2014 IEEE International Conference on Services Computing

Personalized QoS Prediction for Web Services using Latent Factor Models

Dongjin Yu, Yu Liu
College of Computer
Hangzhou Dianzi University
Hangzhou, China
yudj@hdu.edu.cn,
liuyuctrlz@gmail.com

Yueshen Xu
School of Computer Science and Technology
Zhejiang University
Hangzhou, China
xyshzjucs@zju.edu.cn

Yuyu Yin*
College of Computer
Hangzhou Dianzi University
College of Electrical Engineering
Zhejiang University
Hangzhou, China
yinyuyu@hdu.edu.cn

***Abstract*—Recommending the suitable Web service is an important topic in today's society. The critical step is to accurately predict QoS of Web services. However, the highly sparse QoS data complicate the challenges. In the real world, since QoS delivery can be significantly affected by some dominant factors in the service environment (e.g., network delay and the location of user or service), Web services which are published by the same provider usually have the similar fundamental network environment. These factors can be leveraged for accurate QoS predictions, leading to high-quality service recommendations. In this paper, we expound how Latent Factor Models (LFM) can be utilized to predict the unknown QoS values. Meanwhile, we take the factors of provider and its country into consideration, which imply the latent physical location and network status information, as the latent neighbor for the set of Web services. Hence, the novel neighbor factor model is built to evaluate the personalized connection quality of latent neighbors for each service user. Then, we propose an integrated model based on LFM. Finally, we conduct a group of experiments on a large-scale real-world QoS dataset and the results demonstrate that our approach is effective, especially in the situation of data sparsity.**

Keywords-Web Service, QoS prediction, Latent Factor Models, SVD, data sparsity

I. INTRODUCTION

Web services have becoming one of the most important interoperable technologies to connect among heterogeneous applications across the Internet, which are published by different organizations through the standard protocols, such as WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and SOAP (Simple Object Access Protocol) [1]. Nowadays, among the large number of Web services, users could select properly functional ones that share similar or equivalent functionality under the help of Web service search engines. However, non-functional properties of Web services are little cared about [2].

As a matter of fact, non-functional properties (e.g., Quality-of-Service, QoS) are also very important for making Web service selection and recommendation [3]. Here, QoS properties may include availability, response time, throughput, etc. Recently, many researchers argue that the QoS values of Web service can not be easily acquired from the service providers or the third-party organizations. For example, due to the instability of Internet environment and the difference of service user's network infrastructure, the delivered QoS by the provider is always inconsistent for different service users [4], [5]. Therefore, it has become a great challenge to predict the QoS values accurately for personalized requirement.

Considering the huge number of Web services and the enormous cost for service users to invoke all Web services, it is infeasible to immediately acquire the QoS value to select the optimal service. To address this problem, many personalized QoS prediction approaches via Collaborative Filtering (CF) technique have been proposed in recent years [2], [3], [4], [5], [6]. CF approaches have been successfully adopted in the commercial recommended system, e.g., Amazon [7], Netflix, etc. In general, CF approaches are divided into two categories: neighborhood-based and model-based approaches [8]. The neighborhood-based approaches include two types: the user-based approach and the item-based approach. The main procedure of neighborhood-based approaches is to calculate the similarity between users (items) and select the top-k similar neighbors to help predict the unknown values. However, the neighborhood-based approaches can be easily led to failure due to the problems of data-sparsity and cold-start [9], [10]. On the other hand, the model-based approaches, such as clustering model [11], matrix factorization model [12], [13], [14], Bayesian models [15], are rich and various. They usually construct a model with some necessary parameters to be trained by prepared training dataset, which owns a certain ability of predicting the unknown values after the process of learning. For example, matrix factorization model takes advantage of inner product of two low-rank matrix to approximately fit the original matrix. Nevertheless, the model-based approaches may lose their prediction accuracy due to the absence of observed data.

As a classic matrix factorization approach, Latent Factor Models (LFM) achieved successfully on the Netflix Prize by Koren et al. [12]. In this paper, we focus on Singular Value Decomposition (SVD) approach, one of LFM approaches, in which users and items are mapped to a same latent factor space for making a feasible comparison between them. Each

978-1-4799-5066-9/14 $31.00 © 2014 IEEE
DOI 10.1109/SCC.2014.23

user or item associates with a latent feature vector, which contains low dimensions to explain the relation between latent factor and itself. More specifically, in the issue of QoS prediction, for user-service QoS matrix (e.g., Fig. 1 (b)), the latent space tries to explain QoS values automatically which are acquired from the process of services invocation issued by users.

In this paper, we propose a novel approach to QoS prediction of Web services based on LFM model and Latent Neighbor Model, which utilizes the information of Web services' latent neighbors, i.e., service provider and its country. Since QoS of Web services such as response time and throughput is highly related to the network infrastructure and the Internet connection between the service user and the provider, a set of Web services published by the same provider usually have the similar condition of fundamental network environment. Therefore, they share the same latent feature vector in our factor model, which is denoted by the latent feature vector of the certain provider in the above situation. As the same reason, we also take the country factor into consideration. The extensive experiments based on a large-scale real-world Web services QoS dataset demonstrate the effectiveness of our approach.

The contributions of this paper are summarized as follows:

(1) We propose a novel method to integrate the information of Web service' latent neighbors through sharing the latent feature vector of its provider and country, which imply the latent same condition of fundamental network environment.

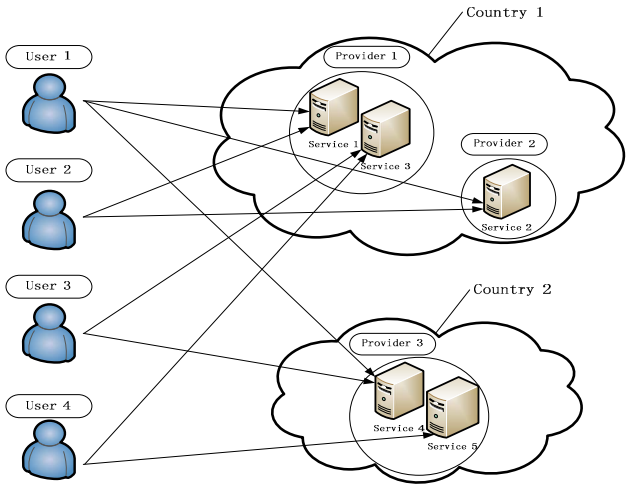
(2) We construct an integrated model LN-LFM, or Latent Neighbor-Latent Factor Models, which combines the LFM model and Latent Neighbor Model to improve the QoS value prediction accuracy.

(3) Based on a large-scale real-world Web services QoS dataset, we conduct extensive experiments to evaluate our approach and demonstrate its effectiveness.

The remainder of the paper is organized as follows. After Section II describes the problem of Web services QoS prediction, Section III reviews the related works in QoS prediction for Web services and Latent Factor Models. Section IV presents our personalized QoS value prediction model with the combination of Latent Factor Models and Latent Neighbor Model. After Section V describes our experiments in detail, Section VI concludes this paper and discusses the future work.

II. Problem Description

Fig. 1 presents a scenario of Web services invocation experienced in the real world. As shown in Fig. 1(a), there are 4 users, 5 services, 3 providers and 2 countries in total with 9 arrows which represent the invocation of services from users. Each non-empty entry in Fig. 1(b) represents the QoS invocation information (e.g., response time in this example) recorded by a user when he or she invokes a service, whereas the empty entry represents that the user has never invoked the service before. Our task is to predict the QoS values of the empty entries employing the available value.



(a) A Simple Real-world Scenario

	S1	S2	S3	S4	S5
U1	5.0	3.2		1.2	
U2	4.0	2.4			
U3			7.0	2.8	
U4			5.0		3.6

(b) User-Service Matrix

	S1	S2	S3	S4	S5
U1	5.0	3.2	**6.7**	1.2	**4.2**
U2	4.0	2.4	**4.5**	**0.8**	**4.8**
U3	**5.9**	**3.4**	7.0	2.8	**4.0**
U4	**4.8**	**2.2**	5.0	**0.9**	3.6

(c) Predicted User-Service Matrix

Figure 1. A Toy Example

As shown in Fig.1(c), a possible result can be obtained by a kind of matrix completion strategy and the bold numbers denotes the predicted QoS values.

More generally, in this problem of QoS value prediction, suppose that we use 4 sets to denote all the users, services, providers and countries:

$$US = \{u_1, u_2, \ldots, u_m\}$$

$$SS = \{s_1, s_2, \ldots, s_n\}$$

$$PS = \{p_1, p_2, \ldots, p_l\}$$

$$CS = \{c_1, c_2, \ldots, c_k\}$$

In addition, we use a $m \times n$ user-service matrix $Q = [q_{ij}]_{m \times n}$ to denote the QoS value of services experienced by users. If $q_{ij} = NULL$, it indicates user u_i has never invoked service s_j. Otherwise, it indicates the

QoS value of service s_j invoked by user u_i. Then, we define a special set:

$$T = \{(i, j, l, k) \mid q_{ij} \in Q, p_l \in PS, c_k \in CS\}$$

which contains all the known information in this problem. Based on the above information, our goal is to predict the entries which equal to *NULL* in the user-service invocation matrix accurately and efficiently.

In the real world, due to the exponentially growing number of Web services, more and more users build web applications through taking advantage of abundant Web services. Therefore, it is a huge challenge to solve this QoS prediction problem to enable effective service selection and recommendation.

III. Related Work

Web service has rapidly and increasingly become a worldwide research topic during recent years. As a result, numerous research results have been published, covering various aspects such as Web service selection [16], [17], composition [18, 19] and recommendation [2, 5, 20]. In general, Web service discovery adopt functional-based approaches, such as semantic-based approach [21], information-retrieval technique [22] and so on, to seek the indispensable services for building Web applications. However, traditional approaches of Web service discovery aim to satisfy the functional requirements of users, but not consider the performance of Web services for meeting the users' non-functional requirements [2]. To address the deficiency of traditional approaches, some QoS-based approaches have been proposed to make Web service selection more efficient [16, 5].

Recently, many QoS-based recommendation approaches have been proposed through adopting CF (Collaborative filtering) techniques. For example, Shao et al. [3] proposed a user-based CF approach for Web service QoS prediction. Zheng et al. [5] proposed neighborhood-based hybrid model, which combines user-based and item-based CF approaches to improve the QoS prediction accuracy. Due to the sparsity of Web services QoS data, nearly all CF approaches have some limitation to solve Web services QoS value prediction problem. In general, previous QoS-based approaches usually assume that the QoS data can be easily acquired, or employ some small-scale restricted data for experiment studies. Zheng et al. released a large-scale Web services QoS dataset, which promotes the progress of Web service research [23]. In addition, Zheng et al. proposed a recommendation framework via user-collaboration mechanism to collect personalized user QoS information, and further combined matrix factorization model and neighbor information to improve the quality of QoS prediction [4].

Many researchers utilize the context information to improve the quality of service recommendation in previous research works. Zheng et al. considered that QoS properties are deeply influenced by physical location and the network infrastructure of service users [5]. Hence, some location-based approaches have been studied for improving the prediction accuracy [9, 24, 10, 25]. In particular, Xu et al. proposed a location-aware PMF model by integrating the service users and their geographical neighbors' invocation experience information [5]. Chen et al. assumed that a group of service users, who locate closely to each other, have the similar invocation experience and further proposed a region-based approach through aggregating the similar IP-based regions as a larger region for making QoS prediction [10].

The Latent Factor Model is demonstrated its effectiveness on the Netflix data. Koren et al. proposed a bias-SVD model which combines the baseline estimating model and SVD model [12]. Different from the previous work, we propose the concept of shared latent feature vector to collect the QoS information of Web services' neighbors and further combine the global information via LFM approach to achieve the higher prediction accuracy. Extensive experiments employing a large-scale real-world QoS dataset are conducted to demonstrate the effectiveness of our approach.

IV. QoS prediction with Latent Factor Models

In this section, we firstly present an extended baseline estimating model for predicting the QoS value in Section IV-A. Afterwards, we show how the LFM model can be employed to solve the Web service QoS value prediction problem in Section IV-B. In Section IV-C, we build the Latent Neighbor Model through utilizing Web services' latent neighbors information. Further, we propose an integrated model by combining the LFM model and the Latent Neighbor Model together in Section IV-D. Finally, the complexity of our approach is analyzed in Section IV-E.

A. The Extended Baseline Estimating Model

To simplify the description of our approach, we choose the response time (i.e., round-trip time, RTT) as the example metric of QoS, which is defined as: the propagation of time from service user sending a request until receiving the corresponding response. As we know during previous sections, RTT is highly related to the network environment between service user and provider. For different service users or providers, the quality of individual network infrastructure influences the RTT greatly. As the same reason for the country where the service users or providers are located, RTT is also influenced by the overall quality of network environment in different countries. For different services, the processing time is decided by what function they will accomplish. However, the process time for the services of similar or equivalent functions, deployed by different providers, might be still different.

Based on the above assumption and a detailed analysis of real-world response time dataset, we could find some holistic and common tendencies: (1) some service users may cost less time than others by invoking the same services, (2) some services may need more time to achieve their tasks than others due to the complexity of them or the poor quality offered by a provider. To take these effects into consideration, we can utilize a simple model, which is called baseline estimating model [12]. For an unknown RTT value q_{ij}, we consider the effects of user and service through the baseline estimating model, which is indicated by b_{ij}:

$$b_{ij} = \mu + b_i + b_j \quad (1)$$

Here, the parameter μ indicates the overall average RTT, whereas b_i and b_j indicate the effect biases of service user u_i and service s_j respectively. Now, we extend the baseline estimate model by taking provider and its country into consideration. An extended baseline estimating model can be employed to predict the unknown RTT value q_{ij}, denoted by $b_{(i,j,l,k)}$:

$$b_{(i,j,l,k)} = \mu + b_i + b_j + b_l + b_k \quad (2)$$

where the parameters b_l and b_k indicate effect biases of provider and country respectively.

In general, we can acquire the effects biases parameters through solving the least squares problem:

$$E_1 = \min \sum_{(i,j,l,k)\in T} (q_{ij} - b_{(i,j,l,k)})^2 + \lambda_1 \cdot (b_i + b_j + b_l + b_k) \quad (3)$$

Here, the parameter λ_1 indicates regularization coefficient, a higher of which meaning the heavy punishment of model complexity.

According to [14], we can estimate the biases parameters b_i, b_j, b_l, b_k as follows:

$$b_i = \frac{\sum_{i\in \mathrm{R}(i)} (q_{ij} - \mu)}{\omega_1 + |\mathrm{R}(i)|} \quad (4)$$

$$b_j = \frac{\sum_{j\in \mathrm{R}(j)} (q_{ij} - \mu - b_i)}{\omega_2 + |\mathrm{R}(j)|} \quad (5)$$

$$b_l = \frac{\sum_{l\in \mathrm{R}(l)} (q_{ij} - \mu - b_i - b_j)}{\omega_3 + |\mathrm{R}(l)|} \quad (6)$$

$$b_k = \frac{\sum_{k\in \mathrm{R}(k)} (q_{ij} - \mu - b_i - b_j - b_l)}{\omega_4 + |\mathrm{R}(k)|} \quad (7)$$

Here, the parameters ω_1, ω_2, ω_3 and ω_4 indicate regularization coefficients, whereas $|\mathrm{R}(\cdot)|$ indicates corresponding statistic. For example, $\mathrm{R}(i) = \{(i,j,l,k) \in T \mid q_{ij} \neq NULL\}$.

B. The Latent Factor Model

For the problem of QoS prediction, we want to find an appropriate pattern to fit the available QoS values and further own the ability of predicting the unknown QoS values. In Section I, we have simply introduced the main idea of LFM based on the SVD approach. In the recommended system domain, typical SVD approaches consider that the interaction between users and items lies in the latent factor space and utilize a number of latent features to explain ratings which users give to items.

Similarly, we can apply SVD approach to model the RTT value prediction problem. We associate each user u_i with a user-factor vector $\mathbf{U}_i \in \mathrm{R}^d$, and each service s_j with a service-factor vector $\mathbf{S}_j \in \mathrm{R}^d$. Here, d indicates the number of latent factors to decide the RTT value of Web services invoked by the users. To distinguish the predicted value from the known q_{ij}, we use $\hat{q}_{ij}$ to indicate the predicted value. As the typical SVD approach, we can predict the unknown RTT value through taking an inner product as the following equation:

$$\hat{q}_{ij} = b_{ij} + \mathbf{U}_i^{\mathrm{T}} \mathbf{S}_j \quad (8)$$

Specially, if considering the broader scope of effects, as described in Section IV-A, the prediction equation could be simply modified as:

$$\hat{q}_{ij} = b_{(i,j,l,k)} + \mathbf{U}_i^{\mathrm{T}} \mathbf{S}_j \quad (9)$$

Similarly as the typical SVD model, we utilize the appropriate regularizing terms to avoid over fitting as follows:

$$E_2 = \min \sum_{(i,j,k,l)\in T} (q_{ij} - b_{(i,j,l,k)} - \mathbf{U}_i^{\mathrm{T}} \mathbf{S}_j)^2 + \lambda_1 \cdot (b_i^2 + b_j^2 + b_l^2 + b_k^2) + \lambda_2 \cdot (\| \mathbf{U}_i \|^2 + \| \mathbf{S}_j \|^2) \quad (10)$$

Here, the parameter λ_1 and λ_2 are the regularization coefficients, whereas $\| \cdot \|^2$ indicates the Frobenius norm. The first term is the original fit object function which should be minimized. In addition, we add two parts of regularizing terms to be the punishment to avoid over fitting. We can apply the stochastic gradient descent [26] to solve this optimization problem.

C. The Latent Neighbor Model

The neighborhood-based CF approaches, including user-based, item-based and the combination of both, have been widely employed to predict QoS value recent years [3, 5]. Since the user-service matrix is extremely sparse in real world, the traditional CF approaches are limited to achieve high precision of QoS values. To overcome the data sparsity problem, similarly as the solutions in recommended system domain [27], a number of research works have been verified effectively to combine the matrix factorization model with neighbor information to improve the accuracy of predicting unknown QoS value [4, 9]. Zheng et al. proposed a matrix factorization model through integrating neighbor model, which collects the global and local information together to achieve higher prediction accuracy [4]. In this section, we introduce a neighbor factor model, which utilizes the Web service's latent neighbor QoS information and collects the local information through sharing the same latent feature vector.

Firstly, let us analyze how the Web service's latent neighbor information can be employed to predict the unknown RTT value. For the individual user u_i, we usually estimate its unknown RTT value for a certain service s_j through consulting neighbors' invocation experience. However, due to the sparsity of data, the similar neighbors are always hard to be identified. Because the value of RTT is

highly related to the network infrastructure and Internet environment, we use the average RTT level to represent the user u_i's typical experience of invoking a set of Web services published by provider p_l . In other words, we consider the personalized connection quality of network environment of provider p_l for user u_i .

Based on the main idea of LFM, we map service user and provider to the same latent factor space. We associate each provider p_l with a provider-factor $\mathbf{P}_l \in \mathrm{R}^d$. We thus predict the average RTT level of provider p_l for service user u_i through taking an inner product as follow:

$$\overline{\hat{q}}_{il} = \mathbf{U}_i^{\mathrm{T}} \mathbf{P}_l \tag{11}$$

Further, we consider the personalized connection quality of overall network status of country c_k . Similarly, we associate each country c_k with a country-factor $\mathbf{C}_k \in \mathrm{R}^d$, and predict the average RTT level of country c_k for service user u_i as the following equation:

$$\overline{\hat{q}}_{ik} = \mathbf{U}_i^{\mathrm{T}} \mathbf{C}_k \tag{12}$$

Now, considering the RTT level of both provider p_l and country c_k , we obtain the following equation:

$$\overline{\hat{q}}_{(i,l,k)} = \mathbf{U}_i^{\mathrm{T}} (\mathbf{P}_l + \mathbf{C}_k) \tag{13}$$

Equation (9) can only predict the average RTT value of Web services published by the same provider because it ignores the latent factor of specific service. In other words, it cannot accurately predict the RTT value of a certain service. However, on the other hand, the latent neighbors' invocation experience can be learnt by the latent provider and country.

D. An Integrated Model

In previous sections, we give a detailed explication of three models: the extended baseline estimating model, the Latent Factor Model and the Latent Neighbor Model. Now, we present the final model named as LN-LFM, which integrates all three models:

$$\hat{q}_{ij} = b_{(i,j,l,k)} + \mathbf{U}_i^T [\alpha \cdot \mathbf{S}_j + (1-\alpha) \cdot (\mathbf{P}_l + \mathbf{C}_k)] \tag{14}$$

where the parameter α is a manipulative coefficient, and $\alpha \in [0,1]$. As the same as previous models, we utilize the appropriate regularizing terms to avoid over fitting for this model, such as:

$$\begin{aligned} E_3 = \min \sum\nolimits_{(i,j,l,k) \in T} & [q_{ij} - b_{(i,j,l,k)} - \mathbf{U}_i^{\mathrm{T}} (\alpha \cdot \mathbf{S}_j \\ & + (1-\alpha) \cdot (\mathbf{P}_l + \mathbf{C}_{\mathbf{k}})]^2 + \lambda_1 \cdot (b_i^2 + b_j^2 + b_l^2 + b_k^2) \\ + \lambda_2 \cdot (\| \mathbf{U}_i \|^2 & + \| \mathbf{S}_j \|^2 + \| \mathbf{P}_l \|^2 + \| \mathbf{C}_k \|^2) \end{aligned} \tag{15}$$

An easy stochastic gradient descent can be employed to solve this optimization problem. In the learning process of stochastic gradient descent, for each training sample, we compute the difference between the actual and predicted value, i.e., $e_{ij} = q_{ij} - \hat{q}_{ij}$. Each time a single iteration ends, we update the model parameters in following way:

$$b_k \leftarrow b_k + \gamma_1 \cdot (e_{ij} - \lambda_1 \cdot b_k) \tag{16}$$

$$b_l \leftarrow b_l + \gamma_1 \cdot (e_{ij} - \lambda_1 \cdot b_l) \tag{17}$$

$$b_j \leftarrow b_j + \gamma_1 \cdot (e_{ij} - \lambda_1 \cdot b_j) \tag{18}$$

$$b_i \leftarrow b_i + \gamma_1 \cdot (e_{ij} - \lambda_1 \cdot b_i) \tag{19}$$

$$\mathbf{C}_k \leftarrow \mathbf{C}_k + \gamma_2 \cdot [e_{ij} \cdot (1-\alpha) \cdot \mathbf{U}_i - \lambda_2 \cdot \mathbf{C}_k] \tag{20}$$

$$\mathbf{P}_l \leftarrow \mathbf{P}_l + \gamma_2 \cdot [e_{ij} \cdot (1-\alpha) \cdot \mathbf{U}_i - \lambda_2 \cdot \mathbf{P}_l] \tag{21}$$

$$\mathbf{S}_j \leftarrow \mathbf{S}_j + \gamma_2 \cdot (e_{ij} \cdot \alpha \cdot \mathbf{U}_i - \lambda_2 \cdot \mathbf{S}_j) \tag{22}$$

$$\mathbf{U}_i \leftarrow \mathbf{U}_i + \gamma_2 \cdot \{e_{ij} \cdot [\alpha \cdot \mathbf{S}_j + (1-\alpha) \cdot (\mathbf{P}_l + \mathbf{C}_k)] - \lambda_2 \cdot \mathbf{U}_i\} \tag{23}$$

Here, the parameters γ_1 , γ_2 both indicate the learning rate. After a certain number of iterative loops, we minimize the object function to solve the squares problem and then acquire the final prediction model. For simplicity in the following experiments, the parameters λ_1 , λ_2 and γ_1 , γ_2 are set the same values respectively, i.e. $\lambda_1 = \lambda_2$ and $\gamma_1 = \gamma_2$.

E. Complexity Analysis

The core of stochastic gradient descent approach is evaluating the object function E_3 and updating the model parameters in accordance with the gradient parameters. The computation cost of single iteration to evaluate the object function E_3 is $O(|T| d)$. Here, $|T|$ denotes the sample size of training data, which is not very large due to the absence of QoS data in real world. On the other hand, d denotes the dimensionality of latent feature space, which is a small positive integer. Because of the updating mode of stochastic gradient descent, the computation cost of gradients is the same as $O(|T| d)$. Consequently, the total computational complexity is $O(|T| d)$ for one iteration. Meanwhile, the procedure of training needs an appropriate number of iterations, which is also a small positive integer in general. Therefore, the computational time of our approach is linear with respect to the sample size of QoS data. In other words, the complexity analysis shows that our approach is very efficient and it can scale to the very large size of dataset.

V. Experiments and Evaluation

In this section, we study the performance of our proposed integrated model based on the extensive experiments, which can be divided into the following two parts: (1) comparing our approach with the other well-known approaches; and (2) studying the influence of model parameters on the prediction accuracy.

A. Dataset Description

Although large amount of data has been produced every day on the Internet, the real-world QoS dataset is limited for

researching in the service computing domain. To evaluate the performance of our approach, we choose the real-world Web service QoS dataset offered by Zheng et al [23]. The detailed statistics of the Web services QoS dataset is summarized in Table I. Our experiments concentrate on the response time QoS characteristic.

TABLE I. STATISTIC OF THE WEB SERVICES QOS DATASET

Statistics	Values
Num. of User	339
Num. of Web Service	5825
Num. of Provider	2841
Num. of Provider Country	73
Num. of Web Service Invocations	1,974,675
Range of Response-time (s)	0-20

B. Metrics

We use the Root Mean Squared Error (RMSE) metrics to measure the prediction quality of our proposed models in comparison with other state-of-the-art approaches. RMSE is a widely used evaluation metrics in recommendation system, defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j}(q_{ij} - \hat{q}_{ij})^2} \tag{24}$$

where q_{ij} denotes the real QoS value of service s_j experienced by user u_i, $\hat{q}_{ij}$ denotes predicted QoS value by user u_i of service, and N denotes the number of values in the testing dataset.

C. Comparison and Performance

To evaluate the prediction performance of our approach, we compare it with other well-known approaches: UMean (User Mean), IMean (Item Mean), UPCC [3], IPCC, UIPCC [5], Funk-SVD [12] and Bias-SVD [12]. UMean utilizes the average RTT value of each user to predict the unknown value, whereas IMean utilizes the average RTT value of each service to predict the unknown value. Besides, UPCC is a user-based CF approach, whereas IPCC is the classical item-based CF approach. UIPCC is the combination of UPCC and IPCC. Finally, Funk-SVD is the basic SVD approach, whereas Bias-SVD is a hybrid model by combining the Baseline Estimates Model and Funk-SVD Model.

In the real world, the user-service matrix is usually very sparse since a user usually invokes only a small number of Web services. In our experiments, we use different proportion matrix density of data (MD=5%, 7.5%, 10%, 12.5% or 15%) to compare the performance of our approach with others. For instance, the matrix density 5% represents we select 5% of values from the user-service matrix as the historical data to predict the remaining 95% part as the unknown values. We set $\alpha = 0.6$, $d = 20$, λ_1 = 0.01 and λ_2 =0.01 respectively.

The results are presented in Table II, showing that the RMSE of our approach is consistently smaller than other approaches, especially in the situation of data sparsity, e.g. MD=5% or MD=7.5%. It indicates that we have improved the quality of prediction by considering the personalized connection quality of latent neighbors for each service user. On the other hand, the performance of Bias-SVD always outperforms other approaches except LN-LFM, which indicates the prediction accuracy can be improved by taking bias effects into account. Specially, under the condition of matrix density 5%, Funk-SVD performs no better than UIPCC but Bias-SVD still performs well. Meanwhile, the degree of improvement is decreased with the increasing matrix density. Finally, the SVD-based approaches perform better than other approach in general.

D. Impact of α

The parameter α controls the impact of Latent Factor Model and Latent Neighbor Model in our approach. In other words, it controls how much proportion we can consult from the performance of service and its personalized connection quality of latent neighbors. In the extreme case, if $\alpha = 0$, we only consider the personalized connection quality of latent neighbors and, therefore, cannot accurately predict the unknown value as what we discuss in Section IV-C. On the other hand, if $\alpha = 1$, we only consider the performance of service and our approach is equivalent to Bias-SVD. Therefore, we need to determine the appropriate range of parameter α for improving the prediction accuracy of our approach.

Fig. 2 shows the impact of the parameter α on RMSE for our approach, where we can obviously find that it achieves smaller RMSE value when α range from 0.5 to 0.7, and achieves the smallest one when $\alpha = 0.6$. In other cases, our approach obtains worse prediction accuracy with the value of α less than 0.4 or greater than 0.7. Therefore, we can conclude that the effect of LFM accounts for a larger proportion.

E. Impact of d

The parameter d determines the number of latent factors, which ranges from 5 to 40 by a stable increment of 5. Fig. 3 shows the impact of the parameter d on RMSE in our approach, where we can find that the prediction accuracy of LN-LFM increases with the parameter d increasing from 5 to 30. However, it performs stably on RMSE with the parameter d continually increasing beyond 30.

VI. CONCLUSION AND FUTURE WORK

In this paper, we try to explain how to model the QoS prediction problem in the latent factor space. Based on the intuition that a set of Web services published by the same provider usually have the same condition of fundamental network environment, we extend the baseline estimating model and propose a neighbor model which evaluates the personalized connection quality of latent neighbors for each user and utilizes the same latent feature vector to collect the local information. Further, we propose an integrated model

to achieve higher prediction accuracy. Finally, we conduct extensive experiments on a large-scale real-world QoS dataset and the results demonstrate the effectiveness of our approach, especially in the situation of high data sparsity.

In the future, we will try to utilize the LFM model to explain some QoS properties other than RTT and further improve the personalized prediction accuracy. We also plan to extend our approach to the cloud computing environment, where the approach can be applied on huge amount of data to satisfy the requirement of the exponential growth.

TABLE II. ACCURACY COMPARISON IN RMSE (A SMALLER RMSE VALUE MEANS BETTER PERFORMANCE)

Approach	Response-time Matrix Density(MD)				
	MD=5%	MD=7.5%	MD=10%	MD=12.5%	MD=15%
UMean	1.8588	1.8570	1.8570	1.8572	1.8589
IMean	1.5726	1.5545	1.5440	1.5366	1.5362
UPCC	1.4042	1.3693	1.3417	1.3253	1.3187
IPCC	1.4351	1.3890	1.3515	1.3227	1.3060
UIPCC	1.3875	1.3592	1.3345	1.3151	1.3042
Funk-SVD	1.4262	1.3252	1.2785	1.2462	1.2155
Bias-SVD	1.3419	1.2958	1.2596	1.2280	1.2081
LN-LFM	1.2942	1.2602	1.2343	1.2140	1.2004

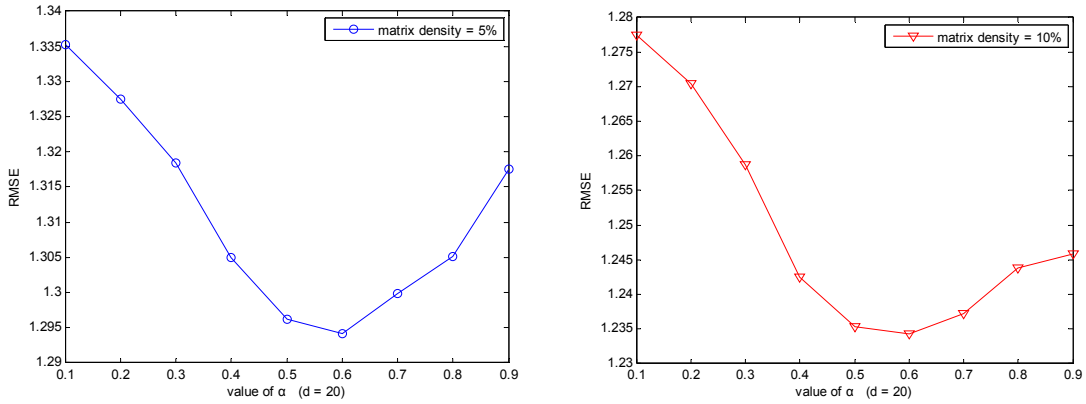


Figure 2. Impact of α

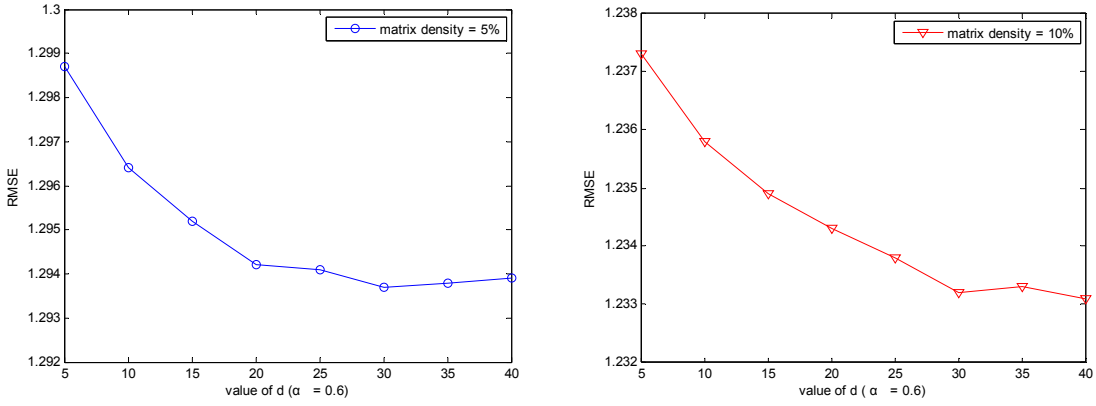


Figure 3. Impact of d

ACKNOWLEDGMENT

This paper is granted by National Natural Science Foundation of China under Grant(No. 61100043), Zhejiang Provincial Natural Science Foundation(No.LY12F02003), The National Key Technology R&D Program under Grant(No. 2012BAH24B04), and China Postdoctoral Science Foundation under Grant(No.2013M540492). The authors would also like to thank anonymous reviewers who made valuable suggestions to improve the quality of the paper.

REFERENCE

[1] D. Benslimane, S. Dustdar, and A. Sheth, "Services mashups: The new generation of web applications," IEEE, Internet Computing, vol. 12, pp. 13-15, 2008.

[2] L. Yao, Q. Z. Sheng, A. Segev, and J. Yu, "Recommending Web Services via Combining Collaborative Filtering with Content-Based Features," in *Proc. 20th Int'l Conf. Web Services (ICWS'13)*, 2013, pp. 42-49.

[3] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *Proc. 5th Int'l Conf. Web Services (ICWS'07)*, 2007, pp. 439-446.

[4] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Service Computing*, 2011.

[5] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Proc. 7th Int'l Conf. Web Services (ICWS'09)*, 2009, pp. 437-444.

[6] Q. Zhang, C. Ding, and C.-H. Chi, "Collaborative filtering based service ranking using invocation histories," in *Proc. 9th Int'l Conf. Web Services (ICWS'11)*, 2011, pp. 195-202.

[7] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," IEEE, Internet Computing, vol. 7, pp. 76-80, 2003.

[8] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, pp. 43-52.

[9] Y. Xu, J. Yin, W. Lo, and Z. Wu, "Personalized Location-Aware QoS Prediction for Web Services Using Probabilistic Matrix Factorization," in *Web Information Systems Engineering–WISE 2013, ed: Springer,* 2013, pp. 229-242.

[10] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware web service recommendation and visualization," *IEEE Transactions on Service Computing*, 2011.

[11] S. Gong, "A collaborative filtering recommendation algorithm based on user clustering and item clustering," Journal of Software, vol. 5, pp. 745-752, 2010.

[12] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426-434.

[13] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems,* 2007, pp. 1257-1264.

[14] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," Advances in neural information processing systems, vol. 13, pp. 556-562, 2001.

[15] R. Jin and L. Si, "A Bayesian approach toward active learning for collaborative filtering," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 2004, pp. 278-285.

[16] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for Web services selection with end-to-end QoS constraints," ACM Transactions on the Web (TWEB), vol. 1, p. 6, 2007.

[17] M. Mehdi, N. Bouguila, and J. Bentahar, "A QoS-Based Trust Approach for Service Selection and Composition via Bayesian Networks," in *Proc. 20th Int'l Conf. Web Services (ICWS'13)*, 2013, pp. 211-218.

[18] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," IEEE Transactions on, Software Engineering, vol. 33, pp. 369-384, 2007.

[19] Y. Feng, L. D. Ngan, and R. Kanagasabai, "Dynamic Service Composition with Service-Dependent QoS Attributes," in *Proc. 20th Int'l Conf. Web Services (ICWS'13)*, 2013, pp. 10-17.

[20] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Proc. 9th Int'l Conf. Web Services (ICWS'11)*, 2011, pp. 211-218.

[21] U. Küster, H. Lausen, and B. König-Ries, "Evaluation of semantic service discovery—a survey and directions for future research," in Emerging Web Services Technology, Volume II, ed: Springer, 2008, pp. 41-58.

[22] E. Stroulia and Y. Wang, "Structural and semantic matching for assessing web-service similarity," International Journal of Cooperative Information Systems, vol. 14, pp. 407-437, 2005.

[23] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *Proc. 8th Int'l Conf. Web Services (ICWS'10)*, 2010, pp. 83-90.

[24] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proc. 8th Int'l Conf. Web Services (ICWS'10)*, 2010, pp. 9-16.

[25] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "WSP: A Network Coordinate Based Web Service Positioning Framework for Response Time Prediction," in *Proc. 19th Int'l Conf. Web Services (ICWS'12)*, 2012, pp. 90-97.

[26] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 4, p. 1, 2010.

[27] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 203-210.