, the reference value of this type of elements should be between 0 and 1. Thus, we can summarize the possible values of I_{ij} as follows.

$$I_{ij} = \begin{cases} \theta, & \text{if } r_{ij} \in R_{\text{pf}} \text{ is a prefilled element} \\ 1, & \text{otherwise} \end{cases} \qquad (8)$$

Here, $\theta \in (0,1)$. In our experiments, we set it to 0.1.

In the adjusted NMF, the latent matrices P and Q are initialized by random values, and then are trained by R_{pf} through the optimization function shown in Equations (7) and (8). Finally, the P and Q with the minimum loss (or error) are treated as the output. Accordingly, the empty elements in the original user-service QoS matrix are predicted by

$$\hat{r_{ij}} = P_i \times Q_j^T, \text{ if } r_{ij} \in R \text{ is empty.} \qquad (9)$$

Subsequently, users can use the predicted QoS to make decision on service selection.

IV. Experimental Evaluation

A. Experimental Setup

To validate the effectiveness of our proposed prefilling-based MF method (PFMF), a widely-used real-world dataset named WS-Dream [21] is used for experimental analysis. This dataset consists of a total of 1,974,675 QoS records of 5,825 Web services, collected from the invocations of 339 users distributed across the world. In the records, the QoS is reflected by the following two factors: response time (RT) and throughput (TP).

To simulate various prediction scenarios with different data sparsity, the records in the original dataset are randomly removed to prepare some user-service matrices with 5%, 10%, 15% and 20% densities. The removed records are treated as the actual QoS values for evaluating the prediction accuracy of different methods.

In the experiments, the prediction accuracy is measured by Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$MAE = \frac{\sum_{i,j} |rl_{ij} - \hat{r_{ij}}|}{N} \qquad (10)$$

$$RMSE = \sqrt{\frac{\sum_{i,j} (rl_{ij} - \hat{r_{ij}})^2}{N}} \qquad (11)$$

where rl_{ij} denotes the actual QoS values of service j observed by user i, $\hat{r_{ij}}$ is the predicted QoS value, and N represents the number of predicted values.

Besides the basic Slope One [8], UPCC [16], IPCC [17], and UIPCC [11] methods, some other MF methods are also taken into account for comparison analysis, such as BiasedMF [22], [23], EMF (extended MF) [24], PMF (probabilistic MF) [25], and NMF [20]. Meanwhile, the parameters in our algorithm are set as below: top-k=5, λ=0.1, λ_P=λ_Q=30 for attribute RT (=800 for attribute TP), and θ=0.1. The parameters of the related MF methods adopt the recommended configuration in Zheng et al.'s work, please refer to [21]. The implementation of existing methods directly adopts the open source code in [21], and our PFMF algorithm is implemented in C++ and further wrapped by Python. The experiments are conducted on the operating system of Windows 10 64-bit.

B. Experimental Results

The results of the comparative experiments are shown in Table II. For the attribute RT, the performance of Slope One method is the worst one. Among the three PCC-based CF methods, the UIPCC achieves the best QoS prediction performance. Generally speaking, MF-based prediction methods can generate better prediction results than the PCC-based methods. While considering the four existing MF-based methods, for the metric MAE, the following dominant relation can be found: NMF>EMF>PMF>BiasedMF. Specifically, when the matrix density is low (i.e., 5%), the advantage of NMF algorithm is

TABLE II
Performance Comparison between QoS Prediction Methods in Different Matrix Densities

Attributes	**Methods**	**Density=5%**		**Density=10%**		**Density=15%**		**Density=20%**	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
RT	Slope One	0.6461	1.4102	0.6254	1.3790	0.6199	1.3693	0.6171	1.3648
	UPCC	0.6369	1.3854	0.5541	1.3120	0.5138	1.2609	0.4859	1.2210
	IPCC	0.6363	1.4068	0.5950	1.3465	0.5113	1.2645	0.4569	1.2095
	UIPCC	0.6226	1.3736	0.5468	1.3007	0.4988	1.2417	0.4669	1.1982
	BiasedMF	0.5896	1.3914	0.5136	1.2655	0.4782	1.2113	0.4584	1.1819
	PMF	0.5697	1.5442	0.4870	1.3186	0.4524	1.2217	0.4307	1.1702
	EMF	0.5589	1.4967	0.4841	1.2928	0.4476	1.2059	0.4274	1.1603
	NMF	0.5462	1.4800	0.4785	1.2865	0.4468	1.2038	0.4274	1.1621
	PFMF	**0.5072**	**1.3225**	**0.4594**	**1.2188**	**0.4330**	**1.1634**	**0.4165**	**1.1322**
TP	Slope One	28.1322	64.7711	27.2791	62.9530	27.0546	62.4212	26.9626	62.1742
	UPCC	27.3082	61.3298	22.7137	54.6768	20.5320	51.0702	19.3004	48.8270
	IPCC	27.0618	63.3529	26.2284	60.4854	25.5953	57.7503	23.9987	54.8184
	UIPCC	26.5052	60.9971	22.4215	54.6319	20.3062	50.6518	18.9339	47.9398
	BiasedMF	21.8181	56.9015	17.8142	48.1414	15.8845	44.1530	14.8684	42.0155
	PMF	19.1349	58.1566	15.9896	48.1060	14.6313	43.9197	13.9480	41.7430
	EMF	20.1498	59.0542	15.7990	48.3354	14.2985	43.9953	13.5291	41.6244
	NMF	18.9101	57.7657	15.6005	47.9164	14.2728	43.8848	13.5458	41.6984
	PFMF	**18.8339**	**53.1834**	**15.1498**	**45.5339**	**13.7204**	**42.1970**	**13.1100**	**40.4608**

obvious. When the matrix density is relatively high (i.e., 20%), the performance of NMF is comparable to that of EMF. For the metric RMSE, in the case of low matrix density, the prediction performance of BiasedMF is the best one among the four MF-based methods. However, in the case of high matrix density (i.e., 20%), BiasedMF becomes the worst one. Compared with the four existing MF-based methods, our PFMF method is always better than them in all cases of matrix densities for both two performance metrics.

With regard to the experimental results on attribute TP, the relation between all QoS prediction methods is similar to the case on attribute RT. That is, our PFMF method achieves the best prediction results in all cases of matrix density. For the metric MAE, the advantage of PFMF method is relatively stable, and its trend of change with matrix density is not obvious. For the metric RMSE, PFMF method always has an obvious advantage, and the advantage is more significant at low matrix density. While considering other QoS prediction methods, the MF-based methods are better than the PCC-based CF methods, and the Slope One methods still has the worst prediction performance.

C. Parameter Impact Analysis

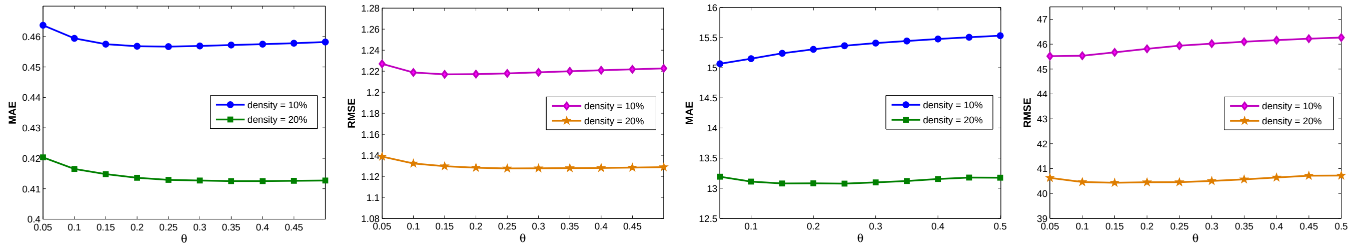
It is not difficult to see that θ is an important parameter in our PFMF model, which controls the reference degree of prefilled values. In the experiment, the value of θ was varied from 0.05 to 0.5 to observe its influence on the prediction performance of the model. For the QoS attribute RT, the trend of MAE and RMSE caused by the change of θ is shown in Figures 2(a) and 2(b), respectively. It can be seen from the experimental results that when θ changes from 0.05 to 0.2, MAE and RMSE decrease slightly in both densities (i.e., 10% and 20%). When θ exceeds 0.2, the prediction performance of the PFMF method remains substantially stable. Therefore, for this QoS attribute, the recommended value of θ is in the interval [0.2, 0.5].

The experimental results about QoS attribute TP are shown in Figures 2(c) and 2(d). When the density of user-service matrix is 10%, both MAE and RMSE increase very slightly with the increase of θ. In the case of density=20%, when θ increases from 0.05 to 0.15, the MAE and RMSE have a very small decrease; when θ is from 0.3 to 0.5, the MAE and RMSE also increase very slightly. Thus, for the attribute TP, the recommended value of θ is in the interval [0.1, 0.3].

More importantly, from the above analysis, it can be seen that the change of parameter θ does not have a significant impact on the prediction performance of our PFMF method. In other words, PFMF is a stable QoS prediction method.

V. Concluding Remarks

In the service computing paradigm, service selection is one of the key aspects of building a service-based system. In addition to the functionality of service, non-functional requirements such as QoS are also important concerns. In reality, service QoS records of users are often incomplete. Therefore, in order to carry out scientific service selection decisions, it is especially necessary to predict the missing QoS values. In this study, we adjust the existing MF-based



(a) RT, MAE

(b) RT, RMSE

(c) TP, MAE

(d) TP, RMSE

Fig. 2. The Impact of Parameter θ

QoS prediction model through prefilling the empty records in user-service QoS matrix. The model training is based on the prefilled matrix rather than the original sparse QoS matrix. In this way, model training can refer to more information, so the resulting model is more accurate. When prefilling the missing values, the traditional PCC-based CF method is improved by considering the fluctuation degree of QoS records.

The effectiveness of our proposed PFMF method is validated by the real-world QoS dataset of Web services. The experimental results confirm that the prediction performance of PFMF method are better than the existing MF-based methods and PCC-based CF methods. At the same time, the performance of the model has less impact on the change of parameters. Of course, PFMF method can still be further optimized. For example, the reference weight of each data item in the prefilled matrix can be refined in the follow-up work.

Acknowledgments

This work was supported in part by the NSFC (Grant Nos. 61762040 and 61462030), the Natural Science Foundation of the Jiangxi Province (Grant Nos. 20162BCB23036 and 20171ACB21031), and the Jiangxi Social Science Research Project (Grant No. TQ-2015-202).

References

[1] A. Bouguettaya, M. Singh, M. Huhns, Q. Z. Sheng, H. Dong, and et al., “A service computing manifesto: The next 10 years,” *Communications of the ACM*, vol. 60, no. 4, pp. 64–72, 2017.

[2] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, “Cloud service selection: State-of-the-art and future research directions,” *Journal of Network and Computer Applications*, vol. 45, no. C, pp. 134–150, 2014.

[3] F. Ishikawa, “QoS-based service selection,” in *Web Services Foundations*, A. Bouguettaya, Q. Z. Sheng, and F. Daniel, Eds. Springer-Verlag New York, 2014, pp. 375–397.

[4] Y. Zhang and M. R. Lyu, *QoS Prediction in Cloud and Service Computing: Approaches and Applications*. Springer Singapore, 2017.

[5] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[6] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45, 2014.

[7] M. Deshpande and G. Karypis, “Item-based top-N recommendation algorithms,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.

[8] D. Lemire and A. Maclachlan, “Slope One predictors for online rating-based collaborative filtering,” in *Proc. of the 2005 SIAM International Data Mining Conference (SDM’05)*, April 2004, pp. 1–5.

[9] C. Mao and J. Chen, “QoS prediction for Web services based on similarity-aware Slope One collaborative filtering,” *Informatica*, vol. 37, no. 2, pp. 139–148, 2013.

[10] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[11] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “QoS-aware Web service recommendation by collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[12] Z. Zheng and M. R. Lyu, *QoS Management of Web Services*. Springer-Verlag Berlin Heidelberg, 2013.

[13] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, “Predicting quality of service for selection by neighborhood-based collaborative filtering,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 428–439, 2013.

[14] C. Mao, J. Chen, D. Towey, J. Chen, and X. Xie, “Search-based QoS ranking prediction for Web services in cloud environments,” *Future Generation Computer Systems*, vol. 50, pp. 111–126, 2015.

[15] J. L. Rodgers and W. A. Nicewander, “Thirteen ways to look at the correlation coefficient,” *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[16] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI’98)*, July 1998, pp. 43–52.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. of the 10th international conference on World Wide Web (WWW’01)*, April 2001, pp. 285–295.

[18] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[19] W. Wang, G. Zhang, and J. Lu, “Collaborative filtering with entropy-driven user similarity in recommender systems,” *International Journal of Intelligent Systems*, vol. 30, pp. 854–870, 2015.

[20] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proc. of the 13th International Conference on Neural Information Processing Systems (NIPS’00)*, 2010, pp. 535–541.

[21] J. Zhu, Z. Zheng, P. He, Y. Xiong, and Y. Lu, “WS-Dream: A package of open source-code and datasets to benchmark QoS prediction approaches of Web services,” Available at: https://github.com/wsdream.

[22] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’08)*, August 2008, pp. 426–434.

[23] D. Yu, Y. Liu, Y. Xu, and Y. Yin, “Personalized QoS prediction for Web services using latent factor models,” in *Proc. of the 2014 IEEE International Conference on Services Computing (SCC’14)*, June 2014, pp. 107–114.

[24] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, “An extended matrix factorization approach for QoS prediction in service selection,” in *Proc. of the IEEE 9th International Conference on Services Computing (SCC’12)*, June 2012, pp. 162–169.

[25] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *Proc. of the 20th International Conference on Neural Information Processing Systems (NIPS’07)*, December 2007, pp. 1257–1264.