A Cluster Feature Based Approach for QoS Prediction in Web Service Recommendation

Shulong Chen, Yuxing Peng, Haibo Mi, Changjian Wang, Zhen Huang
Science and Technology on Parallel and Distributed Laboratory
College of Computer, National University of Defense Technology
Changsha, 410073, China
chenshulong@yeah.net

***Abstract*—With the growing popularity of Service-Oriented-Computing (SOC) architecture, the number of Web services on the internet is increasing rapidly. When faced with a large number of candidate services with similar functionalities, personalized Web service recommendation is becoming an important issue. Quality-of-Service (QoS) is usually used to characterize the non-functional properties of Web services. Thus accurate QoS prediction is an important step in the service recommendation. In this paper, we propose a Cluster Feature based Latent Factor Model (CFLFM) for QoS prediction. First, we cluster users and services into several groups based on history records, respectively. We assume that users or services in the same cluster share some latent features. By incorporating this kind of information, we design an integrated latent factor model. Finally, we conduct comprehensive experiments on a real-world Web service dataset. The experimental results show that our approach can achieve higher QoS prediction accuracy than other competing approaches.**

***Keywords*—Web Service, QoS Prediction, Clustering, Latent Factor Model**

I. Introduction

Web service is a software system that supports machine-to-machine interaction through a network [1]. With the rapid growth of Web services on the internet, how to effectively select the web services that are most suitable for users is becoming an urgent problem. However, traditional Web service discovery approaches can only find the candidates that fulfill users' functional requirements, but neglect the non-functional characteristics. In the fact, such non-functional characteristics (such as response time, throughput, reliability) also play an important role in Web service selection.

QoS is a common set of measurements to describe the service' non-functional performance [2]. As more and more Web services with same or similar functions are available on the internet, QoS-aware service recommendation is becoming a hot spot of research recently. In order to make QoS-aware recommendations, we need to know the QoS values of candidate services. However, it's not an easy thing. Due to heterogeneous environments and various communication links, different users may perceive different QoS values when invoking even the same Web service. In addition, the QoS values obtained from service providers or third-party organizations are not unauthentic, because QoS evaluations should be conducted at the user-side. Directly conducting real world Web services invocations is money-spending, resource-consuming and time-wasting. One feasible way is to predict the missing QoS values using the known records. Therefore, how to accurately predict the missing QoS values is becoming a major challenge in Web service recommendation.

There are many studies on QoS-aware recommendation, among which Collaborative Filtering (CF) based approaches achieve state-of-art performance. As an effective tool to alleviate the information overload problem, CF approaches are wildly used in commercial recommender systems. Existing CF approaches can be divided into three categories: neighborhood-based, model-based and hybrid approaches. Neighborhood-based approaches use local neighbor information to predict the missing ratings. While model-based approaches first train a model using the known ratings and then use the model to predict missing ratings. Latent Factor Model (LFM) is a popular model-based approach. LFM assumes the user-service matrix is global low rank and use this global information to make predictions. Due to the inherent sparsity of QoS data matrix, only considering local information or global information cannot obtain high accuracy. Hybrid approaches which combine the previous two kinds of approaches can usually achieve higher precision.

In this paper, we propose a hybrid model by merging local and global information. First, we cluster users and services into several groups by Fuzzy C-Means (FCM) algorithm, respectively. We assume that users or services in each group share some latent features. Then we design an integrated LFM model by incorporating this kind of latent local information. Finally, comprehensive experiments on a real-world Web service dataset show that our approach can achieve higher QoS prediction accuracy than other competing approaches.

The contributions of the paper are summarized as follows:

1) We propose a fuzzy clustering based approach to get the local features of users and services. We use a factor vector and a bias to represent this kind of local information in each fuzzy cluster.
2) By incorporating the local information of each cluster, we design an integrated latent factor model called CFLFM.
3) We conduct extensive experiments on a large-scale Web service dataset.

The rest of the paper is organized in this way: Section II introduces the related work in QoS-aware Web service recommendation. Section III demonstrates our QoS prediction

0-7695-6394-5/18/$31.00 ©2018 IEEE
DOI 10.1109/SOSE.2018.00041

approach in detail. The experiments are presented in Section IV. Finally, Section V concludes the paper.

II. Related Work

With the rapid growth of Web services available, the problem of QoS-aware service recommendation is becoming more and more important. Existing CF based QoS-aware service recommendation approaches can be divided into four categories: neighborhood-based approaches, model-based approaches, hybrid approaches and context-aware approaches.

Neighborhood-based approaches can be future divided into user-based and service-based approaches. Zheng Z et al. propose UIPCC approaches by combining user-based and service-based approaches [3]. Jian Wu et al. propose a neighborhood-based approach for QoS prediction. By using adjusted-cosine-based similarity measure and similarity fusion, their method outperforms traditional neighborhood-based methods [4]. Zheng Z et al. present a new similarity measure for Web service similarity computation and propose a novel neighborhood-based approach, called NRCF [5]. Neighborhood-based approaches is simple and intuitive. But when the QoS data is sparse, two users may have few or even no common invoked services, which will lead to inaccuracy in similarity calculation. In that case, these methods always perform poorly. Moreover, the time complexity of similarity calculation is quadratic to the user or service number, which makes neighborhood-based methods not effective to extend to very large dataset.

Model-based approaches first train a global model based on the known QoS data and then use it to predict the missing values. Zheng Z et al. adopt a PMF based approach for personalized reliability prediction [6]. Jieming Zhu et al. present a clustering-based approach for QoS prediction [7]. Based on a probabilistic latent model, Bobaker Mohamed et al. describe a novel approach PLMwsp to predict QoS values [8].

The hybrid approaches combine the neighborhood-based approaches and model-base approaches. Yilei Zhang et al. propose a hybrid approach called CloudPred. It first learns the characteristics of users by non-negative matrix factorization (NMF) and then predicts QoS values from similar users [9]. Wei Lo et al. propose a hybrid model, by incorporating two novel relation regularization terms into MF model [10]. Zheng Z et al. propose a neighborhood-integrated MF model (NIMF), by fusing the neighborhood-based and the model-based approaches to achieve better performance [11]. Kesheng Qi et al. propose a matrix factorization method, integrating both user network neighborhood information and service neighborhood [12]. Although these approaches make use of global information and local information, their performance is not satisfactory. Furthermore, the computational complexity of these approaches is relatively high.

Context-aware based QoS approaches assume that the QoS value is highly related to the invocation context (e.g. location, time). Zhang Y et al. propose a method to integrate the information of services' latent neighbors through sharing the latent feature vector of provider and its country [13]. [14] presents a hybrid-based CF method, by incorporating the geographical information. [15] presents a collaborative QoS prediction approach with location-based regularization (LBR). [16] [17] [18] integrate time information into QoS prediction model to make high-quality service recommendation. By considering these side information, context-aware approaches can usually achieve better prediction accuracy. But this kind of context information is not always available. Compared with these context-aware approaches, our approach does not require any side information.

III. Our Approach

In this section, we first give an overview of our approach. in Section A. In section B, we divide users and services into several clusters, respectively. Then section C introduces our CFLFM model in detail. Finally, the computational complexity analysis is presented in section D.

A. System Overview

Most QoS-aware service recommendation approaches only make use of local information (such as neighborhood-based approaches) or global information (such as LFM approaches). Few approaches consider these two kinds of information at the same time. From this point of view, we propose a hybrid recommendation approach, called CFLFM. The core idea of CFLFM is that: we separately cluster users and services into several groups and assume that users or services in the same group share some latent features, then we incorporate this kind of local information into latent factor model. As shown in Fig.1, the architecture of our approach is mainly composed of three parts: PMF, FCM and CFLFM. Next, we give a detail introduction to each part in following sections. All notations used in this paper are summarized in Table I.

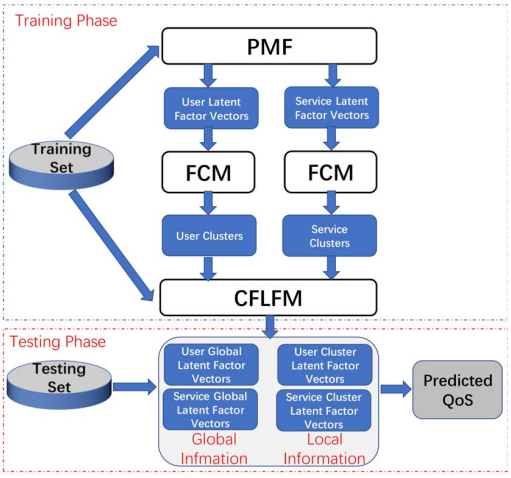


Fig. 1: System Overview of Our Approach

TABLE I: Notations used in this paper

Symbol	Description
u_i	a user, $i \in \{1, \ldots, M\}$ is used to index a user
s_j	a service, $j \in \{1, \ldots, N\}$ is used to index a service
$R_{M\times N}$	user-service matrix, $R_{i,j}$ represents the QoS value from user u_i on service s_j
$U_{D\times M}$	user latent factor matrix in PMF model, U_i represents the latent factor vector of u_i
$V_{D\times N}$	service latent factor matrix in PMF model, V_j represents the latent factor vector of s_j
$P_{M\times D}$	user latent factor matrix, P_i represents the latent factor vector of u_i
$Q_{N\times D}$	service latent factor matrix, Q_j represents the latent factor vector of s_j
$W_{M\times C}$	fuzzy C-partition of $U_{D\times M}$
$H_{M\times C}$	fuzzy C-partition of $V_{D\times N}$
$X_{D\times C}$	user cluster latent factor matrix, X_k represents the latent factor vector of k-th user cluster
$Y_{D\times C}$	service cluster latent factor matrix, Y_k represents the latent factor vector of k-th service cluster
b, d, g, h	the bias terms of users, services or clusters
Ω	$(u_i, s_j) \in \Omega$ indicates the QoS value from user u_i on service s_j is observed
λ	regularization parameter

B. Users and Services Clustering

We make an assumption that if some users (services) have similar history QoS records, they may share some latent features. Then we divide users (services) into several clusters according to their history record. Based on our assumption, users (services) in the same cluster should share some latent features. That features of each user (service) cluster represent some kind of local information, which can be integrated into LFM model.

Because the QoS data matrix is very sparse and has lots of missing values. Directly using Pearson Correlation Coefficient (PCC) or cosine similarity to cluster users or services cannot achieve satisfactory performance. We therefore first factorize R into the product of two low rank matrices $R \approx U^T V$, by minimizing the following loss function [19]:

$$\min_{U,V} \sum_{(u_i,s_j)\in\Omega} (R_{i,j} - U_i^T V_j)^2 + \lambda_U \|U\|_F^2 + \lambda_V \|V\|_F^2 \tag{1}$$

Where $\|\cdot\|_F^2$ indicate the Frobenius norm of a matrix, $\lambda_U \|U\|_F^2$ and $\lambda_V \|V\|_F^2$ are regularization terms to avoid overfitting.

Given the user latent factor matrix U and service latent factor matrix V, we can separately cluster users and services into several groups, according to their latent factor vectors. We choose FCM algorithm to cluster users and services, because it's very effective and flexible [20]. Different from K-means and other hard clustering algorithms, FCM employs the membership degree to measure the degree of data points belonging to clusters.

If we divide users into C clusters, the object function of FCM is defined as follows:

$$\mathcal{J}_u = \sum_{k=1}^{C} \sum_{i=1}^{M} W_{i,k}^m \|U_i - e_k^u\|_F^2 \tag{2}$$

$$\sum_{k=1}^{C} W_{i,k} = 1, \ \ i = 1, 2, \ldots, M \tag{3}$$

Where $m \in [1, +\infty)$ is the weighting exponent, $e^u = \{e_1^u, e_2^u, \ldots, e_C^u\}$ are the centers of user clusters.

Similarly, we can divide services into C clusters by FCM algorithm. The object function is defined as follows:

$$\mathcal{J}_s = \sum_{k=1}^{C} \sum_{j=1}^{N} H_{j,k}^m \|V_j - e_k^s\|_F^2 \tag{4}$$

$$\sum_{k=1}^{C} H_{j,k} = 1, \ \ j = 1, 2, \ldots, N \tag{5}$$

Where $e^s = \{e_1^s, e_2^s, \ldots, e_C^s\}$ are the centers of service clusters. FCM algorithm performs an iterative process to optimize the object functions $\mathcal{J}_u$ and $\mathcal{J}_s$.

C. Integrated Latent Factor Model

After the FCM clustering procedure is completed, we have divided users (services) into C fuzzy clusters. To represent the local information in each user (service) cluster, we associate k-th user (service) cluster with a latent factor vector $X_k \in R^D$ ($Y_k \in R^D$) and a bias $g_k \in R$ ($h_k \in R$). Next, we design an integrated latent factor model by incorporating the local information of each user (service) cluster.

If a user has a higher membership degree to a fuzzy cluster, he is more likely to share the common latent features in that cluster. So, we weigh the cluster latent factor vector by user's membership degree. From the view of local information, the latent factor vector and bias of user u_i can be defined as follows:

$$P_i^l = \sum_{k=1}^{C} W_{i,k} X_k \tag{6}$$

$$b_i^l = \sum_{k=1}^{C} W_{i,k} g_k \tag{7}$$

Similarly, if we only consider the local information from service clusters, the latent factor vector and bias of service s_j

can be defined as follows:

$$Q_j^l = \sum_{k=1}^{C} H_{j,k} Y_k \tag{8}$$

$$d_j^l = \sum_{k=1}^{C} H_{j,k} h_k \tag{9}$$

Only using local information will lead to biased prediction, because it neglects the global information. Similar to [13] [21], we set a parameter α to balance the local and global information of users and set a parameter β to balance the local and global information of services. Then, we can combine these two kinds of information in a linear way.

$$P_i = (1-\alpha) P_i^g + \alpha P_i^l \tag{10}$$

$$b_i = (1-\alpha) b_i^g + \alpha b_i^l \tag{11}$$

Where P_i^g and b_i^g represent the global latent factor vector and global bias of user u_i. P_i and b_i represent the combined latent factor vector and combined bias of user u_i.

$$Q_j = (1-\beta) Q_j^g + \beta Q_j^l \tag{12}$$

$$d_j = (1-\beta) d_j^g + \beta d_j^l \tag{13}$$

Where Q_j^g and d_j^g represent the global latent factor vector and global bias of service s_j. Q_j and d_j represent the combined latent factor vector and combined bias of service s_j.

Then the missing QoS value $R_{i,j}$ can be predicted as follows:

$$\hat{R}_{i,j} = \mu + b_i + d_j + P_i Q_j^T \tag{14}$$

(14) is equivalent to the following formula:

$$\begin{aligned} \hat{R}_{i,j} = {} & [(1-\alpha) P_i^g + \alpha \sum_{k=1}^{C} W_{i,k} X_k][(1-\beta) Q_j^g + \beta \sum_{k=1}^{C} H_{j,k} Y_k] \\ & + (1-\alpha) b_i^g + \alpha \sum_{k=1}^{C} W_{i,k} g_k + (1-\beta) d_j^g + \beta \sum_{k=1}^{C} H_{j,k} h_k \\ & + \mu \end{aligned} \tag{15}$$

Where μ indicate the overall average QoS value. We can find the prediction function (15) has a similar form with SVD-Feature framework [22]. The objective function of CFLFM is defined as follows:

$$\begin{aligned} \mathcal{L} = {} & \sum_{(u_i, s_j) \in \Omega} (R_{i,j} - \hat{R}_{i,j})^2 \\ & + \sum_{(u_i, s_j) \in \Omega} \lambda_1 (\|P_i^g\|_F^2 + \|Q_j^g\|_F^2) + \lambda_2 (\|X\|_F^2 + \|Y\|_F^2) \\ & + \sum_{(u_i, s_j) \in \Omega} \lambda_3 ((b_i^g)^2 + (d_j^g)^2) + \lambda_4 (\sum_{k=1}^{C} g_k^2 + \sum_{k=1}^{C} h_k^2) \end{aligned} \tag{16}$$

where λ_1, λ_2, λ_3, λ_4 are regularization terms to avoid overfitting.

We apply stochastic gradient descent algorithm to train our model. For each training sample $(u_i, s_j, R_{i,j})$, the prediction error is $e_{i,j} = R_{i,j} - \hat{R}_{i,j}$. The model parameters are updated in the following way:

$$P_i^g \leftarrow P_i^g + \eta[(1-\alpha) e_{i,j} ((1-\beta) Q_j^g + \beta \sum_{k=1}^{C} H_{j,k} Y_k) - \lambda_1 P_i^g] \tag{17}$$

$$Q_j^g \leftarrow Q_j^g + \eta[(1-\beta) e_{i,j} ((1-\alpha) P_i^g + \alpha \sum_{k=1}^{C} W_{i,k} X_k) - \lambda_1 Q_j^g] \tag{18}$$

$$X_k \leftarrow X_k + \eta[\alpha e_{i,j} X_{i,k} ((1-\beta) Q_j^g + \beta \sum_{k=1}^{C} H_{j,k} Y_k) - \lambda_2 X_k] \tag{19}$$

$$Y_k \leftarrow Y_k + \eta[\beta e_{i,j} Y_{j,k} ((1-\alpha) P_i^g + \alpha \sum_{k=1}^{C} W_{i,k} X_k) - \lambda_2 Y_k] \tag{20}$$

$$b_i^g \leftarrow b_i^g + \eta[(1-\alpha) e_{i,j} - \lambda_3 b_i^g] \tag{21}$$

$$d_j^g \leftarrow d_j^g + \eta[(1-\beta) e_{i,j} - \lambda_3 d_j^g] \tag{22}$$

$$g_k \leftarrow g_k + \eta[\alpha e_{i,j} X_{i,k} - \lambda_4 g_k] \tag{23}$$

$$h_k \leftarrow h_k + \eta[\beta e_{i,j} Y_{j,k} - \lambda_4 h_k] \tag{24}$$

Where the parameter η indicates the learning rate. When the iterative process converges, we can use it for missing QoS prediction.

D. Computational Complexity Analysis

The training phase of our approach is mainly composed of three parts: PMF, FCM, and CFLFM. The computational complexity of PMF is $O(D|\Omega|)$. Where $|\Omega|$ indicates the number of training samples. For FCM, the iterative method is usually used to optimize the object function [20]. The computational complexity of user clustering is $O(MDC^2)$. Similarly, the computational complexity of service clustering is $O(NDC^2)$. Thus, the overall computational complexity of FCM is $O((M+N)DC^2)$. In fact, user clustering and service clustering can execute in parallel, because there is no data dependency between the two steps. For CFLFM, the most time-consuming step is validating the gradients of the object function. The computational complexity is $O(DC|\Omega|)$. Thus, the computation complexity of the whole train phase is approximately $O((M+N)DC^2 + |\Omega|CD)$. This shows the computation complexity is linear with respect to the number of training samples. Our approach is effective to scale to large dataset.

In the test phase, the computational complexity of predicting a missing QoS value is $O(CD)$.

IV. Experiments

In this section, we conduct experiments to validate our approach. We first introduce our experimental setup in section A. Next in section B, we compare our approach with other QoS-aware prediction approaches. Finally, we study the impact of model parameters in section C and section D.

A. Experimental Setup

We use a real-world Web Service QoS dataset to evaluate our model. This dataset includes QoS properties from 339

users on 5825 Web services. Every user invokes every Web service, respectively. So, a 339×5825 user-service QoS matrix is collected, containing a total of 1,974,675 records. Each element of the QoS matrix contains QoS properties: response time and throughput [23]. In most cases, response time is inverse proportional to throughput. Thus, we only validate our model in response time prediction.

We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the prediction accuracy.

$$MAE = \frac{\sum_{(u_i,s_j)\in T} |R_{i,j} - \hat{R}_{i,j}|}{|T|} \tag{25}$$

$$RMSE = \sqrt{\frac{\sum_{(u_i,s_j)\in T} (R_{i,j} - \hat{R}_{i,j})^2}{|T|}} \tag{26}$$

Where T is the testing set, $R_{i,j}$ is the true QoS value between user u_i and service s_j, $\hat{R}_{i,j}$ denotes the predicted QoS value. Since RMSE puts higher punishment on large errors, it's a more important metric than MAE.

B. Performance Comparison

To validate the effectiveness of our approach, we compare our approach with some popular approaches: UMEAN, IMEAN, UPCC, IPCC, UIPCC [3], PMF, NMF, BiasedMF and CloudPred [9]. UMEAN and IMEAN employ target user's or target service's average QoS value as the prediction. UPCC is a user-based method using PCC similarity measure. IPCC is a service-based method using PCC similarity measure. UIPCC is a linear combination of UPCC and IPCC. UPCC, IPCC and UIPCC only utilize local neighbor information for prediction. PMF, NMF and BiasedMF are model-based approaches that utilize global information for prediction. CloudPred is a hybrid approach, which combine both global and local information.

Since the QoS data matrix is very sparse in real world, we randomly remove some records as testing set. The remaining records are used as training set. We compare the performance in different QoS data density (5%, 15%, 30%). The Latent factor number for above model-based approaches and hybrid approaches are fixed to 10.

For our model, when the density is 5%, we set parameters: $\lambda_1 = \lambda_3 = 0.25$, $\lambda_2 = \lambda_4 = 0.00005$, $\alpha = 0.1$, $\beta = 0.2$. When the density is 15%, we set parameters: $\lambda_1 = \lambda_3 = 0.1$, $\lambda_2 = \lambda_4 = 0.00001$, $\alpha = 0.2$, $\beta = 0.3$. When the density is 30%, we set parameters: $\lambda_1 = \lambda_3 = 0.07$, $\lambda_2 = \lambda_4 = 0.00001$, $\alpha = 0.2$, $\beta = 0.3$. In all cases, we set cluster number $C = 6$ and weighting exponent $m = 2$. The performance comparison results are list in Table II.

From the results, it can be seen that CFLFM approach significantly outperforms other approaches, especially when the QoS dataset is sparse. This phenomenon indicates that it's necessary to combine both local and global information in sparse dataset.

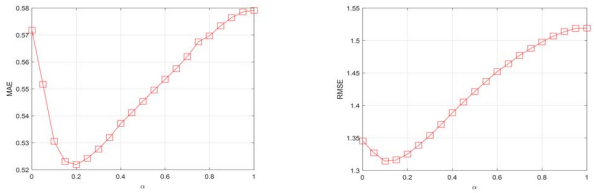
C. Impact of α and β

In this experiment, we set the QoS data density to 5% and set parameters: $D = 10$, $\lambda_1 = \lambda_3 = 0.25$, $\lambda_2 = \lambda_4 = 0.00005$, $C = 6$, $m = 2$.

TABLE II: Performance Comparison

Methods	Density=5%		Density=10%		Density=15%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.8777	1.8526	0.8757	1.8546	0.8748	1.8548
IMEAN	0.7020	1.5681	0.6837	1.5325	0.6785	1.5237
UPCC	0.6383	1.3813	0.5155	1.2602	0.4537	1.1714
IPCC	0.6354	1.3996	0.5096	1.2606	0.4155	1.1526
UIPCC	0.6163	1.3646	0.4919	1.2302	0.4206	1.1330
PMF	0.5387	1.4091	0.4552	1.1930	0.4126	1.1166
NMF	0.5355	1.4310	0.4479	1.2014	0.4075	1.1287
BiasedMF	0.5718	1.3450	0.4691	1.1866	0.4183	1.1101
CloudPred	0.5493	1.3367	0.4510	1.1808	0.4041	1.1099
CFLFM	**0.5032**	**1.2900**	**0.4360**	**1.1570**	**0.3989**	**1.0954**

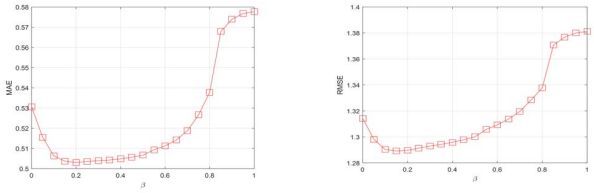
α is responsible for balancing the effect of user global information and user local information. $\alpha = 0$ means only user global information is considered. On the contrary, $\alpha = 1$ means only user local information is considered. To study the impact of parameter α, we first fix parameter $\beta = 0$. The results are shown in Fig.2.



(a) MAE vs. α (b) RMSE vs. α

Fig. 2: Impact of parameter α

From Fig.2, we can find that by considering user cluster local information, we can improve the performance of LFM. For our experiment setup, $\alpha = 0.1$ can achieve the best accuracy in RMSE. Then we fix $\alpha = 0.1$ to study the impact of parameter β. The results are shown in Fig.3.



(a) MAE vs. β (b) RMSE vs. β

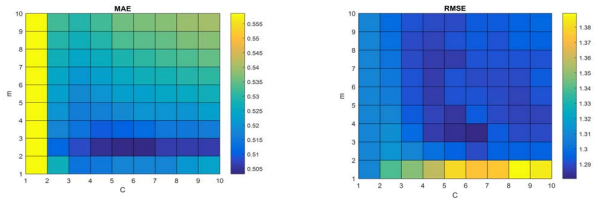
Fig. 3: Impact of parameter β

By combining service global information and local information, the performance can be further improved. Form Fig.3, we find $\beta = 0.2$ is the best choice.

D. Impact of Clustering Parameters

In the FCM clustering procedure, cluster number C and weighting exponent m are the most important parameters. The

cluster number C controls the scale of clusters. When $m \to 1.0$, FCM will be similar to K-means clustering method. When $m \to +\infty$, each user or service has the same membership degree to each cluster. In this experiment, we set the data density to 5%, $D = 10$, $\alpha = 0.1$, $\beta = 0.2$. We use grid search method to find the best cluster parameters, as shown in Fig.4.



(a) MAE vs. C and m (b) RMSE vs. C and m

Fig. 4: Impact of Clustering Parameters

When $C = 6$ and $m = 2$, we can achieve a good balance between MAE and RMSE. Although the computation complexity of our approach is proportional to the square of C, we can usually obtain satisfactory performance with a small cluster number.

V. Conclusion

This paper proposes a new hybrid collaborative filtering approach CFLFM for QoS prediction, by incorporating user and service latent local information. Experiments on a real-world dataset show our approach can achieve higher prediction accuracy than other popular approaches.

In the future, we shall consider how to improve our approach by incorporating context information (such as location, time). In addition, we will explore other approaches for QoS prediction.

Acknowledgment

This work is supported by The National key Research and Development Program of China (2016 YFB1000100).

References

[1] Z. P. D. Liang-Jie, Z. P. D. Jia, and C. P. D. Hong, *Services Computing*. Springer Berlin Heidelberg, 2007.

[2] M. P. Papazoglou, "Service-oriented Computing: Concepts, Characteristics and Directions." in *International Conference on Web Information Systems Engineering*, 2003, p. 3.

[3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommender System," in *IEEE International Conference on Web Services*, 2009, pp. 437–444.

[4] 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 Trans. Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 428–439, 2013.

[5] Z. Zheng, J. Chen, and M. R. Lyu, "Personalized Web Service Recommendation via Normal Recovery Collaborative Filtering," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2013.

[6] Z. Zheng and M. R. Lyu, "Personalized Reliability Prediction of Web Services," *ACM Trans. Softw. Eng. Methodol.*, vol. 22, no. 2, pp. 1–25, 2013.

[7] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "A Clustering-Based QoS Prediction Approach for Web Service Recommendation," in *IEEE International Symposium on Object/component/service-Oriented Real-Time Distributed Computing Workshops*, 2012, pp. 93–98.

[8] B. M. A. Madi, Q. Z. Sheng, L. Yao, Y. Qin, and X. Wang, "PLMwsp: Probabilistic Latent Model for Web Service QoS Prediction," in *IEEE International Conference on Web Services*, 2016, pp. 623–630.

[9] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring Latent Features for Memory-Based QoS Prediction in Cloud Computing," in *Reliable Distributed Systems*, 2011, pp. 1–10.

[10] L. Wei, J. Yin, S. Deng, Y. Li, and Z. Wu, "An Extended Matrix Factorization Approach for QoS Prediction in Service Selection," in *IEEE Ninth International Conference on Services Computing*, 2012, pp. 162–169.

[11] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2013.

[12] K. Qi, H. Hu, W. Song, J. Ge, and J. Lu, "Personalized QoS Prediction via Matrix Factorization Integrated with Neighborhood Information," in *IEEE International Conference on Services Computing*, 2015, pp. 186–193.

[13] D. Yu, Y. Liu, Y. Xu, and Y. Yin, "Personalized QoS Prediction for Web Services Using Latent Factor Models," in *IEEE International Conference on Services Computing*, 2014, pp. 107–114.

[14] X. Chen, X. Liu, Z. Huang, and H. Sun, "RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation," in *IEEE International Conference on Web Services*, 2010, pp. 9–16.

[15] L. Wei, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative Web Service QoS Prediction with Location-Based Regularization," in *IEEE International Conference on Web Services*, 2012, pp. 464–471.

[16] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware web service recommendation via non-negative tensor factorization," in *International Conference on World Wide Web*, 2014, pp. 585–596.

[17] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Towards Online, Accurate, and Scalable QoS Prediction for Runtime Service Adaptation," in *IEEE International Conference on Distributed Computing Systems*, 2014, pp. 318–327.

[18] Y. Zhang, Z. Zheng, and M. R. Lyu, 'WSPred: A Time-Aware Personalized QoS Prediction Framework for Web Services," in *IEEE International Symposium on Software Reliability Engineering*, 2011, pp. 210–219.

[19] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in *International Conference on Neural Information Processing Systems*, 2007, pp. 1257–1264.

[20] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c -means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[21] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-Based Hierarchical Matrix Factorization for Web Service Recommendation," in *IEEE International Conference on Web Services*, 2014, pp. 297–304.

[22] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, "SVDFeature: a toolkit for feature-based collaborative filtering," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3619–3622, 2012.

[23] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS Evaluation for Real-World Web Services," in *IEEE International Conference on Web Services*, 2010, pp. 83–90.