IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2018, accepted November 26, 2018, date of publication November 29, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883939

Time-Aware QoS Prediction for Cloud Service Recommendation Based on Matrix Factorization

SHUN LI[1], JUNHAO WEN[1], FENGJI LUO[2], (Member, IEEE), AND GIANLUCA RANZI[2], (Member, IEEE)

[1]School of Big Data and Software Engineering, Chongqing University, Chongqing 400044, China
[2]School of Civil Engineering, The University of Sydney, Sydney NSW 2006, Australia

Corresponding author: Junhao Wen (jhwen@cqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 6167060382, in part by the Australian Research Council Future Fellowship Scheme under Grant FT140100130, and in part by the China Scholarship Council under Grant 201706050085.

ABSTRACT Prediction of quality of service (QoS) is a critical area of research for cloud service recommendation. The disadvantage of QoS values is that they are directly related to time series of service status and network condition and thus instantly vary over time. The main contribution of this paper is to consider service invocation time as a dynamic factor in the collaborative filtering model and recommend high-quality services for target user. In particular, this paper proposes a time-aware matrix factorization (TMF) model that integrates QoS time series to provide two-phase QoS predictions for cloud service recommendation. The TMF model uses an adaptive matrix factorization model on a sparse QoS dataset to predict the missing QoS values. A temporal smoothing method is then developed and applied to the predicted result to perform the time-varying QoS prediction that accounts for the dependence of QoS values at different time intervals. The numerical experiments presented are conducted to validate the accuracy of the proposed method on a public QoS dataset.

INDEX TERMS Could service, recommender system, QoS prediction, time-aware, matrix factorization.

I. INTRODUCTION

As a distributed computing paradigm, cloud computing has attracted wide attentions in both academia and industry [1]. Cloud computing offers highly-distributed platforms to deliver high-quality and advanced services to users. State-of-the-art platforms, such as Microsoft Azure, Google App Engine, and Amazon EC2, are currently serving thousands of commercial companies [2]–[4]. In addition to commercial cloud platforms, many open-source cloud systems have also been developed and are widely used, such as Apache Hadoop and Spark [5]. In cloud computing, the available resources (e.g. software, hardware, storages, and application development tools) at the cloud server side are encapsulated as services and delivered to end users over the Internet. With the growing number of resources supported through cloud platforms, the cloud marketplace systems can coordinately manage cloud services, service providers, and cloud users [6].

The number of cloud services available nowadays is very large. The complexity of the cloud services is increasing due to adoption of advanced service development and service composition techniques [7]–[9]. For example, there are over 19,380 APIs available in the Programmable Web platform [10], a famous Web service repository for public search. As a result, users are facing the challenge of having to select the most appropriate services from a large number of candidates. This is referred to as the information overload problem. Fig. 1 depicts a service invocation scenario, where an end user is confused by a large number of functionally similar services in a cloud marketplace. Considering the fact that many services perform equivalent or similar functions while differing in the non-functional properties, service recommendation methods can be developed and implemented to filter and help users to find the most satisfactory available services among candidates with similar functions [11]–[13]. Specifically, service recommendation techniques based on Quality of Service (QoS) prediction have been applied in cloud service marketplaces to address the information overload problem [14]–[16].

QoS values (e.g. invocation response time, failure probability, and throughput rate) are usually different at different invocation time instants, because the Internet communication

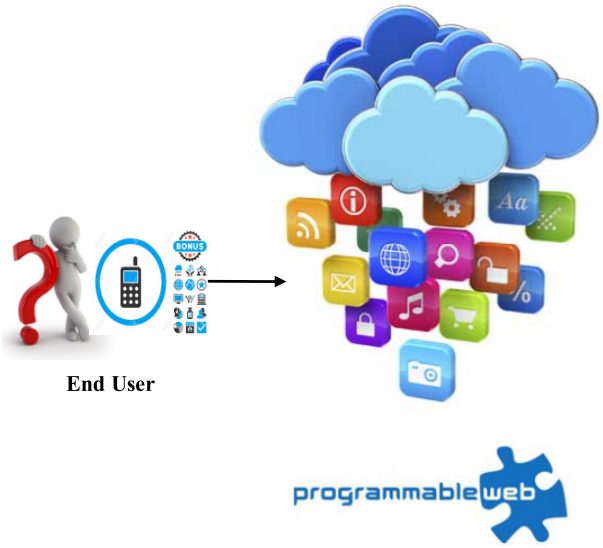

FIGURE 1. Service invocation scenario in cloud service marketplace.

condition between server and client constantly varies. Such a time-varying scenario could benefit from the development of time-aware cloud service recommendation approaches that integrate QoS information predictions. However, users often invoke specific services for limited times within a certain period, therefore leading to scarce service invocation records. Since the time-related information is not sufficient, it would be not appropriate to apply time series prediction methods to QoS prediction. In this sense, Collaborative Filtering (CF) methods, which are suitable for handling sparse datasets, are applied to time-aware service recommendation [17].

CF is one of the most prevailing prediction approaches among recommender systems. By measuring the similarity of historical QoS invocation records between candidate users and the target user, CF can deliver personalized service recommendation with QoS information. However, existing research on CF based QoS prediction [18], [19] does not seem to consider temporal characters of QoS and neglects the fact that QoS values of currently invoked cloud services are closely correlated with QoS values of previous time instants.

To tackle the aforementioned limitation, this paper proposes a time-aware cloud service recommendation approach, based on a matrix factorization model that makes full use of QoS records. The proposed method relies on a two-phase prediction approach. In particular, we first apply an adaptive matrix factorization method on a sparse, 3-dimensional matrix to make QoS prediction. This is then followed by the application of a curve smoothing method to smooth the predicted QoS curves generated in the first phase, which can improve the QoS prediction accuracy. The experiments and comparison studies presented are conducted based on a real QoS dataset to validate the proposed approach.

The remainder of this paper is organized as follows: Related work of service recommendation are presented in Section II; basic principles of the proposed TMF are introduce in Section III; the proposed TMF approach is presented in Section IV; experiments results are reported in Section V; conclusions and future works are discussed in Section VI.

II. RELATED WORK

A. COLLABORATIVE FILTERING

Collaborative filtering has been proved to be an effective solution to address the information overload problem in recommender systems. It has been firstly employed in the Amazon online sales' system [20]. Since then, CF has been widely studied and applied to different domains of people's daily lives, such as E-commercial, online multi-media, and medical care.

Shepstone *et al.* [21] extend the basic collaborative filtering model to include a closed-set speaker identification. The extended model can generate log-likelihood scores, which can reduce the impact of unwanted ratings in the rating matrix. The work in [21] is tested on a public film dataset and demonstrated to be effective when compared to prior collaborative filtering methods. Xie *et al.* [22] specify a threshold to filter inaccurate similarities in CF model, and then validate their approach in the MapReduce framework of a cloud computing platform. Zheng *et al.* [23] employ CF in Web service computing with historical QoS values. Their work shows that CF could accurately make QoS predictions in real-word Web service invocation scenarios.

B. CLOUD SERVICE RECOMMENDATION

To address the information overload problem caused by the rapid expansion of cloud services, recommender systems are currently being integrated into the cloud computing paradigm. This trend has been supported by the efforts of both academia and industry, and main contributions are summarized as below.

In [24], Google describes the work mechanism of YouTube's recommender system, which is based on the TensorFlow and uses distributed training on a CF model to generate recommended videos for the target user. Djiroun *et al.* [25] proposed a service recommendation framework in cloud service marketplace based on service content and users' behaviors identified by automatic learning technique. However, cloud services are quality-sensitive due to the unpredicted Internet environment, and the method in [25] does not consider the quality of service and neglects the impact of QoS on service recommendation.

Han *et al.* [26] introduce network QoS into cloud service computing and test their approach on a virtual machine platform to recommend cloud services to end users. Wang *et al.* [27] analyze the QoS diversity and compute the similarity of tenants' QoS information to make service recommendation for multi-tenant SaaS. Zheng *et al.* [28] employ the spearman coefficient on QoS similarity computing in the CF model to predict QoS values and the ranking of cloud services. However, the above methods [23], [28] only use QoS information as an important attribute to do service recommendation, and neglect the fact that QoS values are time sensitive and variable.

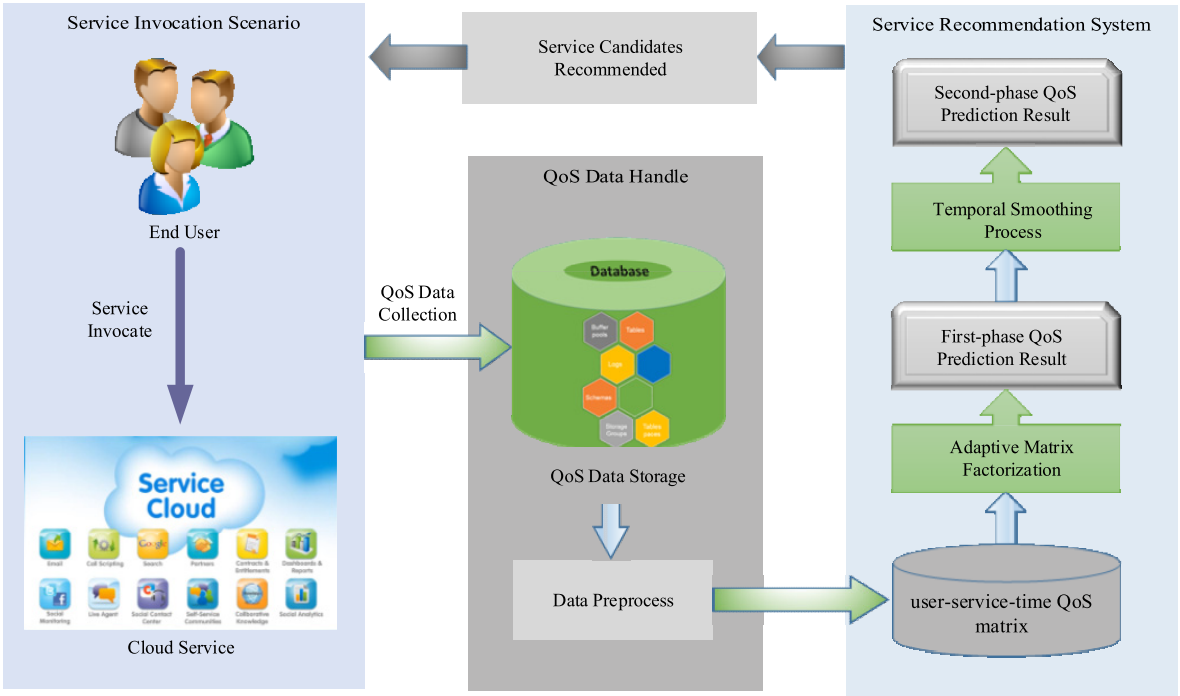

FIGURE 2. Time-aware cloud service recommendation framework.

C. TIME-AWARE QOS PREDICTION

QoS value is affected by the Internet condition and thus constantly varies over time. In this context, time-varying QoS prediction should be considered for cloud service recommendation.

Zhang *et al.* [29] propose a personalized service recommendation approach based on historical QoS invocation records taken over the period considered. They develop a time-aware QoS prediction method with a tensor factorization model in a user-service-time based three-dimensional matrix. Zhang *et al.* [30] extend the work in [29] by developing a non-negative tensor factorization model to analyze the triadic relation of user, service, and invocation time. However, these CF based, time-aware QoS prediction methods [29], [30] do not consider the fact that QoS values of a target user at a specific time interval will not only be affected by other similar users' QoS values, but also by the QoS values of previous time intervals.

Different from the previous time-varying QoS prediction researches [29], [30], we consider the impact of time series in cloud service recommendation. In this approach, we firstly use adaptive matrix factorization to predict missing QoS values, and then use time series to smooth the predicted curve. As shown in Section V, the proposed method can achieve higher cloud service prediction accuracy.

III. FUNDAMENTAL PRINCIPLES

In this section, we present the principles of the proposed TMF and explain the reason for using time smoothing method into QoS prediction.

A. PRINCIPLES OF TMF

Previous time-aware service recommendation approaches (see [29], [30]) only employ CF on the QoS historical data analysis, and do not consider the impact of time series. Based on the realization that QoS values are time-coupled, we propose a two-phase QoS prediction framework that can be described by the schematic shown in Fig. 2. Adaptive matrix factorization is firstly used to predict missing QoS values from the original sparse dataset. A temporal smoothing method is then used on the predicted QoS values to produce the final QoS prediction.

As shown in Fig. 2, when the end users invoke specific cloud services, the invocation records that contain QoS information for different temporal intervals are collected as a sparse dataset. The workflow of the proposed TMF can be subdivided in the following steps:

1) End users provide QoS data with temporal series information by invoking cloud services on the cloud platform;
2) The original QoS data is preprocessed by a logistic transformation method to avoid overlarge fluctuation of collected QoS values;
3) An adaptive matrix factorization model is applied on the user-service-time matrix to convert QoS predictions for the cloud service;
4) Apply the temporal smoothing approach on the QoS values predicted in 3) and generate the final recommendation results.
5) High-quality services are selected from the recommendation results and recommended to the target user.

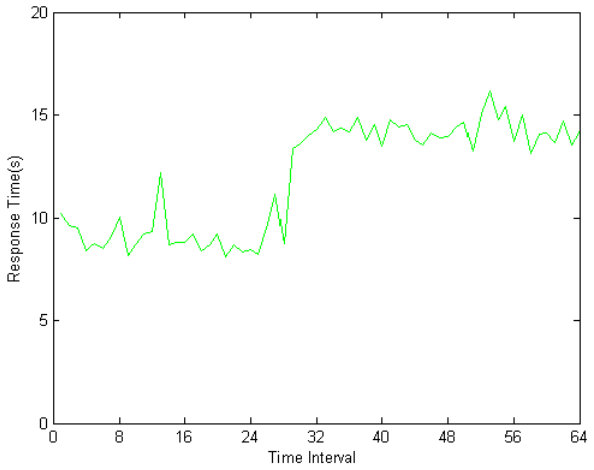

FIGURE 3. Response-Time Invocation over the whole time interval.

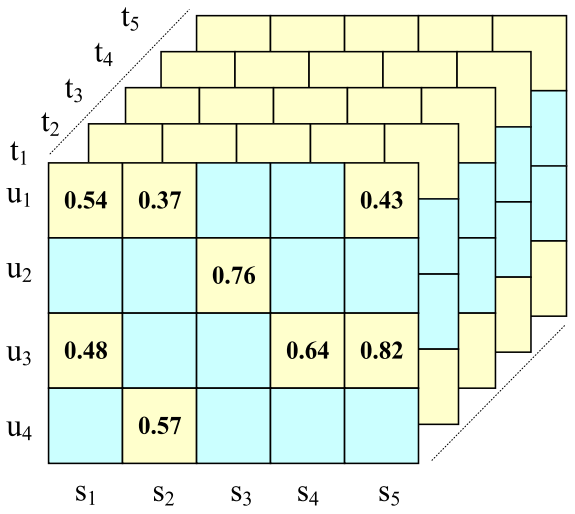

FIGURE 4. Observed User-Service-Time QoS data in real cloud service invocation.

B. REASON FOR USING TIME SMOOTHING METHOD

Based on the realizations of: (1) QoS values are time-varying, and; (2) the QoS values at one time interval should be similar to those at close time intervals, we analyze QoS values of the original dataset, i.e. a public QoS dataset [29]. Fig. 3 demonstrates how the response time changes over time when some users invoke a specific cloud service.

As shown in Fig. 3, it can be seen that the QoS value at time instant t is similar with the corresponding values at instants $t-1$ and $t+1$, even for time intervals in which the QoS values highly fluctuate. This indicates that the QoS values at neighboring time intervals significantly influence each other, and this aspect should be considered in the QoS prediction. Under these conditions, the temporal smoothing method is expected to effectively smooth out the predicted QoS values.

IV. TIME-AWARE QOS PREDICTION BASED ON MF

In this section, we present the definition of the time-dependent QoS problem followed by the description of the adaptive matrix factorization model and of the temporal smoothing method.

A. PROBLEM DEFINITION

The collected QoS dataset can be represented by a three-dimensional matrix (Fig. 4). Each entry in the matrix represents a QoS value of a cloud service invoked from a specific user during a specific time interval. The matrix is sparse, because users only invoke a specific set of cloud services during a specific time interval.

The focus of this work is to define a procedure to predict unknown QoS values in the aforementioned sparse three-dimensional matrix, as these missing QoS values would significantly affect the cloud service recommendation performance. More precisely, the problem is defined as described below.

Suppose m users invoked n cloud services over k time interval. Denote U as the user set, S as the set of cloud services, and T as the total number of time intervals. The three-dimensional matrix can be considered to comprise $|T|$ two-dimensional matrices $R \in \mathbb{R}^{m\times n}$. Each entry of the tth decomposed two-dimensional matrix is denoted as $R_{ij}(t) \in \mathbb{R}^{m\times n}$, $t = 1, lT$, representing the QoS value of the cloud service S_j invoked by the user U_i at time t. We use indicators $I_{ij}(t) = 0$ and $I_{ij}(t) = 1$ to denote the missing and the observed QoS values of $R_{ij}(t)$, respectively. The missing QoS values $\{R_{ij}(t)|I_{ij}(t) = 0\}$ are then predicted from the observed values $\{R_{ij}(t)|I_{ij}(t) = 1\}$.

Statistically, the QoS values observed in Fig. 3 vary within the range of [0, 20]. Without loss of generality, we map the range to a smaller range of [0, 1] using the following logistic function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

B. ADAPTIVE MF IN QOS PREDICTION

Matrix factorization is a widely-adopted method that predicts missing values in the rating matrix of recommender systems. The underlying rationale of matrix factorization is to initiate two low-rank matrices and iteratively update them to approximate the original sparse matrix. The computational complexity is reduced by decomposing the high-rank original matrix into low-rank matrices. The two low-rank matrices represent the latent features of the user and service, respectively.

Based on the definition given in Section III, the approximation of the QoS matrix can be calculated as the product of two low-rank matrices:

$$R_{ij} \approx U_i S_j^T = \sum_{p=1}^{d} u_{ip} s_{jp} \tag{2}$$

where $U_i = [u_{ip}]_{m\times d}$ denotes the latent factors of user U_i, $S_j = [s_{jp}]_{n\times d}$ represents the latent factors of service S_j, and parameter d ($d \ll \min(m, n)$) denotes the number of latent factors.

When the cloud service is invoked by the cloud user, QoS attributes of the cloud service are associated to some service-specific factors (e.g. server condition and service load). In order to analyze the impact of these factors, the bias of the cloud service S_j on the QoS value is considered into Eq. (1) as a constraint term v_j. It is worth pointing out that also user-specific factors, e.g. geographical information, computer capability, can affect the QoS values. We denote q_i as the bias of the user U_i. By considering these factors, an improved approximation of the original QoS matrix can be formulated as:

$$\hat{R}_{ij} = q_i + v_j + \sum_{p=1}^{d} u_{ip} s_{jp} \tag{3}$$

To estimate the missing values of the $|T|$ two-dimensional matrices introduced in Section 3.1, Eq. (3) is reformulated as:

$$\hat{R}_{ij}(t) = q_i(t) + v_j(t) + \sum_{p=1}^{d} u_{ip}(t) s_{jp}(t) \tag{4}$$

where $\hat{R}_{ij}(t)$ denotes the predicted QoS values and the remaining parameters, i.e. $q_i(t)$, $v_j(t)$, $u_{ip}(t)$, and $s_{jp}(t)$, need to be determined as outlined below. Once these four parameters are identified, the approximation value $\hat{R}_{ij}(t)$ is taken as the prediction result of $R_{ij}(t)$.In order to enhance the prediction accuracy, we introduce implicit feedback computed by the identity of the items users explicitly rate into prediction model and then (4) can be reformed as following:

$$\widehat{R}_{ij}(t) = q_i(t) + v_j(t) + \sum_{p=1}^{d} \left(u_{ip}(t) + \sum_{h \in I_i} \frac{y_{hp}(t)}{\sqrt{|I_i|}} \right) s_{jp}(t) \tag{5}$$

where I_i denotes the set of indices of the services invoked by user U_i. $\sum_{h \in I_i} \frac{y_{hp}(t)}{\sqrt{|I_i|}}$ here is used to adjust user's latent factor based on his/her implicit feedback $y_{hp}(t)$, which can be learned in our method.

We then estimate the unknown values of the $|T|$ matrices by solving the optimization model (5), which minimizes the sum of the error between $R_{ij}(t)$ and the normalized value of $\hat{R}_{ij}(t)$.

$$\begin{aligned} \min \Delta\varphi = \sum_{t=1}^{k} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(t) \Bigg(& \frac{1}{2} \left(R_{ij}(t) - \hat{R}_{ij}(t) \right)^2 \\ & + \frac{\lambda}{2} \Big(\sum_{p=1}^{d} \Big(\sum_{i=1}^{m} u_{ip}(t)^2 + \sum_{j=1}^{n} s_{jp}(t)^2 + \sum_{j=1}^{n} y_{jp}(t)^2 \Big) \\ & + \sum_{i=1}^{m} q_{ui}(t)^2 + \sum_{j=1}^{n} q_{sj}(t)^2 \Big) \Bigg) \end{aligned} \tag{6}$$

where λ is a parameter controlling the relative importance of the regularization terms. Eq. (6) is the objective function of the matrix factorization that minimizes the squared error between the predicted and actual data.

To solve model (6), parameters $q_i(t)$, $v_j(t)$, $u_{ip}(t)$, $s_{jp}(t)$ and $y_{hp}(t)$ need to be continuously updated. In this study, stochastic gradient descent [34] is used to train the TMF model to estimate the unknown parameters. Values of the parameters are randomly initialized at the beginning, and then iteratively updated as follows:

$$u_{ip}(t) \leftarrow u_{ip}(t) - \alpha \Big(\sum_{j=1}^{n} e_{ij}(t) \cdot s_{jp}(t) + \lambda u_{ip}(t) \Big) \tag{7}$$

$$s_{jp}(t) \leftarrow s_{jp}(t) - \alpha \Big(\sum_{i=1}^{m} e_{ij}(t) \cdot \left[u_{ip}(t) + \sum_{h \in I_i} \frac{y_{hp}(t)}{\sqrt{|I_i|}} \right] + \lambda s_{jp}(t) \Big) \tag{8}$$

$$y_{hp}(t) \leftarrow y_{hp}(t) - \alpha \Big(\sum_{j=1}^{n} \frac{e_{ij}(t) \cdot s_{jp}(t)}{\sqrt{|I_i|}} + \lambda_{hp}(t) \Big) \tag{9}$$

$$q_i(t) \leftarrow q_i(t) - \alpha \Big(\sum_{j=1}^{n} e_{ij}(t) + \lambda q_i(t) \Big) \tag{10}$$

$$v_j(t) \leftarrow v_j(t) - \alpha \Big(\sum_{i=1}^{m} e_{ij}(t) + \lambda v_j(t) \Big) \tag{11}$$

where $e_{ij}(t) = R_{ij}(t) - \hat{R}_{ij}(t)$ denotes the difference between predicted value and observed value and $\alpha > 0$ is set as the learning rate to control the step size of each iteration.

C. TEMPORAL SERIES SMOOTHING

To take into account the dependency of QoS values between neighboring time intervals, we apply a temporal smoothing method [36] on the prediction results generated in the first phase, expressed as:

$$R_{ij}(t) = \begin{cases} \dfrac{2R_{ij}(t) + R_{ij}(t+1)}{3}, & t = 1 \\ \dfrac{2R_{ij}(t) + R_{ij}(t-1)}{3} & t = |T| \\ \dfrac{R_{ij}(t-1) + 2R_{ij}(t) + R_{ij}(t+1)}{4} & otherwise \end{cases} \tag{12}$$

V. EXPERIMENTS

A. EXPERIMENT SETUP

In this part, our proposed approach is validated against the values of the Response Time (RT), one of the most important QoS attributes. RT measures the time duration of the user to get response from the cloud service. A series of experiments are conducted on a computer with one 3.6∼4.2 GHz Intel CPU, 32 GB RAM, and an Ubuntu operation system.

A public service QoS dataset [29] is used for the experiments. The dataset contains more than 20 million response time records, which have been produced by 142 users who invoked 4,532 services over 16 hours (64 time slices, with 15-minute collection interval). Although we only consider one QoS attribute (i.e. response time) in this study, the proposed prediction model can also be directly applied to other QoS attributes.

The whole RT dataset is randomly divided into two parts: a training dataset containing 70% data items of the RT dataset, and a testing dataset containing the remaining 30%. Based on the consideration that end users would only invoke several specific services at a certain time, we randomly remove some data items from the dataset to simulate this real-world service scenario.

B. EVALUATION METRICS

The prediction accuracy of the proposed TMF model is compared with other methods. For this purpose, two widely adopted statistical metrics are used, i.e. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are defined as:

$$MAE = \frac{\sum_{i,j,t} \left| R_{ij}(t) - \hat{R}_{ij}(t) \right|}{N} \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i,j,t} \left| R_{ij}(t) - \hat{R}_{ij}(t) \right|^2}{N}} \tag{14}$$

where $R_{ij}(t)$ is the observed QoS values of the user U_i on cloud service S_j at time slice T_t; $\hat{R}_{ij}(t)$ denotes the predicted value of $R_{ij}(t)$; N denotes the total number of predicted data items.

C. COMPARISON STUDY

Our proposed method is compared with following recommendation approaches:

- UPCC [31]: a classical recommendation method based on the similarity between users' preferences. In service computing it is often used to make QoS prediction based on users' shared preferences;
- IPCC [32]: it is similar to UPCC, but it predicts missing QoS values based on the item similarity;
- UIPCC [33, 35]: a hybrid method that integrates both UPCC and IPCC;
- PMF [34]: a widely used model-based CF method. It is used to make predictions in the rating matrix by means of probability theory analysis;
- WSPred [29]: a method that applies a tensor model to a 3-dimensional user-service-time matrix to predict missing QoS values.

Most of the above methods [31]–[33] work on two-dimensional matrix, while the proposed TMF model deals with the 3-dimensional matrix by decomposing it into a set of 2-dimensional matrices. For comparative purposes, we calculate the averaged MAE and RSME of the two-dimensional matrices.

In real world scenarios, users are only interested in some specific cloud services at certain time slices. Thus, the observed QoS values are expected to be very sparse. To simulate this situation, we randomly remove a certain number of QoS values from the original QoS dataset. After the data preprocessing, density of the three-dimensional dataset is set to 5%, 10%, 15%, and 20%, respectively.

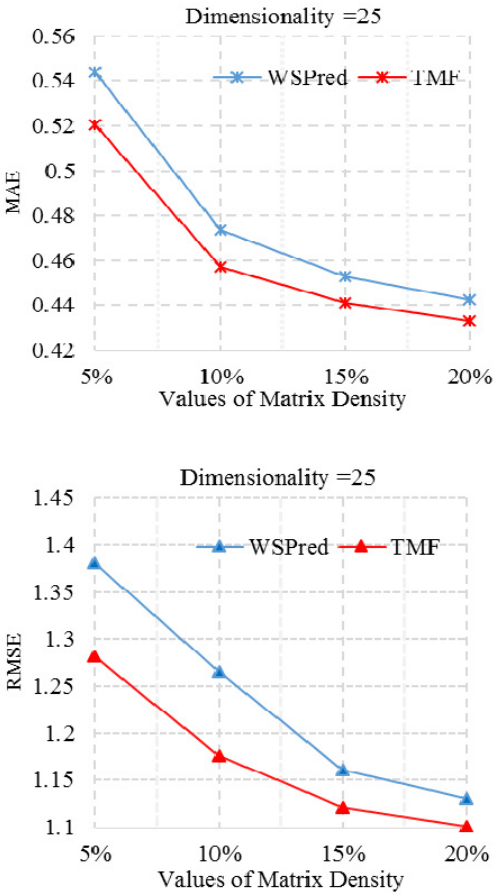

FIGURE 5. Impact of matrix density.

5% density means there are 95% missing QoS values, which are to be predicted by the TMF model. Same parameter settings are used in the above five benchmark methods and the proposed TMF method: $\lambda = 0.001$, $\alpha = 0.8$ and $d = 25$.

Results are reported in Table 1 and show how the proposed TMF method leads to smaller MAE and RSME than other methods under the same density condition. This indicates that the proposed TMF possesses a higher QoS prediction accuracy. The averaged MAE and RSME become smaller for increasing density values (from 5% to 20%). This trend reflects the fact that, when more QoS data is collected, higher prediction accuracy can be achieved. The comparison results show that by using the temporal smoothing method in the second phase prediction, higher prediction accuracy is achieved. It also indicates that the QoS value at a time interval is influenced by those at previous time intervals.

D. IMPACT OF DENSITY

The density parameter represents the sparsity degree of the QoS values in the dataset. We vary the density value from 5% to 20% and perform sensitivity study to investigate the impact of density on our TMF model. The parameter d is fixed to be 25.

Fig. 5 shows that the MAE and RMSE of both TMF and WSPred decrease when the matrix density increases,

TABLE 1. Prediction accuracy comparison between previous methods.

Methods	Matrix Density							
	5%		10%		15%		20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UIPCC	0.5961	1.3969	0.5331	1.3072	0.5041	1.2537	0.4762	1.1978
IPCC	0.6239	1.4338	0.5889	1.3551	0.5542	1.2957	0.5248	1.2139
UPCC	0.5957	1.4042	0.5536	1.3278	0.5217	1.2769	0.4963	1.2521
PMF	0.5813	1.4468	0.5154	1.2971	0.4843	1.2334	0.4609	1.1838
WSPRED	0.5441	1.3816	0.4734	1.2653	0.4529	1.1605	0.4428	1.1301
TMF	**0.5197**	**1.2818**	**0.4558**	**1.1759**	**0.4412**	**1.1214**	**0.4331**	**1.1009**

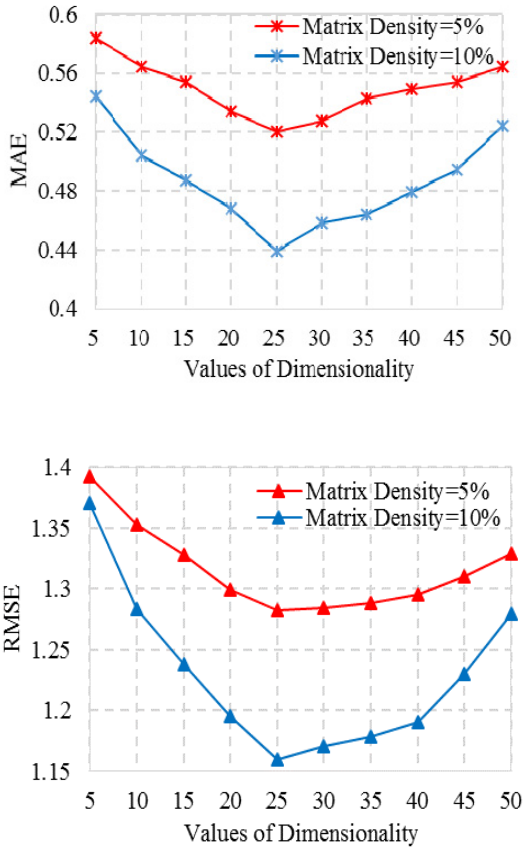

FIGURE 6. Impact of dimensionality d.

therefore indicating that, when more QoS values are observed and shared, TMF and WSPred can achieve higher prediction accuracy. TMF has consistently smaller MAE and RMSE than WSPred, reflecting the superior performance of the TMF.

E. IMPACT OF DIMENSIONALITY d

Parameter d determines the number of latent factors of the TMF model. Sensitive analysis is performed to explore the impact of parameter d in our TMP method. The matrix densities are set to be 5% and 10%.

Fig. 6 shows that the best prediction result is achieved when d is set to be 25. Both MAE and RMSE decrease at the beginning. This means when d is smaller than 25, the prediction performance can be improved with the inclusion of more latent factors. However, when the number of latent factors exceeds a certain threshold (i.e. 25 in this case), an overfitting problem would arise. This is attributed to the fact that more noisy data is introduced into the training model.

VI. CONCLUSIONS AND FUTURE WORKS

Time-aware information is critical for QoS prediction, and the latter is recognized as a key feature of service recommendation. This study proposes an efficient time-aware QoS prediction method, i.e., Time-aware Matrix Factorization model, for cloud service recommendation. This approach starts with an adaptive matrix factorization model for the real-time QoS prediction, and it then uses a temporal smoothing method on the prediction result to generate the final time-aware QoS prediction for service recommendation. An experiment has been presented that demonstrates that, when time series information is introduced into the prediction model, higher QoS prediction accuracy can be achieved and high-quality cloud services can be consequentially recommended to target users.

In this study, we only consider the impact of time series information on QoS values. In real cloud service invocations, the QoS performance would also be greatly affected by other contextual information, such as location, communication condition, and server workload. In future, more contextual factors can be considered to improve the accuracy of QoS prediction.

REFERENCES

[1] L. A. Tawalbeh, R. Mehmood, E. Benkhlifa, and H. Song, "Mobile cloud computing model and big data analysis for healthcare applications," *IEEE Access*, vol. 4, no. 99, pp. 6171–6180, 2017.
[2] P. Wiewiura, M. Malawski, and M. Piwowar, "Distributed execution of dynamically defined tasks on microsoft azure," in *Proc. 11th Int. Conf. Parallel Process. Appl. Math.*, Sep. 2015, pp. 291–301.
[3] C. Krintz, "The AppScale cloud platform: Enabling portable, scalable Web application deployment," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 72–75, Mar. 2013.
[4] A. Marathe, R. Harris, D. K. Lowenthal, B. R. de Supinski, B. Rountree, and M. Schulz, "Exploiting redundancy and application scalability for cost-effective, time-constrained execution of HPC applications on Amazon EC2," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2574–2588, Sep. 2016.
[5] B. Akil, Y. Zhou, and U. Röhm, "On the usability of Hadoop MapReduce, Apache Spark & Apache flink for data science," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2017, pp. 303–310.

[6] A. Bestavros and O. Krieger, "Toward an open cloud marketplace: Vision and first steps," *IEEE Internet Comput.*, vol. 18, no. 1, pp. 72–77, Jan. 2014.
[7] C. Lee, C. Wang, E. Kim, and S. Helal, "Blueprint flow: A declarative service composition framework for cloud applications," *IEEE Access*, vol. 5, pp. 17634–17643, 2017.
[8] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 382–393, May/Jun. 2016.
[9] I. A. Ridhawi, Y. Kotb, and Y. A. Ridhawi, "Workflow-net based service composition using mobile edge nodes," *IEEE Access*, vol. 5, pp. 23719–23735, 2017.
[10] S. Lyu, J. Liu, M. Tang, G. Kang, B. Cao, and Y. Duan, "Three-level views of the Web service network: An empirical study based on programmableWeb," in *Proc. IEEE Int. Congr. Big Data*, Oct. 2014, pp. 374–381.
[11] F. Tao, Y. LaiLi, L. Xu, and L. Zhang, "FC-PACO-RM: A parallel method for service composition optimal-selection in cloud manufacturing system," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2023–2033, Nov. 2013.
[12] M. Parhi, B. K. Pattanayak, and M. R. Patra, "A multi-agent-based framework for cloud service discovery and selection using ontology," *Service Oriented Comput. Appl.*, vol. 12, no. 2, pp. 137–154, 2018.
[13] L. Qu, Y. Wang, M. A. Orgun, L. Liu, H. Liu, and A. Bouguettaya, "CCCloud: Context-aware and credible cloud service selection based on subjective assessment and objective assessment," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 369–383, May/Jun. 2015.
[14] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.
[15] S. Meng, W. Dou, X. Zhang, and J. Chen, "KASR: A keyword-aware service recommendation method on MapReduce for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3221–3231, Dec. 2014.
[16] K. Huang, Y. Fan, and W. Tan, "Recommendation in an evolving service ecosystem based on network prediction," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 906–920, Jul. 2014.
[17] J. Li, J. Wang, Q. Sun, and A. Zhou, "Temporal influences-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE Int. Conf. Services Comput.*, Jun. 2017, pp. 471–474.
[18] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware Web service recommendation and visualization," *IEEE Trans. Services Comput.*, vol. 6, no. 1, pp. 35–47, Oct. 2013.
[19] Y. Ma, S. Wang, P. C. K. Hung, C.-H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown Web service QoS values," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 511–523, Aug. 2017.
[20] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[21] S. E. Shepstone, Z.-H. Tan, and M. S. Kristoffersen, "Using closed-set speaker identification score confidence to enhance audio-based collaborative filtering for multiple users," *IEEE Trans. Consum. Electron.*, vol. 64, no. 1, pp. 11–18, Feb. 2018.
[22] F. Xie, Z. Chen, H. Xu, X. Feng, and Q. Hou, "TST: Threshold based similarity transitivity method in collaborative filtering with cloud computing," *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 318–327, Jun. 2013.
[23] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr./Jun. 2011.
[24] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.
[25] R. Djiroun, M. A. Guessoum, K. Boukhalfa, and E. Benkhelifa, "A novel cloud services recommendation system based on automatic learning techniques," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, pp. 42–49.
[26] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh, "Efficient service recommendation system for cloud computing market," in *Proc. 2nd Int. Conf. Interact. Sci., Inf. Technol., Culture Hum.*, Nov. 2009, pp. 839–845.
[27] Y. Wang, Q. He, and Y. Yang, "QoS-aware service recommendation for multi-tenant SaaS on the cloud," in *Proc. IEEE Int. Conf. Services Comput.*, Jun./Jul. 2015, pp. 178–185.
[28] X. Zheng, L. Da Xu, and S. Chai, "Ranking-based cloud service recommendation," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 136–141.
[29] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for Web services," in *Proc. IEEE 22nd Int. Symp. Softw. Rel. Eng. (ISSRE)*, Nov./Dec. 2011, pp. 210–219.
[30] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware Web service recommendation via non-negative tensor factorization," in *Proc. 23rd Int. Conf. World Wide Web*, Apr. 2014, pp. 585–596.
[31] M. Hasan, S. Ahmed, M. A. I. Malik, and S. Ahmed, "A comprehensive approach towards user-based collaborative filtering recommender system," in *Proc. Int. Workshop Comput. Intell. (IWCI)*, Dec. 2016, pp. 159–164.
[32] F. Lu, L. Hong, and L. Changfeng, "The improvement and implementation of distributed item-based collaborative filtering algorithm on Hadoop," in *Proc. 34th Chin. Control Conf.*, Jul. 2015, pp. 9078–9083.
[33] B. Wang, J. Huang, L. Ou, and R. Wang, "A collaborative filtering algorithm fusing user-based, item-based and social networks," in *Proc. IEEE Int. Conf. Big Data*, Oct./Nov. 2015, pp. 2337–2343.
[34] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn.*, Jun. 2008, pp. 880–887.
[35] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2009, pp. 437–444.
[36] L. Chen, H. Ying, Q. Qiu, J. Wu, H. Dong, and A. Bouguettaya, "Temporal pattern based QoS prediction," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, Nov. 2016, pp. 223–237.

SHUN LI received the B.Eng. degree from the School of Software Engineering, Chongqing University, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include service computing, recommendation system, and machine learning.

JUNHAO WEN received the Ph.D. degree from Chong Qing University in 2008. He is currently the Vice Head and a Professor with the School of Software Engineering, Chongqing University. His research interests include service computing, cloud computing, and software dependable engineering. He has published more than 80 refereed journal and conference papers in these areas. He has more than 30 research and industrial grants and developed many commercial systems and software tools.

FENGJI LUO received the B.S. and M.S. degrees in software engineering from Chongqing University, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from The University of Newcastle, Australia, in 2014. He is currently an Early Career Development Fellow with the School of Civil Engineering, The University of Sydney, Australia. His research interests include the evolutionary computation, renewable energy, and personalized recommendation technique and its applications in smart grid. He has published more than 100 papers in these areas. He received the 2015 FEBE Research Excellence of The University of Newcastle, Australia and the 2016 Australia-Japan Emerging Research Leader Award.

GIANLUCA RANZI (M'16) received the degree in management and production engineering from the Politecnico di Milano, Italy, the B.E. (Hons.) from the University of Wollongong, Australia, the degree in civil engineering from Universita' Politecnica delle Marche, Italy, and the Ph.D. degree from the University of New South Wales, Australia. He is currently an ARC Future Fellow, a Professor, and the Director of the Centre for Advanced Structural Engineering, The University of Sydney, Australia. His research interests include high-performance building, building-to-grid technology, and demand-side management.