

Personalized QoS Prediction of Cloud Services via Learning Neighborhood-Based Model

Hao Wu¹(✉), Jun He², Bo Li¹, and Yijian Pei¹

¹ School of Information Science and Engineering, Yunnan University,
Kunming 650091, China

{haowu, libo, pei3p}@ynu.edu.cn

² Nanjing University of Information Science and Technology,
Nanjing 210044, China

hejun.zz@gmail.com

Abstract. This paper proposes neighborhood-based approach for QoS-prediction of cloud services by taking advantages of collaborative intelligence. Different from heuristic collaborative filtering and matrix-factorization, we set a formal neighborhood-based prediction framework which allows an efficient global optimization scheme, and then exploits different baseline estimate components to improve predictive performance. To validate our methods, a large-scale QoS-specific dataset which consists of invocation records from 339 service users on 5,825 web services on a world-scale distributed network is used. Experimental results show that the learned neighborhood-based models can overcome existing difficulties of heuristic collaborative filtering methods and achieve superior performance than state-of-the-art prediction methods.

Keywords: Cloud services · QoS prediction · Neighborhood model · Parameter learning

1 Introduction

The explosion of cloud services on the Internet brings new challenges in service discovery and selection [1]. As confronted with a number of functionally similar cloud services, a user may feel hard to judge what is the extent of candidates in line with the individual needs. A further comparison of the non-functionality of candidates (generally the properties of QoS) is needed in order to make the best choice. However, due to constraints on time, costs and other factors [2], service providers cannot dispose a large number of software sensors in cloud environments to monitor QoS information for every service. Also, it is not realistic for users to carry out large-scale testings to experience the individual differences of QoS. Consequently, how to obtain personalized QoS of cloud services and assist users selecting appropriate services become one urgent issue.

Recently, researchers begin to pursuit solution for this problem by drawing lessons from recommender systems [2–12]. The main idea of their works is to

analyze the QoS usage data of users in service-oriented systems, further exploit collaborative intelligence to prediction unknown QoS values. With distinguishable quality values of candidate services, users can take a decision on choosing appropriate services. In such a manner, it can avoid direct QoS measurement [13], and thereby save time and economic costs for both service providers and users.

As for the collaborative QoS-prediction, neighborhood-based collaborative filtering (NbCF) [3–7] and matrix-factorization (MF) [2, 8–12] are commonly used methods. NbCF is simplicity, justifiability and efficiency [14]. However, the model is not justified by a formal model. Moreover, heterogenous similarity metrics and sparsity-sensitive problem make these models not robust and scalable enough. In contrast, MF approaches comprise an alternative way to CF with the more holistic goal to uncover latent features from user-service usage data, and thus can mitigate the problem of sensitivity to sparse data. Since MF can be presented as a formal optimization problem and solved by machine learning methods, thus it provides attractive accuracy and scalability for QoS prediction. However, MF is always uncertain, resulting in difficulty as explain the predictions for users. It is an important issue as users of cloud services certainly hope to get reasonable explanations for the QoS predictions provided by a service recommendation system. Inspired by this, we propose learning neighborhood-based models for personalized QoS-prediction of cloud services, to have the best of both worlds (NbCF and MF). We define formal neighborhood models which permit an efficient global optimization scheme and exploit different baseline estimate components to improve prediction performance. Experimental results demonstrate that learning neighborhood model can overcome existing difficulties, and perform superior to the state-of-art prediction methods.

The remainder of the paper is structured as follows. We review some existing works that are most relevant to ours in Sect. 2. We provide details of the proposed neighborhood-based prediction approaches in Sect. 3. We measure the effectiveness of the proposed methods via a set of experiments on real QoS data in Sect. 4 and conclude in Sect. 5.

2 Related Works

2.1 QoS Prediction Based on Neighborhood Model

Shao et al. [3] at first proposed the use of NbCF based on the user. Firstly, Pearson correlation coefficient (PCC) measurement is utilized to calculate pairwise-similarity among all users on the user-service matrix of QoS data. Secondly, historical quality values of target service provided by the top-k similar users of active user are fused to achieve prediction result. Follow-up research works basically follow this idea but concentrate on improving the similarity metrics to accurately quantify the correlations of users or services. Zheng et al. [4] proposed a mixed model that integrated user-based and item-based approaches linearly by confidence weights and proved the mixed model is better than a single one. Sun et al. considered the distribution characteristics of QoS data to calculate the

similarity [5]. Wu et al. [6] and Chen et al. [7] proposed location-aware similarity metrics to find neighbors of users and services. These methods have not been exploited machine learning technology, hence can be seen as heuristic-based prediction ways. NbCF became very popular because they are relatively simple to implement and provide intuitive explanations for the prediction results. However, some concerns about NbCF always exist. Most notably, these methods are not justified by a formal model. The selection of heterogenous similarity metric clearly affects the accuracy of QoS-prediction, thus make these models not robust. When the QoS data are sparse, the predictive power can be significantly reduced, resulting in the sparsity-sensitive problem. This motivates us to develop more accurate neighborhood models to resolve existing difficulties.

2.2 QoS Prediction Based on Matrix Factorization

MF can partially alleviate the sparsity-sensitive problem of collaborative filtering, and thus improves the accuracy of QoS prediction. In recent years, numerous efforts have been made on improving MF-based models. The works are focused on utilizing additional information, such as spatial and temporal information associated with users or services. Zhang et al. [9] use collective matrix factorization that simultaneously factor the user-service quality matrix, service category and location context matrices. Yin et al. [10] develop a location-based regularization framework for PMF prediction model. Lo et al. [12] exploit PMF with a location-based pre-filtering stage on QoS matrix. He et al. [11] develop location-based hierarchical matrix factorization. Yu et al. [8] experience trace-norm regularized MF. MF is still uncertain, resulting in difficulty as explain the predictions for users. Also, data sparsity has a negative effect on these methods, as data becomes extremely sparse, the prediction performance will be not optimistic.

3 Learning Neighborhood-Based Model for QoS Prediction

We reserve special indexing letters for distinguishing users from services: for users u, v , and for services i, j . A QoS r_{ui} indicates the observed quality of user u on service i . We distinguish predicted quality from known ones, by using the notation \hat{r}_{ui} for the predicted value of r_{ui} . The (u, i) pairs for which r_{ui} is known are stored in the set $\mathbf{E} = \{(u, i) | r_{ui} \text{ is known}\}$.

3.1 Neighborhood-Based Models(NbModels)

In cloud computing, the context with users is more complicated and dynamic than that of services. Prediction leveraged by similar users other than services is more reasonable. Thus, our focus is on user-oriented approaches, but parallel techniques can be developed in a service-oriented fashion, by switching the roles of users and services. We borrow ideas from collaborative filtering research [15],

which allows an efficient global optimization scheme and offers improved accuracy. To facilitate global optimization, we abandon user-specific weights in favor of global weights independent of a specific user. The weight from user v to user u is denoted by w_{uv} are able to be learned from the data through optimization. By this, we can overcome the weaknesses with existing neighborhood-based models. An initial sketch of the model describes each quality score r_{ui} by (1):

$$\hat{r}_{ui} = b_{ui} + \sum_{v \in \mathcal{N}_u} (r_{vi} - b_{vi})w_{uv}, \quad (1)$$

where \mathcal{N}_u is the neighbor set of user u , b_{ui} is the basic estimate that we will gradually construct considering different factors.

With respect to the interpretation of weights, usually they represent interpolation coefficients relating unknown quality scores to the existing ones in a traditional neighborhood model. Here, we adopt them in a different viewpoint that weights represent offsets to basic estimates and residual, $r_{vi} - b_{vi}$, are viewed as the coefficients multiplying those offsets. For two similar users u and v , w_{uv} is always expected to get high, and vice versa. So, our estimate will not deviate much from the basic estimate by a user v that accessed i just as expected ($r_{vi} - b_{vi}$ is around zero), or by a user v that is not known to be predictive on u (w_{uv} is close to zero).

Generally, we can take all users in \mathcal{N}_u other than u , however, this would increase the number of weights to be estimated. In order to reduce complexity of the model, we suggest pruning parameters corresponding to unlikely user-user relations. Let \mathcal{N}_u^k be the set of k users most similar u , as determined by a similarity measure (e.g. PCC). Further, we let $\mathcal{N}_{(i;u)}^k \triangleq \mathcal{N}_i \cap \mathcal{N}_u^k$, where \mathcal{N}_i is the set of users have used the service i . Now, when predicting r_{ui} according to Formula (1), it is expected that the most influential weights will be associated with users similar to u . Hence, we replace (1) with:

$$\hat{r}_{ui} = b_{ui} + |\mathcal{N}_{(i;u)}^k|^{-\frac{1}{2}} \sum_{v \in \mathcal{N}_{(i;u)}^k} (r_{vi} - b_{vi})w_{uv} \quad (2)$$

When $k = \infty$, rule (2) coincides with (1). When $k = 0$, $\hat{r}_{ui} = b_{ui}$. However, for other values of k , it offers the potential to significantly reduce the number of variables involved. This final prediction rule permits fast online prediction, since more computational works, such as similarity calculation and parameter estimation, have been made in the pre-processing stage. Recall that unlike matrix-factorization, neighborhood models allow an intuitive explanation of their predictions, and do not require re-training the model for handling new services.

3.2 Components for Estimating b_{ui}

Baseline estimate. Typical QoS data exhibit large user and service effects-i.e., systematic tendencies for some users to achieve better QoS than others, and for some services to receive better QoS than others. It is customary to adjust

the QoS data by accounting for these effects, which we encapsulate within the baseline estimates. Denote by μ the average QoS value observed in the entire dataset. A baseline estimate for an unknown QoS r_{ui} is denoted by b_{ui} and accounts for the user and service effects:

$$b_{ui} = \mu + b_u + b_i \quad (3)$$

The parameters b_u and b_i respectively indicate the observed deviations of user u and service i from the average. Suppose we want to estimate the response-time of Google-Search service by Tom. Now, leave the average response time, $\mu=2\text{ms}$. Further, since Google-search is recognized better than an average search service, we can suppose it is faster 0.5ms than the average. In addition, Tom's network condition is not a good, which tends to be 1ms delay than the average. Thus, the baseline estimate for Google-Search's response-time by Tom would be 2.5ms by calculating: $2-0.5+1$. Substituting (3) into (2), we will obtain NbModel1. To estimate b_u , b_i and w_{uv} one can solve the least squares problem:

$$\begin{aligned} \min_{w^*, b^*} \sum_{(u,i) \in \mathbf{E}} [r_{ui} - b_{ui} - |\mathcal{N}_{(i;u)}^k|^{-\frac{1}{2}} \sum_{v \in \mathcal{N}_{(i;u)}^k} (r_{vi} - b_{vi})w_{uv}]^2 \\ + \lambda_1 \sum_{v \in \mathcal{N}_{(i;u)}^k} w_{uv}^2 + \lambda_2 (b_u^2 + b_i^2) \end{aligned} \quad (4)$$

Here, the first term $\sum_{(u,i) \in \mathbf{E}} (\cdot)^2$ strives to find b_u 's, b_i 's and w_{uv} 's that fit the given usage data. The second part and the third part both are the regularizing terms, employed to avoid overfitting by penalizing the magnitudes of the parameters. λ_1 and λ_2 are the specific regularization parameters [15].

Weighted features. The baseline estimate just considers the mean effect of user and item in an intuitive manner. However, more bias information (such as user features, item features, time bias) can be used to enhance the prediction model. We may consider an unknown quality value r_{ui} as a linear combination of the features of user and item. Here, the user-specific QoS mean and the service-specific QoS mean are taken as the two key features, because they strongly reflect the bias effect of users and services, and the expect prediction of QoS can be found in a value domain determined by them. Denote by μ_u the average QoS value observed by user u and μ_i the average QoS value observed by service i . A feature-weighted estimate for r_{ui} is denoted by b_{ui} as followings:

$$b_{ui} = w_u \mu_u + w_i \mu_i \quad (5)$$

The parameters w_u and w_i indicate the feature importance of user u and item i , respectively. Substituting (5) into (2), we will obtain NbModel2. To estimate w_u , w_i , we need to solve the following least squares problem:

$$\begin{aligned} \min_{w^*} \sum_{(u,i) \in \mathbf{E}} [r_{ui} - b_{ui} - |\mathcal{N}_{(i;u)}^k|^{-\frac{1}{2}} \sum_{v \in \mathcal{N}_{(i;u)}^k} (r_{vi} - b_{vi})w_{uv}]^2 \\ + \lambda_1 \sum_{v \in \mathcal{N}_{(i;u)}^k} w_{uv}^2 + \lambda_3 (w_u^2 + w_i^2) \end{aligned} \quad (6)$$

Here, regularizing term, $\lambda_3(w_u^2 + w_i^2)$, avoid overfitting by penalizing the magnitudes of the parameters w_u 's and w_i 's.

Hybrid approach. Beyond estimating b_{ui} based on either the baseline estimate or the weighted features, we may combine them together to have the best of both worlds. This leads to a new prediction rule for b_{ui} :

$$b_{ui} = \mu + b_u + b_i + w_u \mu_u + w_i \mu_i \quad (7)$$

Substituting (7) into (2), we will obtain the combined model-NbModel3. To estimate w_u , w_i , b_u , b_i , we need to solve the regularized least squares problem as followings:

$$\begin{aligned} \min_{w^*, b^*} \sum_{(u,i) \in \mathbf{E}} [r_{ui} - b_{ui} - |\mathcal{N}_{(i;u)}^k|^{-\frac{1}{2}} \sum_{v \in \mathcal{N}_{(i;u)}^k} (r_{vi} - b_{vi}) w_{uv}]^2 \\ + \lambda_1 \sum_{v \in \mathcal{N}_{(i;u)}^k} w_{uv}^2 + \lambda_2 (b_u^2 + b_i^2) + \lambda_3 (w_u^2 + w_i^2) \end{aligned} \quad (8)$$

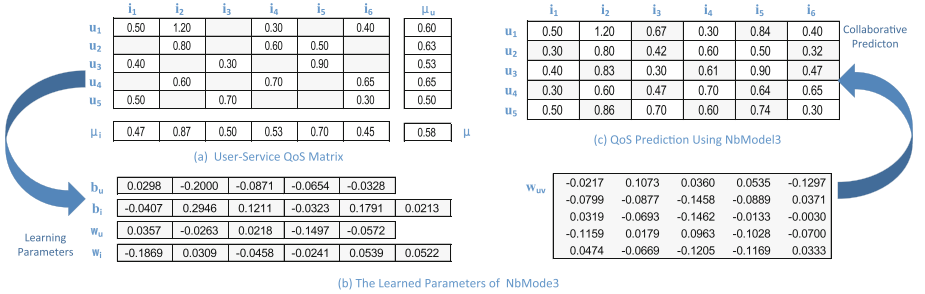


Fig. 1. An example on QoS prediction with learning NbModel3.

3.3 Models Learning

In a sense, our neighborhood models provide two-tier models for personalized QoS prediction. The first tier, b_{ui} , describes general properties of the service and the user, without accounting for any involved interactions. The second tier—“Neighborhood tier” contributes fine grained adjustments that are hard to profile. Model parameters are determined by minimizing the associated regularized squared error function through gradient descent. Recall that $e_{ui} \triangleq r_{ui} - \hat{r}_{ui}$. We loop over all known scores in \mathbb{E} . For a given training case r_{ui} , we modify the parameters by moving in the opposite direction of the gradient, yielding:

$$\begin{aligned} - b_u &\leftarrow b_u + \gamma_1 (e_{ui} - \lambda_2 b_u) \\ - b_i &\leftarrow b_i + \gamma_1 (e_{ui} - \lambda_2 b_i) \end{aligned}$$

$$\begin{aligned}
& - w_u \leftarrow w_u + \gamma_1 (e_{ui}\mu_u - \lambda_3 w_u) \\
& - w_i \leftarrow w_i + \gamma_1 (e_{ui}\mu_i - \lambda_3 w_i) \\
& - \forall v \in \mathcal{N}_{(i;u)}^k: w_{uv} \leftarrow w_{uv} + \gamma_2 \left(|\mathcal{N}_{(i;u)}^k|^{-\frac{1}{2}} e_{ui} (r_{vi} - b_{vi}) - \lambda_1 w_{uv} \right)
\end{aligned}$$

Note that, update rules set forth can fit all of the least squares problems in Formulas (4), (6) and (8). When assessing the method on a given dataset, we took advantage of following values for the meta parameters: $\lambda_1 = \lambda_2 = \lambda_3 = 0.001$, $\gamma_1 = \gamma_2 = 0.001$. It is beneficial to decreasing step sizes (the γ 's) by a factor of 0.9 after each iteration. For another parameter k , our experiments demonstrate that increasing k always benefits the RMSE. Hence, the choice of k should reflect a tradeoff between prediction accuracy and computational cost. A toy-example on QoS prediction using NbModel3 is given in Fig. 1, where we let $k = 5$.

4 Experiments

4.1 Datasets

To evaluate the QoS prediction performance, we use a large-scale dataset collected by Zheng et al. [2]. The dataset consists of a total of 1, 974, 675 real-world web service invocation results are collected from 339 users on 5, 825 real-world web services [2]. This dataset can be considered as a set of usage data for real-world cloud services from distributed locations. In our experiments, we only think the response time (the range scale is 0-20s). However, the proposed approach can be applied to additional QoS properties easily.

4.2 Evaluation Metrics

Mean absolute error (MAE) and root mean squared error (RMSE) metrics, two basic statistical accuracy metrics [16], have been extensively used in performance evaluation of rating predictions [15], are used to measure the QoS-prediction performance of selected methods. MAE and RMSE are defined in (9), where r_{ui} is the observed QoS value, \hat{r}_{ui} is the predicted one, N is the number of test cases. The MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction [16]. The RMSE is a quadratic scoring rule which measures the average magnitude of the error.

$$MAE = \sum_{u,i} |r_{ui} - \hat{r}_{ui}| / N \quad RMSE = \sqrt{\sum_{u,i} (r_{ui} - \hat{r}_{ui})^2 / N} \quad (9)$$

4.3 Comparison

To show the prediction accuracy of our neighborhood-based approaches, we compare our methods with the three kinds of popular approaches:

1. *Statistical approaches.* *GMEAN* takes the average QoS value of the whole dataset as the predictive QoS value of user u to service i , i.e. $\hat{r}_{ui} = \mu$; *UMEAN* takes the average QoS value known by u as the predictive QoS value of u to i , i.e., $\hat{r}_{ui} = \mu_u$; *IMEAN* takes the average QoS value observed from i as the predictive QoS value of u to i , i.e. $\hat{r}_{ui} = \mu_i$;
2. *Heuristic-based CF.* *UPCC* is user-based collaborative prediction model. Top-k neighbors of users are found using PCC-based similarity [3]; *IPCC* is item-based collaborative prediction model. Top-k neighbors of items (services) are found using PCC-based similarity [4]; *UIPCC* combines the user-based and item-based collaborative prediction approaches and employs both the similar users and similar services for the QoS value prediction [4].
3. *MF-based approaches.* *PMF* uses probabilistic matrix factorization [17] to factorize user-service QoS matrix for the prediction [2]; *NMF* uses non-negative matrix factorization [18] to factorize the QoS matrix into two matrices p and q , with the property that all three matrices have no negative elements; *BiasedMF* exploits a combination of baseline estimate (same to (3)) and matrix factorization prediction rule for collaborative filtering [15]. We adopt it for the QoS prediction.

Table 1. Performance comparisons of QoS prediction models using different matrix density, where ‘a’ and ‘b’ indicate 1st-class and 2nd-class, respectively.

Methods	MD=0.5 %		MD=1 %		MD=5 %		MD=10 %	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GMEAN	0.9721	1.9735	0.9915	1.9726	0.9860	1.9740	0.9920	1.9725
UMEAN	0.8873	1.8914	0.8899	1.8705	0.8746	1.8600	0.8744	1.8575
IMEAN	0.8317	1.9326	0.7870	1.8136	0.7002	1.5746	0.6890	1.5410
UPCC	0.9709	1.9727	0.9468	1.9457	0.6173	1.3925 ^b	0.5446	1.3119
IPCC	0.9721	1.9735	0.9888	1.9716	0.6675	1.4272	0.6430	1.3798
UIPCC	0.9708	1.9725	0.9453	1.9430	0.6162	1.3900 ^b	0.5439	1.3102
PMF	0.8317	2.0624	0.8195	2.0010	0.6354	1.5071	0.5541	1.3393
NMF	0.7656 ^a	1.8480	0.7092 ^a	1.7650	0.6516	1.5145	0.6461	1.4527
BiasedMF	0.8052	1.7643	0.7832	1.7128	0.6376	1.4361	0.5518	1.3088
NbModel1	0.7986	1.6989 ^a	0.7471	1.6163 ^a	0.6106	1.5333	0.5336	1.3563
NbModel2	0.7678 ^a	1.7942	0.7196 ^b	1.6436 ^b	0.5817 ^b	1.3943 ^b	0.5156 ^b	1.2832 ^b
NbModel3	0.7838 ^b	1.7847 ^b	0.7428	1.6582	0.5793 ^a	1.3795 ^a	0.5138 ^a	1.2739 ^a

For the CF-based methods, we choose the neighborhood size of users at $k = 10$ and services at $k = 50$. For the MF-based methods, the regularization parameters for user and service are set at $\lambda_u = \lambda_v = 0.001$, and the dimensionality of latent factors is fixed at 10. For all selected methods, we use their implementations in LibRec (<http://www.librec.net/index.html>). Note that, for a fair comparison of selected methods, all of them exploit only the information

supplied by the user-service QoS matrix, and no additional information (e.g., geo-locations of users) is allowed. To examine the impact of data sparsity in experiments, we randomly remove entries from the user-item matrix with different density, specially, we take 0.5% – 1% for the case of sparse data and 5% – 10% for the case of dense data. For instance, MD (Matrix Density)=0.5% means that we randomly select 0.5% of the QoS entries to predict the remaining 99.5% of QoS entries. The original QoS values of the removed entries are used as the expected values to study the prediction accuracy. All selected methods and neighborhood-based variants, NbModel1, NbModel2 and NbModel3, are used to forecast the QoS values of the removed entries. The experimental results are shown in Table 1.

Depending on Table 1, NbModel3 and NbModel2 respectively rank the first class and the second class on both MAE and RMSE in the case of dense data. Both models obtain smaller MAE and RMSE values consistently for response-time with MD=5% and MD=10%. MAE and RMSE values of neighborhood models become smaller, since denser matrix provides more information for the missing value prediction. In the case of sparse data, NMF achieve best performance in term of MAE followed by three neighborhood models. However, our methods perform much better than all other counterparts on RMSE. Among all the prediction models, our methods achieve better performance on both MAE and RMSE, telling that learning neighborhood model can achieve higher prediction accuracy. Also, neighborhood-based models preserve the explainability of memory-based CF, and enable to give users a reason for their predictions.

4.4 Impact of Top-K

To examine the influence of top-k neighbors selection on our prediction models, we distinguish from two cases: sparse data and dense data. With sparse data, we found that increasing k value cannot lead to significant performance improvements and sometimes we may experience decreased performance. There are two reasons for this. One is that sparser matrix cannot offer the “neighborhood tier” more information to contribute fine-grained adjustments. The other is that the leaned component b_{ui} (sees (2)) has given ideal predictions. Nevertheless, “neighborhood ties” can be invoked as a regularization component to

Table 2. Performance of neighborhood-based models in case of sparse data.

Methods	k	MD=0.5 %		MD=1 %	
		MAE	RMSE	MAE	RMSE
NbModel1	0	0.7987	1.6990	0.7411	1.5770
	80	0.7986	1.6989	0.7471	1.6163
NbModel2	0	0.7678	1.7942	0.7428	1.6567
	80	0.7678	1.7942	0.7196	1.6436
NbModel3	0	0.7840	1.7848	0.7409	1.6474
	80	0.7838	1.7847	0.7428	1.6582

avoid the overfitting of baseline predictor even if the usage data are sparse. The experimental results are shown in Table 2, where $\hat{r}_{ui} = b_{ui}$ if $k = 0$.

In the case of condensed data, we conduct experiments to see the impact of top-K similar users based on NbModel3. The experimental results are shown in Fig. 2. From Fig. 2, we find that the RMSE consistently decreases as increasing the value of k with different matrix density (range from 5 % to 15 %). While the value distribution of MAE presents U-shaped curve, and the best configuration for our dataset is about $k = 80$. In addition, more gains can be noted when the matrix becomes denser as for neighborhood-based models. For the neighborhood models, since the computational cost always increases with the increment of K , the choice of k should reflect a tradeoff between prediction accuracy and computational cost.

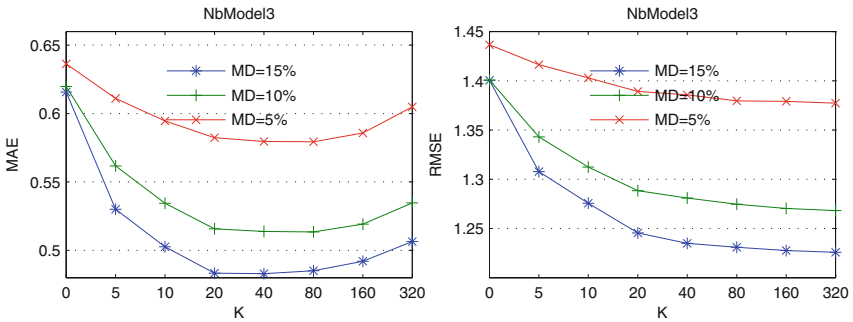


Fig. 2. Performance of NbModel3 with different k and matrix density.

5 Conclusion and Future Works

Founded on the principles of collaborative filtering and machine learning, we propose a neighborhood-based framework for making personalized QoS-prediction of cloud services. It provides an efficient global optimization scheme, thus offers robust and accurate prediction results. Also, it preserves explainability for QoS-prediction tasks which would be helpful for users make more definite selections. The extensive experimental analysis indicates the effectiveness of our approach.

Since neighborhood-based models are distinctly of MF, we would want to integrate them together to have both of worlds in the future. We currently conduct experimental studies only on response-time, thus expect to adapt the proposed methods to the prediction tasks of other cloud service QoS properties, such as reliability, throughput and scalability. In addition, extra information, such geo-information of users and items, temporal use of service invocation, can enter into this framework to offer more accurate prediction results.

Acknowledgements. This work is supported by the Special Funds for Middle-aged and Young Core Instructor Training Program of Yunnan University, the Applied Basic Research Project of Yunnan Province (2013FB009,2014FA023), the Program for Innovative Research Team in Yunnan University (XT412011), and the National Natural Science Foundation of China (61562090,61562092,61472345).

References

1. Sun, L., Dong, H., Hussain, F.K., Hussain, O.K., Chang, E.: Cloud service selection: state-of-the-art and future research directions. *J. Netw. Comput. Appl.* **45**, 134–150 (2014)
2. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. *IEEE T. Serv. Comput.* **6**(3), 289–299 (2013)
3. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS prediction for Web services via collaborative filtering. In: 2007 IEEE International Conference on Web Services, ICWS 2007, pp. 439–446. IEEE Press, New York (2007)
4. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware Web service recommendation by collaborative filtering. *IEEE T. Serv. Comput.* **4**(2), 140–152 (2011)
5. Sun, H., Zheng, Z., Chen, J., Lyu, M.R.: Personalized Web service recommendation via normal recovery collaborative filtering. *IEEE T. Serv. Comput.* **6**(4), 573–579 (2013)
6. Wu, J., Chen, L., Feng, Y., Zheng, Z., Zhou, M., Wu, Z.: Predicting quality of service for selection by neighborhood-based collaborative filtering. *IEEE Trans. Syst. Man Cybern.: Syst.* **43**(2), 428–439 (2013)
7. Chen, X., Zheng, Z., Yu, Q., Lyu, M.R.: Web service recommendation via exploiting location and QoS information. *IEEE Trans. Parallel Distrib. Syst.* **25**(7), 1913–1924 (2014)
8. Yu, Q., Zheng, Z., Wang, H.: Trace norm regularized matrix factorization for service recommendation. In: 2013 IEEE 20th International Conference on WebServices, ICWS 2013, pp. 34–41. IEEE Press, New York (2013)
9. Zhang, R., Li, C., Sun, H., Wang, Y., Huai, J.: Quality of web service prediction by collective matrix factorization. In: 2014 IEEE 11th International Conference on Services Computing, SCC 2014, pp. 432–439. IEEE Press, New York (2014)
10. Yin, J., Lo, W., Deng, S., Li, Y., Wu, Z., Xiong, N.: Colbar: a collaborative location-based regularization framework for QoS prediction. *Inf. Sci.* **265**, 68–84 (2014)
11. He, P., Zhu, J., Zheng, Z., Xu, J., Lyu, M.R.: Location-based hierarchical matrix factorization for Web service recommendation. In: 2014 IEEE International Conference on Web Services, ICWS 2014, pp. 297–304. IEEE Press, New York (2014)
12. Lo, W., Yin, J., Li, Y., Wu, Z.: Efficient Web service QoS prediction using local neighborhood matrix factorization. *Eng. Appl. AI* **38**, 14–23 (2015)
13. Tsai, W., Zhou, X., Chen, Y., Bai, X.: On testing and evaluating service-oriented software. *IEEE Comput.* **41**(8), 40–46 (2008)
14. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 107–144. Springer, US (2011)
15. Koren, Y.: Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* **4**(1), 1 (2010)
16. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *Int. J. Forecast.* **22**(4), 679–688 (2006)

17. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: Advances in Neural Information Processing Systems 20, NIPS 2007, pp. 1257–1264. Curran Associates, Inc. (2008)
18. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems 13, NIPS 2000, pp. 556–562. MIT Press (2000)