Collaborative Web Service Quality Prediction via Network Biased Matrix Factorization

Wenhao Zhong, Yugen Du*, Chuang Shan, Hanting Wang, Fan Chen
Shanghai Key Laboratory of Trustworthy Computing
Software Engineering Institute, East China Normal University
Shanghai, China
ygdu@sei.ecnu.edu.cn

***Abstract*—Facing a large number of candidate Web service with the same function, user wishes to get the most appropriate one. Quality-of-Service (QoS) which represents non-functional attributes of Web services, has become a major concern for choosing service. But it is time-consuming and resource-consuming to assess all the QoS values by invoking candidate services one by one. Thus, QoS prediction is considered an effective method to obtain QoS information. Although most of QoS prediction methods claim be able to capture the interaction between users and services, few of them take account non-interaction factors, especially the factors arising from the network environment. In this paper, the non-interaction factors from the network environment are referred as network bias, and a network biased matrix factorization (NBMF) method is proposed for QoS prediction. The method packages network bias into a linear regression model and puts the user-service interaction into a matrix factorization model, which is more sophisticated in adapting diversified circumstance, particularly in complex network environment. In addition, extensive experiments are conduct on real-world QoS dataset, and the result prove that the NBMF method achieves better performance than other state-of-the-art methods.**

***Index Terms*—Web services, QoS prediction, matrix factorization, network bias**

I. Introduction

The statistics published by ProgrammableWeb[1] indicate rapid growth in the number of published Web services over the past few years. The popularity of Web services allows different service-oriented applications and systems to be built to meet the increasingly complex business requirements [1].

To ensure the performance of service-oriented applications and systems, the quality of their component Web services needs to be assured. The quality of Web services can be described by their functional and non-functional attributes. Quality-of-Service (QoS) represents non-functional attributes of Web services, such as response time, throughput, availability, and reliability [2]. Since there are many Web services with similar functions on the network, investigating non-functional QoS attributes becomes a major concern for service selection [3], [4]. In practice, it is not easy to obtain the QoS values of all candidate services. First, the QoS values observed by users depend heavily on the invocation environment, and the quality of the same web service observed by different users may be very different [5]. Second, it is time-consuming and resource-consuming to assess all the QoS values by invoking candidate services one by one, due to a large number of users and services [6], [7]. Therefore, QoS prediction has attracted the attention of many researchers over the past few years and is considered an effective way to obtain QoS information.

Matrix factorization (MF) is arguably the most popular model-based collaborative filtering technique for QoS prediction [8], [9]. MF attempts to capture the interaction between users and services [10], [11], which factorizes the high-rank user-service matrix into two low-rank feature matrices [1], [3], and the inner product of the feature matrices represents the predicted QoS values of services observed by users. In addition to the user-service interaction, there are many factors unrelated to the interaction in the real-world prediciton. Although Zhu et al. [10] propose using user bias and service bias to capture the non-interaction from users and services, they do not take into account the non-interaction from the network environment. In the case of response time, the user-perceived response time must include network latency [12], which can vary significantly depending on the network path.

In this paper, we refer to the non-interaction factors from the network environment as network bias, and propose a network biased matrix factorization (NBMF) method for QoS prediction. Our method considers both interaction and non-interaction between users and services. Among the factors related to the non-interaction, we focus on the influence of network bias on QoS prediction. Specifically, our method packages network bias into a linear regression model, while putting the user-service interaction into a matrix factorization model, which makes our method adaptable to complex network environment.

In summary, the main contributions of this paper include:

- We propose a network biased matrix factorization method for QoS prediction. Our method considers both interaction and non-interaction between users and services, which makes our method adaptable to complex network environment.
- We conduct extensive experiments on a real-world QoS dataset to evaluate the performance of our method. The result demonstrates that our method can achieve better performance than the state-of-the-art baseline methods.

[1]https://www.programmableweb.com/

DOI reference number: 10.18293/SEKE2022-113

The remainder of this paper is organized as follows. In Section II, we explain the motivation of this work. In Section III, we introduce the proposed method. In Section IV, we conduct experiments on a real-world dataset to illustrate the effectiveness of the proposed method. In Section V, we review work related to our method. Finally, we conclude our work with future directions in Section VI.

II. Motivation

The traditional MF may not produce personalized QoS prediction, because it ignores the non-interaction bias between users and services, especially the non-interaction bias from the network environment. To explain this problem in detail, this section first introduces MF-based QoS prediction and then gives an example to illustrate the influence of network bias on MF-based QoS prediction.

A. Matrix Factorization

MF uses a factor model to fit the historical invocation matrix for prediction, which factorizes the user-service matrix into two low-rank feature matrices [1], [3]. The low-rank feature matrix attempts to explain the QoS data by describing the values on various latent features (e.g., system structure, hardware composition, software configuration). Each row of the user feature matrix represents the latent feature of a user, each row of the service feature matrix represents the latent feature of a Web service, and the dot product of them represents the user-service interaction, that is, the QoS value of the service observed by the user. Thus, the general objective function for the MF-based QoS prediction method can be derived as:

$$\hat{Q}_{ij} = U_i W_j^T \tag{1}$$

Where $U \in \mathbb{R}_{m \times d}$ denotes the user latent feature matrix, and $W \in \mathbb{R}_{n \times d}$ denotes the service latent feature matrix. The vector $U_i (1 \leq i \leq m)$ denotes the latent feature vector of user i, and the vector $W_j (1 \leq j \leq n)$ denotes the latent feature vector of service j. The number of latent features in our model is d. The predicted QoS value of Web service j observed by user i is $\hat{Q}_{ij}$.

B. Motivating Example

To explain the influence of network bias on MF-based QoS prediction, a straight-away example is given in Fig. 1(a). In this example, we need to predict the QoS values of services w_1, w_2, w_1', w_2' observed by users u_1, u_2. We assume that u_1, u_2 are located in region R_1, w_1, w_2 are located in region R_2, and w_1', w_2' are located in region R_3. Where services w_1, w_2 have the same performance as services w_1', w_2', the average response time between R_1 and R_2 is 1s, and the average response time between R_1 and R_2 is 2s.

The QoS prediction of traditional matrix factorization is shown in Fig. 1(b), we assume that the QoS value of the service observed by the user is determined by two latent features, which are hardware composition and software configuration. In the following, we will focus on user u_1 and services w_1, w_1'. The user-perceived hardware composition feature to u_1 is 0.6, and the user-perceived software configuration feature to u_1 is 0.4. The service-provided hardware composition feature to w_1 is 1.0, and the service-provided software configuration feature to w_1 is 0.5. The dot product of u_1 and w_1 is 0.8, which represents the QoS value of w_1 observed by u_1. Since the performance of services w_1 and w_1' are exactly the same, they have the same degree of latent features, the QoS value of w_1' observed by u_1 is also 0.8.

However, the traditional matrix factorization ignores the non-interaction bias between users and services, especially the non-interaction bias from the network environment. In Fig. 1(a), although the performance of services w_1 and w_1' are the same, they belong to different regions (w_1 is located in R_2, w_1' is located in R_3), and their QoS values observed by u_1 should be very different. In this case, the QoS prediction made by previous work is not accurate.

III. Method

Since the non-interaction bias from the network environment accounts for a large proportion of the observed QoS values, it is crucial to model this bias accurately. To achieve this goal, we propose a network biased matrix factorization (NBMF) method for QoS prediction. In this section, we first introduce the proposed NBMF method, then give an example to illustrate the improved prediction, and finally describe the model training and parameter optimization process.

A. Network Biased Matrix Factorization

If we consider the network bias to be continuous rather than discrete, then linear regression can be used to predict the network bias between users and services. Following the above idea, we propose a network biased matrix factorization method. Our method packages network bias into a linear regression model, while putting the user-service interaction into a matrix factorization model. Thus, the general objective function for NBMF-based QoS prediction method can be derived as:

$$\hat{Q}_{ij} = \alpha(\mu_{xy} + \mathbf{b}_i + \mathbf{p}_j) + (1 - \alpha) U_i W_j^T \tag{2}$$

The first term $\alpha(\mu_{xy} + \mathbf{b}_i + \mathbf{p}_j)$ of the objective function is a linear regression model used to predict the network bias between user i and service j. Where, x is the network of user i, y is the network of service j, and μ_{xy} is the average QoS value between network x and network y. $\mathbf{b}_i (1 \leq i \leq m)$ denotes the bias between the QoS observed by user i and other users in the same network. $\mathbf{p}_j (1 \leq j \leq n)$ denotes the bias between the QoS provided by service j and other services in the same network.

The second term $(1 - \alpha) U_i W_j^T$ of the objective function is a matrix factorization model used to capture the interaction between user i and service j , where $U_i (1 \leq i \leq m)$ denotes the latent feature vector of user i, $W_j (1 \leq j \leq n)$ denotes the latent feature vector of service j, and their dot product $U_i W_j^T$ represents the interaction between user i and service j.

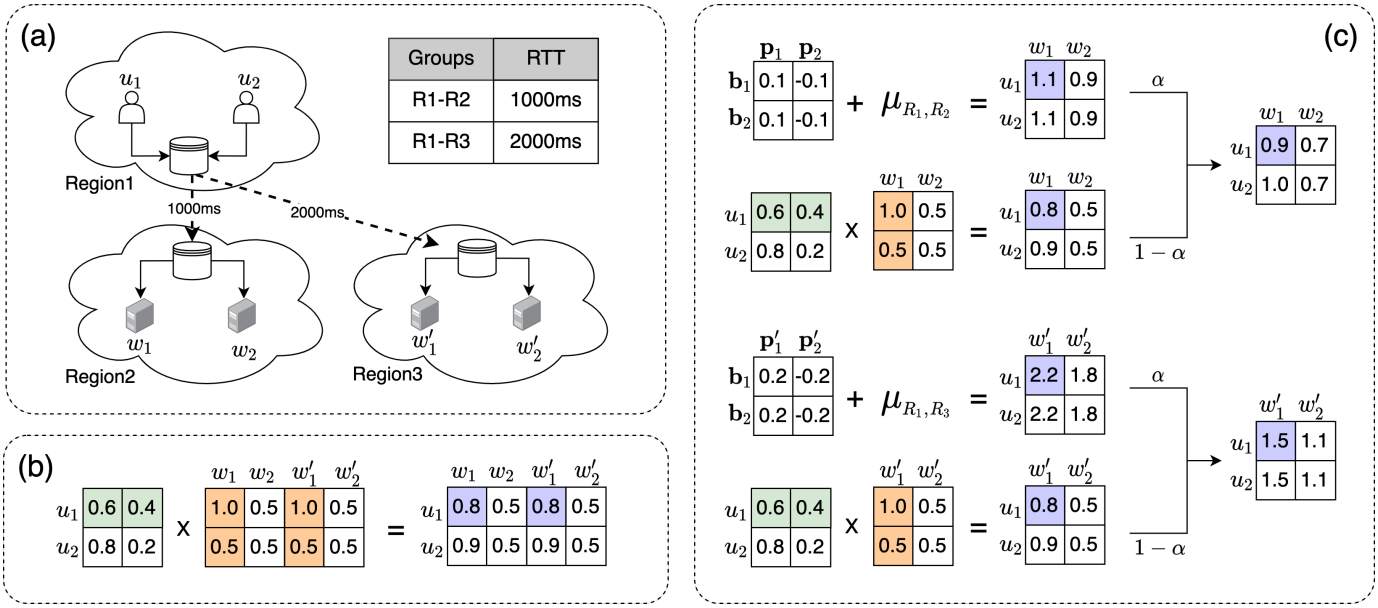

Fig. 1. An example to explain the influence of network bias on MF-based QoS prediction: (a) A real-world Web service invocation scenario; (b) MF-based QoS prediction; (c) NBMF-based QoS prediction.

The weight $\alpha(0 \leq \alpha \leq 1)$ measures the degree of network bias in our prediction model. α is an adjustable parameter. If α is set to 0, our prediction model does not consider network bias and only uses matrix factorization for prediction. If α is set to 1, our prediction model does not consider the user-service interaction and only uses linear regression for prediction. In order to investigate the impact of α on our model and find an optimal model, the value of α will be evaluated in Section IV.

B. Method Example

The improved QoS prediction is shown in Fig. 1(c). In the case of user u_1 invoking services w_1, w_1', the average QoS μ_{R_1,R_2} between region R_1 and region R_2 is 1, and the average QoS μ_{R_1,R_3} between region R_1 and region R_3 is 2. We assume the bias $\mathbf{b}_1$ between u_1 and other users in R_1 is 0, the bias $\mathbf{p}_1$ between w_1 and other services in R_2 is 0.1, the bias $\mathbf{p}_1'$ between w_1' and other services in R_3 is 0.2. Then the network bias between u_1 and w_1 is 1.1, and the network bias between u_1 and w_1' is 2.2. The next part of the QoS prediction is the user-service interaction, and the detailed procedure can be found in Section II-B.

The parameter α is set to 0.5 by default. After weighted sum of the network bias and the interaction, we can get the predicted QoS value of w_1 observed by u_1 is 0.9, and the predicted QoS value of w_1' observed by u_1 is 1.5.

Although the performance of services w_1 and w_1' are the same, they belong to different regions, and their QoS values observed by u_1 are different. It can be seen that our method is adaptable to complex network environment.

C. Model Training

The latent feature matrices and the bias vectors in Eq. (2) can be constructed by statistical learning theory. To estimate the values of matrices U, W and vectors $\mathbf{b}, \mathbf{p}$, we approximate the original matrix Q with the following objective function, and the minimization formula is as follows:

$$L = min_{U,W,\mathbf{b},\mathbf{p}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (Q_{ij} - \hat{Q}_{ij})^2 \tag{3}$$

Where I_{ij} is the indicator function that returns 1 if user i has invoked service j, and 0 otherwise. $\hat{Q}_{ij}$ is the prediction function as in Eq. (2). To avoid overfitting in approximating the original matrix, we add four regular terms related to U, W and $\mathbf{b}, \mathbf{p}$.

$$\begin{aligned} L = min_{U,W,\mathbf{b},\mathbf{p}} & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (Q_{ij} - \hat{Q}_{ij})^2 \\ & + \frac{\lambda}{2} (\|U\|_F^2 + \|W\|_F^2 + \|\mathbf{b}\|_F^2 + \|\mathbf{p}\|_F^2) \end{aligned} \tag{4}$$

Where $\|\cdot\|$ denotes the *Frobenius norm* [13], the parameter λ controls the degree of regularization. The objective function based on L_2-norm as in Eq. (4) is not convex, it is unrealistic to design an algorithm to find the global minimum [14]. Instead, the stochastic gradient descent technique [15] can be employed to find the approximate optimal solution. For each QoS record observed when user i invokes service j, we have the following update rules:

$$U_i' = U_i - \eta \frac{\partial L}{\partial U_i} \tag{5}$$

$$W_j' = W_j - \eta \frac{\partial L}{\partial W_j} \tag{6}$$

$$\mathbf{b}_i' = \mathbf{b}_i - \eta \frac{\partial L}{\partial \mathbf{b}_i} \tag{7}$$

$$\mathbf{p}_j' = \mathbf{p}_j - \eta \frac{\partial L}{\partial \mathbf{p}_j} \tag{8}$$

where $\eta > 0$ represents the learning rate of updating the feature matrices and bias vectors, and

$$\frac{\partial L}{\partial U_i} = \lambda U_i - (Q_{ij} - \hat{Q}_{ij})(1-\alpha)W_j \tag{9}$$

$$\frac{\partial L}{\partial W_j} = \lambda W_j - (Q_{ij} - \hat{Q}_{ij})(1-\alpha)U_i \tag{10}$$

$$\frac{\partial L}{\partial \mathbf{b}_i} = \lambda \mathbf{b}_i - (Q_{ij} - \hat{Q}_{ij})\alpha \tag{11}$$

$$\frac{\partial L}{\partial \mathbf{p}_j} = \lambda \mathbf{p}_j - (Q_{ij} - \hat{Q}_{ij})\alpha \tag{12}$$

The overall optimization procedure of our method is given in Algorithm 1. Let r denote the number of iterations to achieve convergence, let s denote the number of valid invocation records in the original matrix Q, and let d denote the dimensions of the user latent feature matrix and the service latent feature matrix. The main time cost of Algorithm 1 lies in the updating of matrices U, W and vectors $\mathbf{b}, \mathbf{p}$. In each iteration, updating U, W takes $\mathcal{O}(sd)$ time and updating $\mathbf{b}, \mathbf{p}$ takes $\mathcal{O}(s)$ time. Thus, the overall time complexity of our method is $\mathcal{O}(rsd)$.

Algorithm 1 Optimization procedure of NBMF

Input: $Q \in \mathbb{R}^{m \times n}, \alpha, d, \lambda, \eta$;
Output: $U \in \mathbb{R}^{m \times d}, W \in \mathbb{R}^{n \times d}, \mathbf{b} \in \mathbb{R}^m, \mathbf{p} \in \mathbb{R}^n$

```
Randomly initialize U and W;
Initialize b and p with zero;
repeat
    for each record (i, j, Q_ij) observed in Q do
        Update U_i according to Eq. (5);
        Update W_j according to Eq. (6);
        Update b_i according to Eq. (7);
        Update p_j according to Eq. (8);
    end for
until Convergence
return U, W, b, p;
```

IV. Experiment

In this section, we conduct extensive experiments on a real-world QoS dataset to evaluate the performance of NBMF. The experiments are designed to address the following questions: (1) How does NBMF method compare with other state-of-the-art baseline methods? (2) How does the matrix density affect the prediction accuracy? (3) How does the weight α of the network bias affect the prediction accuracy? (4) How does the dimension d of the latent feature matrix affect the prediction accuracy? We implement our method and all baseline methods in Python 3.7, and all experiments were performed on a Linux server with Intel i5-10400 2.9GHz CPU and 16GB RAM running 64-bit Ubuntu 16.04.

A. Dataset

We conduct all experiments on a publicly real-world QoS dataset named WS-DREAM[2]. The dataset includes 1,974,675 QoS records, which were collected from 339 users in 30 regions on 5825 Web services in 73 regions. There is a QoS record between each user and each service, and we focus on the response time (RT) in the QoS attribute. Also, the dataset collects the IP, region and other information of these users and services. More details about this dataset can be found in [16].

B. Evaluation Metrics

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics are used to measure the accuracy of prediction by calculating the difference between the predicted QoS value and the actual QoS value.

MAE is defined as:

$$MAE = \frac{1}{N} \sum_{i,j} |Q_{ij} - \hat{Q}_{ij}| \tag{13}$$

RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j} (Q_{ij} - \hat{Q}_{ij})^2} \tag{14}$$

Where Q_{ij} and $\hat{Q}_{ij}$ represent the actual and predicted QoS value of the user i invoking service j, and N denotes the number of predicted QoS values. It can be observed from the formula that RMSE is more sensitive to large errors. MAE and RMSE range from 0 to ∞, and their smaller values indicate better performance of the prediction method.

C. Accuracy Comparison

To demonstrate the prediction accuracy of our method, we reproduce the 6 most representative QoS prediction methods and compare the NBMF with them.

- UMEAN: This method employs the average QoS value of a user to predict the unknown QoS value.
- IMEAN: This method employs the average QoS value of a server to predict the unknown QoS value.
- UIPCC [17]: This method is user-based and service-based CF, which employs similar users and similar services for QoS prediction.
- PMF [18]: This method is probability-based MF, which factorizes the matrix into two low-rank latent feature matrices for QoS prediction.
- HMF [7]: This method improves the MF with location clustering.
- AMF [10]: This method improves the MF with user bias and service bias.
- NBMF: This method improves the MF with network bias.

[2]https://github.com/wsdream/wsdream-dataset

TABLE I
QOS PREDICTION ACCURACY (SMALLER MRE AND RMSE INDICATE BETTER ACCURACY)

Methods	MAE					RMSE				
	D = 15%	D = 20%	D = 25%	D = 30%	Improve	D = 15%	D = 20%	D = 25%	D = 30%	Improve
UMEAN	0.8739	0.8738	0.8737	0.8735	47.95%	1.8569	1.8568	1.8564	1.8576	36.60%
IMEAN	0.6821	0.6805	0.6789	0.6781	33.11%	1.5366	1.5305	1.5280	1.5267	23.08%
UIPCC	0.5861	0.5768	0.5725	0.5714	21.14%	1.4464	1.4322	1.4262	1.4247	17.81%
PMF	0.5236	0.5015	0.4904	0.4652	8.16%	1.2556	1.2303	1.2185	1.1949	3.88%
AMF	0.5024	0.4836	0.4589	0.4444	3.72%	1.244	1.2196	1.1977	1.1772	2.68%
HMF	0.4994	0.4713	0.4545	0.4438	2.67%	1.2354	1.2162	1.1819	1.1582	1.73%
NBMF	**0.4847**	**0.4579**	**0.4459**	**0.4306**	-	**1.2232**	**1.1843**	**1.1627**	**1.1388**	-

In the real world, the user-service matrix is very sparse, as users usually invoke only a small number of Web services. In this paper, to simulate the matrix environment with different densites, we randomly remove a certain number of QoS values from the dataset to generate the user-service matrix with densities of 15%, 20%, 25%, 30%. The removed QoS values are used as expected values to evaluate the prediction accuracy achieved by different methods. For example, a matrix density of 15% means that we randomly select 15% QoS values in the original matrix to predict the remaining 85% QoS values.

In the experiments, the parameters of the baseline methods are initialized according to the corresponding papers for optimal performance, and the parameters of our NBMF method are set to $\alpha = 0.6$, $d = 6$, $\lambda = 0.02$, $\eta = 0.003$, the maximum number of iterations in the model training is set to 300. In addition, we perform an early stopping strategy during the model training, where we stop training if the evaluation metric on the testing set increases five times in a row.

Table I provides the prediction accuracies of different methods at 15% to 30% matrix density. We can observe that NBMF achieves an improvement of 2.67∼47.95% in MAE and 1.73∼36.6% in RMSE compared with other classical prediction methods, and NBMF has the smallest MAE and RMSE regardless of matrix density, which indicates that our method has the best prediction accuracy. Compared with the AMF method, NBMF method achieves 3.72% and 2.68% improvement in MAE and RMSE. Because the real-world network environment is very complex, it is more suitable to consider network bias than user bias and service bias for real QoS prediction systems. Compared with the HMF method, NBMF method achieves 2.67% and 1.73% improvement in MAE and RMSE. Because the HMF method clusters based on regions, while the NBMF method clusters based on network paths, which is more adaptable to complex network environments. The prediction accuracy of all methods improved significantly as the matrix density increased from 15% to 30%, suggesting that more QoS information can contribute to higher prediction accuracy.

D. Impact of Parameter α

The parameter α measures the degree of network bias in our prediction model. If α is set to 0, our prediction model does not consider network bias, in which case NBMF is equivalent to PMF. If α is set to 1, our prediction model does not consider the user-service interaction and only uses linear regression for prediction. To evaluate the impact of α and find an optimal model, we set the dimension d to 6 and set the density D to 15% and 30%.

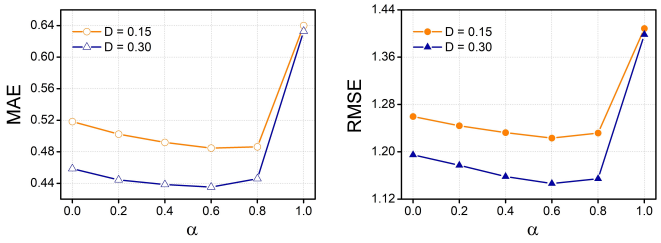

Fig. 2. Impact of Parameter α

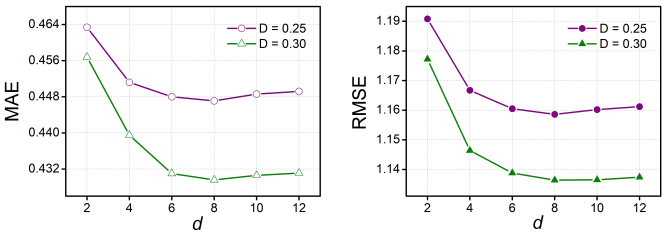

Fig. 3. Impact of Dimension d

Fig. 2 shows us the changes in MAE and RMSE when α is adjusted from 0 to 1. Before the prediction accuracy reaches the best, the values of MAE and RMSE decrease as the value of α increases, indicating that the prediction accuracy is improved. However, when the value of α exceeds a certain threshold, the values of MAE and RMSE increase instead, indicating a decrease in prediction accuracy. We observe that the thresholds for both MAE and RMSE are around $\alpha = 0.6$ for all densities of the matrix environment. The existence of the thresholds confirm our intuition that the best prediction performance can be achieved by a proper combination of MF and network bias. In addition, we find that our NBMF method is quite stable, as it maintains similar trends for different criteria in all configurations.

E. Impact of Dimension d

In our proposed method, d denotes the dimension of the low-rank latent feature matrix, i.e., the number of latent features in matrix factorization. If d is small, only a few key latent features determine the QoS value. If d is large, many latent features jointly determine the QoS value. To investigate the impact of the dimension d on prediction results, we set the parameter α to 0.6 and set the density D to 25% and 30%.

Fig. 3 presents the changes in MAE and RMSE when the dimension d is adjusted from 2 to 12. As the dimension increases, the values of MAE and RMSE decrease rapidly at first, indicating that only a few latent features cannot achieve good prediction results. However, when the dimension exceeds a certain threshold, the values of MAE and RMSE gradually increase. Because higher dimension leads to overfitting problems, which reduces the prediction performance.

V. Related Work

In recent years, collaborative filtering (CF) has been widely used for QoS prediction [8], [9]. Existing CF-based prediction methods can be classified into two types: memory-based CF and model-based CF. Memory-based CF first finds similar users or services by Pearson correlation coefficient (PCC), and then uses the QoS values of similar users or services to predict the missing values [5], [17]. However, a user may have invoked only few services in the real world, which reduces the accuracy of calculating similarity using PCC [1]. Model-based CF trains a global model to make predictions based on observed historical invocation records, and it performs well when dealing with sparse user-service matrix [3].

Matrix factorization (MF) is arguably the most popular model-based CF technique [8], [9], which attempts to capture the interaction between users and services [10], [11]. He et al. [7] introduced a hierarchical MF method based on location grouping, they assumed that local invocations reflect more interaction than global ones. Zhang et al. [2] proposed a neighborhood-integrated MF method, which uses PCC to calculate the similarity between users. Tang et al. [19] employed the IP addresses and Ryu et al. [20] employed the location information to improve the calculation of similar users, they assumed that similar users have similar interaction with services. In addition to the user-service interaction, there are many factors unrelated to the interaction in the real-world prediciton. Although Zhu et al. [10] propose using user bias and service bias to capture the non-interaction from users and services, they do not take into account the non-interaction from the network environment.

VI. Conclusion

We propose a network biased matrix factorization method for QoS prediction. Our method considers both interaction and non-interaction between users and services, which makes our method adaptable to complex network environment. We conduct extensive experiments on a real-world QoS dataset. The result demonstrates that our method can achieve better performance than the state-of-the-art baseline methods.

In the future, we intend to improve the current work as follows: First, we will conduct experiments to evaluate the prediction performance of NBMF on other QoS attributes. Second, considering the dynamic nature of Web services, we will try to implement real-time QoS prediction with weighing factor based on time preference.

References

[1] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, and Y. Yang, "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.

[2] 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, 2012.

[3] F. Ye, Z. Lin, C. Chen, Z. Zheng, and H. Huang, "Outlier-resilient web service qos prediction," in *Proceedings of the Web Conference 2021*, 2021, pp. 3099–3110.

[4] J. El Hadad, M. Manouvrier, and M. Rukoz, "Tqos: Transactional and qos-aware selection algorithm for automatic web service composition," *IEEE Transactions on Services Computing*, vol. 3, no. 1, pp. 73–85, 2010.

[5] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *Ieee international conference on web services (icws 2007)*. IEEE, 2007, pp. 439–446.

[6] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang, "Qos prediction of web services based on two-phase k-means clustering," in *2015 ieee international conference on web services*. IEEE, 2015, pp. 161–168.

[7] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for web service recommendation," in *2014 IEEE international conference on web services*. IEEE, 2014, pp. 297–304.

[8] S. H. Ghafouri, S. M. Hashemi, and P. C. Hung, "A survey on web service qos prediction methods," *IEEE Transactions on Services Computing*, 2020.

[9] Z. Zheng, L. Xiaoli, M. Tang, F. Xie, and M. R. Lyu, "Web service qos prediction via collaborative filtering: A survey," *IEEE Transactions on Services Computing*, 2020.

[10] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online qos prediction for runtime service adaptation via adaptive matrix factorization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2911–2924, 2017.

[11] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative web service qos prediction with location-based regularization," in *2012 IEEE 19th international conference on web services*. IEEE, 2012, pp. 464–471.

[12] A. Klein, F. Ishikawa, and S. Honiden, "Towards network-aware service composition in the cloud," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 959–968.

[13] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[14] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1–24, 2010.

[15] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[16] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *2010 IEEE International Conference on Web Services*. IEEE, 2010, pp. 83–90.

[17] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *2009 IEEE International Conference on Web Services*. IEEE, 2009, pp. 437–444.

[18] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, 2007.

[19] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative web service quality prediction via exploiting matrix factorization and network map," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126–137, 2016.

[20] D. Ryu, K. Lee, and J. Baik, "Location-based web service qos prediction via preference propagation to address cold start problem," *IEEE Transactions on Services Computing*, 2018.