Collaborative Web Service QoS Prediction via Location-Aware Matrix Factorization and Unbalanced Distribution

Wei Xiong[1], Qiong Gu[1], Bing Li[2], Zhao Wu[1], Lei Yuan[1]

[1] School of Mathematics and Computers, HuBei University of Arts and Science, China
[2] International School of Software, Wuhan University, China
xwei9093@126.com, gujone@163.com, bingli@whu.edu.cn, wuzhao73@163.com, 1390405958@qq.com

Abstract

QoS prediction is critical to Web service selection and recommendation. This paper proposes a location-aware collaborative approach to QoS prediction of Web services by utilizing the past Web service usage history of service users, which avoids expensive and time-consuming Web service invocations. We first acquire and process client-side spatial location information. Then, an approach, which integrates spatial location constraint and LFM method and considers unbalanced distribution of data, is designed to achieve higher prediction accuracy for Web service QoS value based on the collected QoS data and location information. To validate our approach, large-scale experiments are conducted based on a real-world Web service dataset, WSDream. The results show that our proposed approach achieves higher prediction accuracy than other approaches.

Keywords: Web service, QoS prediction, User-collaboration, Matrix factorization, Location-aware

1 Introduction

Web services are self-described programmable applications conducted to achieve interoperability and accessibility over a network, which is implemented in standard interfaces and published through specific protocols [1]. Open services on the Web become increasingly abundant in the past several years. Meanwhile, the wide-spread use of Web services in cloud computing, especially in Software-as-a-Service (SaaS) [2], asks for the effective approaches for Web services.

The nonfunctional properties termed as Quality of Service (QoS) are identified as distinguishing characteristic of Web services [3]. They are mainly comprised of performance factors that include availability, response time, reliability, throughput, and etc [4-7]. A lot of QoS-based approaches have been proposed for Web service composition [8-10], web service selection [11-14], fault-tolerant Web services [15], and etc. Accurate QoS values of web services are desired to work well for these approaches. The QoS values of Web services can be measured both at server-side and at client-side. QoS values measured at server-side are published by service provider and uncover the shared feature of web services, which are consistent to all users (e.g., price, popularity, etc). However, QoS values measured at client-side are different. Users in different locations may experience different QoS performance of Web services due to unpredictable network environment. Therefore, it is necessary to obtain accurate and personalized client-side web service QoS values or their estimates (e.g., response-time, throughput, availability, etc.) [5, 16-17].

Conducting real-world Web service evaluation at the client-side, however, is a critical challenge. Web service invocations take costs. They may be charged in terms of the resources consumed in the Cyberspace or time elapse of invocations, since the server status such as workload, number of clients and the network environment such as congestions may change by time. Real-time performance testing may introduce extra transaction workload, which may impact the user experience on the system. Thus, the performance evaluation may not be accurate due to extra workloads, and it is difficult for various QoS-based approaches to perform excellently without accurate Web service QoS values in case of lacking sufficient client-side evaluations.

Recently, a few works have noted that users' locations serve to improve QoS prediction accuracy [18-21].

We propose a location-aware matrix factorization (LAMF) approach for collaborative and personalized QoS prediction. This approach utilizes the history log of conducting real-world Web service evaluation at the client-side to predict client-side QoS values.

Our approach first calculates the distances between users and reaches similar users. Then, the LAMF

*Corresponding Author: Qiong Gu; E-mail: gujone@163.com
DOI: 10.3966/160792642018081904010

approach employs both local location information of and global information to learn a factor model by training, and predict personalized Web service QoS values using this model. Based on collaboration theory [4], the QoS of Web service can be effectively predicted even current user did not conduct any evaluation on Web service.

Different from other QoS-based approaches for Web services, our approach focuses on providing more accurate and personalized QoS for service users by changing cost function of latent factor model (LFM) [22], which adds spatial location constraint and considers unbalanced distribution of data.

The contributions of this paper can be summarized as follows:

(1) We illustrate and verify the effectiveness of spatial location to QoS prediction for Web services.

(2) We propose an approach to integrate spatial location information for improving the prediction accuracy, considering unbalanced distribution of data.

(3) We conduct comprehensive experiments on the real-world dataset, demonstrating the effectiveness of our approach.

The remainder of this paper is organized as follows: Section 2 presents our collaborative Web service QoS prediction via location-aware matrix factorization. Section 3 describes our experiments in detail. Section 4 discusses the related works, and Section 5 concludes the paper.

2 Collaborative Qos Prediction via Location-Aware Matrix Factorization

In this section, we first present a scenario to illustrate the motivation of our work in Section 2.1. Then, we describe the issues of location-aware QoS Prediction in Section 2.2, next, we propose an approach to predicting QoS values with client-side different spatial locations in Section 2.3, 2.4, 2.5 and 2.6.

2.1 Motivative Scenario

In this section, we present a scenario to illustrate the motivation of our work. Assume that there exists a QoS-Based Web Service Recommendation system which consists of service providers and service users except system itself, where service providers and service users are distributed all over the world. Figure 1 shows such a scenario of location-aware Web service invocation. The top part of the diagram represents the client-side, and the bottom part represents the server-side. The link lines connect users to services from underlying network.

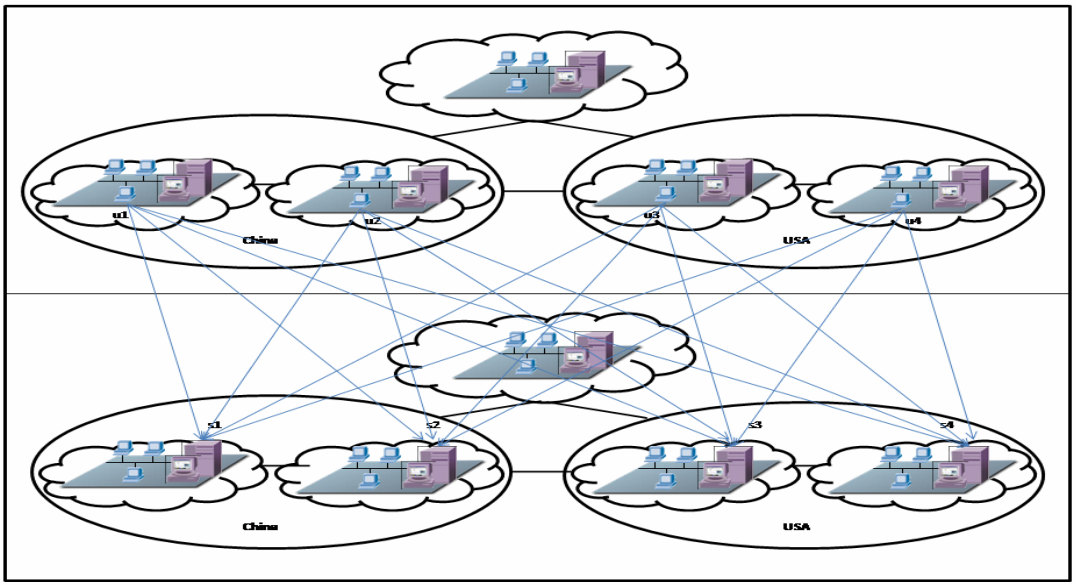

Figure 1. Motivative scenario

According to the report of Akamai in the second quarter of the year 2016 [23], South Korea (13.3Mbps) and Japan (12.0Mbps) are top two countries in the average connection speed, while China (2.8Mbps) has relatively lower connection speed. Therefore, while users in Seoul, Tokyo and Beijing invoke the same Web service for weather forecast, users in Seoul and Tokyo will experience shorter response time than in Beijing. Meanwhile, the response time may be even shorter for users in Seoul than in Tokyo.

According to the above, it is certainly valuable to integrate location information into QoS prediction, which makes it more accurate.

The distributions of response-time and throughput values based on WSDream dataset 2 [24] are shown in Figure 2. Figure 2 (a) and Figure 2 (d) show that the ranges of global distribution of response-time and throughput are 0-20 s and 0-1,000 kbps, respectively. Figure 2 (a) shows that most of the response-time values are 0-5 s and Figure 2 (d) shows that most of the

throughput values are 0-250 kbps. Figure 2 (b) and Figure 2 (e) shows that the ranges of bias distribution of response-time and throughput are 0-12 s and 0-560 kbps, respectively. Figure 2 (b) shows that most of the response-time values are 0-3 s, and Figure 2 (e) shows that most of the throughput values are 0-140 kbps. Figure 2 (c) and Figure 2 (f) shows that the ranges of user bias distribution of response-time and throughput are 0-5 s and 0-75 kbps, respectively. Figure 2 (c) shows that most of the response-time values are 0-1 s, and Figure 2 (f) shows that the throughput values are evenly split, most of them are in 30-45, 45-60, 60-75 kbps.

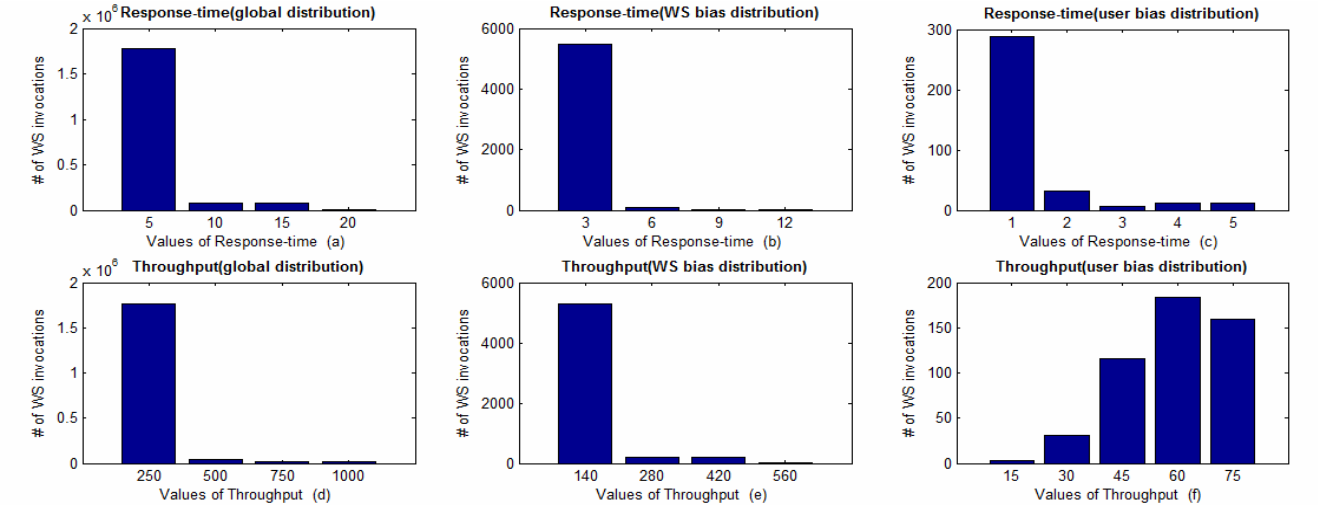

Figure 2. Values distributions

Therefore, it is certainly valuable to take the unbalanced distribution of data into consideration for QoS prediction.

Finally, there are some issues which need to be addressed as well: (1) How to represent client-side location? (2) How to acquire client-side spatial location? (3) How to integrate location information to collaborative filtering method? (4) How to consider unbalanced distribution of data? (5) How do we design experiments for performance evaluation?

2.2 Problem Description

The process of Web service QoS value prediction usually includes a user-item matrix, as shown in Figure 3, where each entry in this matrix represents the value of a certain QoS property (e.g., response-time in this example) of a Web service (e.g., to) observed by a service user (e.g., to). A real-world distance between users is utilized to specify the similarity on the spatial location, with larger distance for lower similarity.

	s1	s2	s3	s4	s5	s6
u1	q11					
u2		q22				
u3				q34		
u4					q45	
u5	q51					
u6			q63			q66

Figure 3. User-service invocation matrix

The intractable issue on QoS prediction is data sparsity. High data sparsity means that most entries in user-service invocation matrix are empty. Thus, our task is to fulfill missing values in this matrix. However, QoS values are principally affected by the context of Web services. Thus, current approaches should be modified to work more effectively.

Now we formally define the problem of QoS prediction for Web services as follows: Let U be the set of m users, S be the set of n Web services, A QoS element is a triple (i, j, q_{ij}) representing the observed quality of Web service s_j by user u_i ,where $i \in \{1,....m\}$, $j \in \{1,...,n\}$ and $q_{ij} \in R^p$ is a p-dimensional vector representing the QoS values of p criteria. User u_i is a tetrad $(i, ip, country, location)$. Let Ω be the set of all two-tuples $\{i, j\}$ and Λ be the subset of all known two-tuples $\{i, j\}$ in Ω. Consider a tensor $Y \in R^{m \times n}$ with each entry Y_{ij} representing the observed p-th criterion value of service s_j by user u_i . Then the missing entries $\{Y_{ij} / (i, j) \in \Omega - \Lambda\}$ should be predicted based on the existing entries $\{Y_{ij} / (i, j) \in \Lambda\}$ and User u_i 's tetrad.

2.3 Location Information Representation, Acquisition and Process

From above, we represent user information as a User u_i 's tetrad $(i, ip, country, location)$, where ip denotes IP address of user host, $country$ denotes the country that IP address belong to, and the location is a two-tuples $(longitude, latitude)$.

It's easy to acquire and construct the above location information, because the users' IP addresses are already known. To obtain full location information of a user, we further need to identify longitude and latitude which can be induced from the users' IP address. There are a lot of available online services and tools online for this purpose. For example, IP Locator, a query tool combined with Google maps, can map the IP addresses to longitude and latitude in real time.

An key issue on handling location information is how to measure user similarity or service similarity regarding their locations. In this aspect, our method is different from the work by Chen et al. [18]. In that method, location similarity is computed based on IP similarity. That is, if two users have similar IP addresses, they are deemed as physically close. This seems to be reasonable but may cause inaccuracies in reality. Due to several factors, such as IPv4 address shortage and multi-homing, IP prefixes (i.e., IP address blocks assigned to networks) are constantly divided into finer granularities [25]. Therefore, two IP addresses with similar values do not necessarily belong to the same network or geographically close.

Employing the Euclidean distance, the similarity between two user u_i and user u_j can be computed based on their observed longitude and latitude with the following equation:

$$Dist(i,j) = 1 \Big/ \sqrt{(u_i.x - u_j.x)^2 + (u_i.y - u_j.y)^2}, \quad \textbf{(1)}$$

where x represents longitude and y represents latitude.

S_{ij} is the normalized similarity score between user u_i and user u_j, which can be calculated by

$$S_{ij} = \frac{Dist(i,j)}{\sum_{j \in T(i)} Dist(i,j)}, \quad \textbf{(2)}$$

where $T(i) = \{k \mid Dist(i,k) > 0, i \neq k\}$.

Finally, we reach the normalized similarity score matrix S based on spatial location.

2.4 Basic Location-aware Matrix Factorization

User-based collaborative filtering methods [2] (named as UserCF) apply similar users to predict the missing QoS values by the following equation:

$$\hat{r}_{ui} = \sum_{v \in S(u) \cap N(i)} p_{uv} \cdot q_{vi}, \quad \textbf{(3)}$$

where $\hat{r}_{u,i}$ is a vector of QoS values of the missing value $r_{u,i}$ in the user-item matrix, $S(u)$ includes a set of the similar users to the current user u, $N(i)$ is a set of users to act the object i, p_{uv} is the similarity value between user u_i and user u_v, q_{vi} represents the interest of u_v to object i.

Since QoS values of Web service invoked by users nearby is similar, we hold the view that it is enough to predict QoS values using the user's spatial location. Thus, we substitute p_{uv} in equation (3) with S_{ij} from equation (2), and infer the following equation:

$$\hat{r}_{ui} = \sum_{v \in S(u) \cap N(i)} S_{uv} \cdot q_{vi}, \quad \textbf{(4)}$$

where q_{vi} is the element of all r_{ui} and S_{uv} is the location-aware normalized similarity between users.

2.5 Location-aware Matrix Factorization

Basic LAMF model may work using the spatial distance between users. However, this model potentially omits similar relation of users or items. In order to predict the missing value as accurate as possible, we propose an approach LAMF to predict missing values by combining the information of Latent factor and the spatial distance between users.

Latent factor model (named as LFM) proposed by Simon Funk is a matrix factorization method [22]. We can decompose Score matrix to multiplication of two low dimensional matrices with the following equation:

$$\hat{R} = P^T Q, \quad \textbf{(5)}$$

where $P \in R^{f \times m}$ and $Q \in R^{f \times n}$ are matrices by dimensionality reduction. Thus, the missing QoS values can be predicted by the following equation:

$$\hat{r}_{ui} = \sum_f p_{uf} \cdot q_{fi}, \quad \textbf{(6)}$$

where $p_{uf} = P(u,f)$ and $q_{if} = Q(i,f)$. We can learn the matrix P and Q by minimizing cost function with observed value of training set.

Simon Funk defines the cost function as following:

$$C(p,q) = \sum_{(u,i) \in Train} (r_{ui} - \hat{r}_{ui})^2 = \sum_{(u,i) \in Train} (r_{ui} - \sum_{f=1}^{F} p_{uf} \cdot q_{fi})^2, \quad \textbf{(7)}$$

Direct optimizing $C(p,q)$ from equation (7) may cause overfitting, therefore, term $\lambda(\| p_u \|^2 + \| q_i \|^2)$ is added to prevent overfitting where λ is regularization parameter. Consequently, the following equation:

$$C(p,q) = \sum_{(u,i) \in Train} (r_{ui} - \sum_{f=1}^{F} p_{uf} \cdot q_{fi})^2 + \lambda(\| p_u \|^2 + \| q_i \|^2), \quad \textbf{(8)}$$

is obtained. It also has a probabilistic interpretation with Gaussian observation noise, which is detailed in [18].

The above LFM utilizes the global information of all the available QoS values in the user-item matrix for predicting missing values. This approach is generally effective at estimating overall structure (global information) that relates simultaneously to all users and items. However, it does not consider the spatial location. We add spatial location constraint to LFM for

preserving both global information and spatial information mentioned above, and consider unbalanced distribution of data. Hence, we can minimize the following sum-of-squared-errors objective functions with quadratic regularization terms:

$$C(p,q)=\sum_{(u,i)\in Train}(r_{ui}-\hat{r}_{ui})^2+\lambda(\|p_u\|^2+\|q_i\|^2+\|b_u\|^2+\|b_i\|^2). \quad (9)$$

The missing QoS values can be predicted by the following equation:

$$\hat{r}_{ui}=\mu+b_u+b_i+\alpha\cdot\sum_v S_{uv}\cdot q_{vi}+(1-\alpha)\sum_f p_{uf}\cdot q_{fi}, \quad (10)$$

where μ is global average of all r_{ui}, b_u is user bias term, b_i is item bias term, $p_{uf}=P(u,f)$, $q_{if}=Q(i,f)$, α is a balance parameter, q_{vi} is QoS value of user v on item i as the element of all r_{ui} and S_{uv} is the location-aware normalized similarity score between users where v is neighbors of user u.

Considering that the overall distribution of the data collected are different in case of different environment, and the distribution of different dimensions such as uses and servers are different, μ, b_u, b_i are added to equation (10). μ addresses global unbalanced distribution of QoS value. b_u indicates the average related to user u. b_i indicates the average related to service i.

A local minimum of the objective function given by equation (9) can be found by performing gradient descent in b_u, b_i, p_{uf}, q_{if}:

$$\frac{\partial C}{\partial b_u}=-2e_{ui}+2\lambda b_u, \quad (11)$$

$$\frac{\partial C}{\partial b_i}=-2e_{ui}+2\lambda b_i, \quad (12)$$

$$\frac{\partial C}{\partial p_{uf}}=-2(1-\alpha)e_{ui}\cdot q_{if}+2\lambda p_{uf}, \quad (13)$$

$$\frac{\partial C}{\partial q_{if}}=-2(1-\alpha)e_{ui}\cdot p_{uf}+2\lambda q_{if}, \quad (14)$$

where e_{ui} is defined as following:

$$e_{ui}=r_{ui}-\hat{r}_{ui}, \quad (15)$$

2.6 Complexity Analysis

The main computational cost of LAMF model arises from the procedure of gradient descent on Equation (9), whose iteration number is an absolutely small constant. So we only need to analyze the computational complexity of Eq. (11), (12), (13), (14).

The computational complexity of $\frac{\partial C}{\partial b_u}$, $\frac{\partial C}{\partial b_i}$, $\frac{\partial C}{\partial p_{uf}}$, $\frac{\partial C}{\partial q_{if}}$ in a single iteration is $O(\rho_Q(k+f))$ each, where ρ_Q denotes the non-empty values in the user-service invocation matrix, f is the dimensionality of latent feature vectors which also is a small constant, k denotes the number of the user's neighbors. Finally, equation (11), (12), (13), (14) are approximately combined into $O(\rho_Q(k+f))$ which indicates that the computational complexity is linearly scalable to the size of datasets, so the LAMF model can be employed on very large datasets.

3 Experiments

In this section, we conduct experiments to compare the prediction accuracy of our LAMF approach with other state-of-the-art collaborative filtering methods. Our experiments are intended to address the following questions: 1) How does user location affect the QoS values of services? Does closeness in location indicates similarity in QoS values? 2) How does our approach compare with published state-of-the-art collaborative filtering algorithms? 3) How does the model parameter α affect the prediction accuracy? 4) What is the impact of the matrix density, training user number and dimensionality on the prediction accuracy?

3.1 Data Description

We adopt a real-world Web service dataset: WSDream dataset 2, which was published in references [25]. The dataset contains QoS records of service invocations on 5825 Web services from 339 service users, which are transformed into a user-service matrix. Each item of the user-service matrix is a pair of values: response-time and throughput. Therefore the original user-service matrix can be decomposed into two simpler matrices: response-time matrix and throughput matrix. We use either response-time matrix or throughput matrix to compute user similarity and service similarity in the experiments.

Although we only study the response-time and throughput in the experiments, the proposed LAMF approach can be applied to other QoS properties easily. When predicting value of a certain QoS property, the value of the entry in the user-item matrix is the corresponding QoS value (e.g., response-time, throughput, failure probability) observed by a user on a certain Web service. Our LAMF approach can be employed on different QoS properties directly without any modifications.

3.2 Metrics

We use mean absolute error (MAE) and root-mean-

squared error (RMSE) metrics to measure the prediction quality of our method in comparison with other collaborative filtering methods. MAE is defined as

$$MAE = \frac{\sum_{i,j} \left| R_{ij} - \hat{R}_{ij} \right|}{N}, \quad \textbf{(16)}$$

where R_{ij} denotes the observed QoS value of Web service j observed by user i, $\hat{R}_{ij}$ is the predicted QoS value, and N is the number of predicted values. The MAE is the average over the verification sample of the absolute values of the differences between a prediction result and the corresponding observation. The MAE is a linear score, which means that all the individual differences are weighted equally in the average.

RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i,j} \left| R_{ij} - \hat{R}_{ij} \right|^2}{N}}, \quad \textbf{(17)}$$

where the difference between a prediction result and the corresponding observed values are each squared and then averaged over the sample. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable.

3.3 Comparison

In this section, to show the prediction accuracy of our Basic LAMF and LAMF approach, we compare our method with the following approaches:

UMEAN (user mean). This method employs a service user's average QoS value on the used Web services to predict the QoS values of the unused web services.

IMEAN (item mean). This method employs the average QoS value of the web service observed by other service users to predict the QoS value for a service user who never invokes this Web service previously.

UPCC (user-based collaborative filtering method using PCC). This approach is much similar with user-based CF, which first calculates the similarity between users based on PCC and then gains the predicted value as the weighted average of the known values of the similar users [26-27].

IPCC (item-based collaborative filtering method using PCC). This approach is similar with UPCC, except that the key procedure is the similarity calculation between items [28].

WSRec (a collaborative filtering based web service recommender system). This approach integrates UPCC and IPCC [29].

LACF (Location-Aware Collaborative Filtering for QoS-Based Service Recommendation). This is an approach of location-aware QoS prediction for Web services using hierarchy region [19].

WL-PMF (weighted Location-Aware PMF). This is an approach of personalized location-aware QoS prediction for Web services using probabilistic matrix factorization [21].

In the real world, user-item matrices are usually very sparse since an user usually invokes a small number of Web services. In this paper, to conduct our experiments realistically, we randomly remove entries from the user-item matrix to make the matrix sparser with different density (i.e., 10, 20, and 30 percent). Matrix density 10 percent, for example, means that we randomly select 10 percent of the QoS entries to predict the remaining 90 percent of QoS entries. The original QoS values of the removed entries are used as the expected values to study the prediction accuracy. The above seven methods together with our LAMF method are employed for predicting the QoS values of the removed entries. The parameter settings of all methods are $\alpha = 0.1$, $training\ user\ number = 100,140$, $top-k = 40$, $matrix\ density = 10\%, 20\%, 30\%$, and $Dimensionality = 100$ in the experiments. The experimental results are shown in Table 1, and detailed investigations of parameter settings will be provided in Sections 3.4 to 3.6.

The experimental results of Table 1 show that:

(1) Basic LAMF approach outperforms the approaches (UMEAN and IMEAN) and is slightly below the approaches (UPCC and IPCC). Moreover, it is very close to IPCC. This observation indicates that location-aware information which is orthogonal with user-item matrices information works solely in collaborative filtering methods, which makes it possible that combining the information of user-item matrices and the spatial distance may provide more accurate.

(2) Under all experimental settings, our LAMF method obtains smaller MAE and RMSE values consistently, which indicates better prediction accuracy.

(3) The MAE and RMSE values of LAMF become smaller with the increase of the given number from 10 to 30, indicating that the prediction accuracy can be improved by providing more QoS values.

With the increase of the training user number from 100 to 140, and with the increase of the training matrix density from 10 to 30 percent, the prediction accuracy also achieve significant enhancement, since larger and denser training matrix provides more information for the prediction.

Table 1. Performance Comparison

QoS properities	Methods	Training Users=100						Training Users=140					
		Matrix Density=10%		Matrix Density=20%		Matrix Density=30%		Matrix Density=10%		Matrix Density=20%		Matrix Density=30%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Response-time	UMEAN	0.9502	1.8258	0.9468	1.8218	0.9463	1.8150	0.8922	1.7721	0.8902	1.7660	0.8900	1.7605
	IMEAN	0.9086	1.9643	0.9227	2.0132	0.9449	2.0641	0.8275	1.8815	0.8454	1.9260	0.8656	1.9563
	UPCC	0.7753	1.7322	0.7368	1.7072	0.7437	1.7753	0.7032	1.6428	0.6693	1.6341	0.6736	1.6972
	IPCC	0.9757	2.0071	0.8395	1.9268	0.7872	1.9420	0.8676	1.9116	0.7495	1.8443	0.7078	1.8449
	WSRec	0.9107	1.8401	0.7537	1.7309	0.7437	1.7753	0.8032	1.7424	0.6666	1.6534	0.6179	1.6636
	WL-PMF	0.9013	1.9541	0.7838	1.8919	0.7652	1.8075	0.7933	1.8494	0.7671	1.8153	0.6928	1.7163
	LACF	0.9131	2.0156	0.7998	1.8977	0.7774	1.8735	0.8039	1.9849	0.7471	1.8466	0.7033	1.7165
	BASIC LAMF	**0.9020**	**2.0426**	**0.8684**	**1.9921**	**0.8402**	**1.9430**	**0.8663**	**1.9845**	**0.8391**	**1.9406**	**0.8168**	**1.9033**
	LAMF	**0.8315**	**1.9533**	**0.7738**	**1.8547**	**0.7306**	**1.7614**	**0.7894**	**1.8838**	**0.7325**	**1.7852**	**0.6880**	**1.6953**
Throughput	UMEAN	53.26	106.08	52.89	105.88	52.81	105.26	53.63	107.01	53.70	106.35	53.65	105.78
	IMEAN	38.47	106.85	39.64	109.92	41.58	115.18	40.29	111.30	41.06	112.30	42.85	117.19
	UPCC	36.40	101.52	35.14	104.87	36.76	112.32	36.14	100.99	33.87	103.50	34.84	110.27
	IPCC	46.20	111.73	42.54	110.49	42.35	115.13	46.96	114.97	43.26	112.34	43.19	117.21
	WSRec	43.08	113.77	38.20	102.79	37.73	106.91	43.63	106.15	38.29	103.33	36.63	105.45
	WL-PMF	42.46	110.44	37.64	103.34	36.66	101.88	42.73	109.83	36.88	104.78	35.91	104.19
	LACF	44.98	115.95	39.90	108.57	38.38	110.11	45.15	111.61	40.61	107.59	38.18	106.70
	BASIC LAMF	**43.82**	**113.77**	**41.88**	**110.08**	**40.16**	**106.33**	**45.08**	**115.84**	**43.29**	**112.12**	**41.71**	**108.89**
	LAMF	**40.44**	**108.78**	**37.52**	**103.16**	**34.84**	**97.73**	**41.13**	**109.93**	**38.07**	**104.07**	**35.28**	**98.66**

3.4 Impact of Parameter α

In our LAMF method, parameter α controls how much our method relies on spatial location information and users' latent factor. If $\alpha = 0$, we only employ the users' latent factor information for making prediction. If $\alpha = 1$, we predict the users' QoS values purely by their spatial location information. In other cases, we integrate spatial location information withusers' latent factor for missing QoS value prediction. Currrent parameter settings are *training user number* $= 100$, *Dimensionality* $= 100$ and *matrix density* $= 10\%, 20\%$.

Figure 4. shows the impacts of parameter α on the prediction results. We observe that optimal α value settings can achieve better prediction accuracy, which demonstrates that integrating spatial location information with users' latent factor methods will improve the prediction accuracy. No matter for response-time or throughput, as α increases, the MAE and RMSE values decrease (prediction accuracy increases) at first, but when α surpasses a certain threshold, the MAE and RMSE values increase slowly with further increase of the value of α. This phenomenon confirms the intuition that purely using the spatial location information method or purely employing the latent factor method cannot generate better QoS value prediction performance than integrating these two favors together.

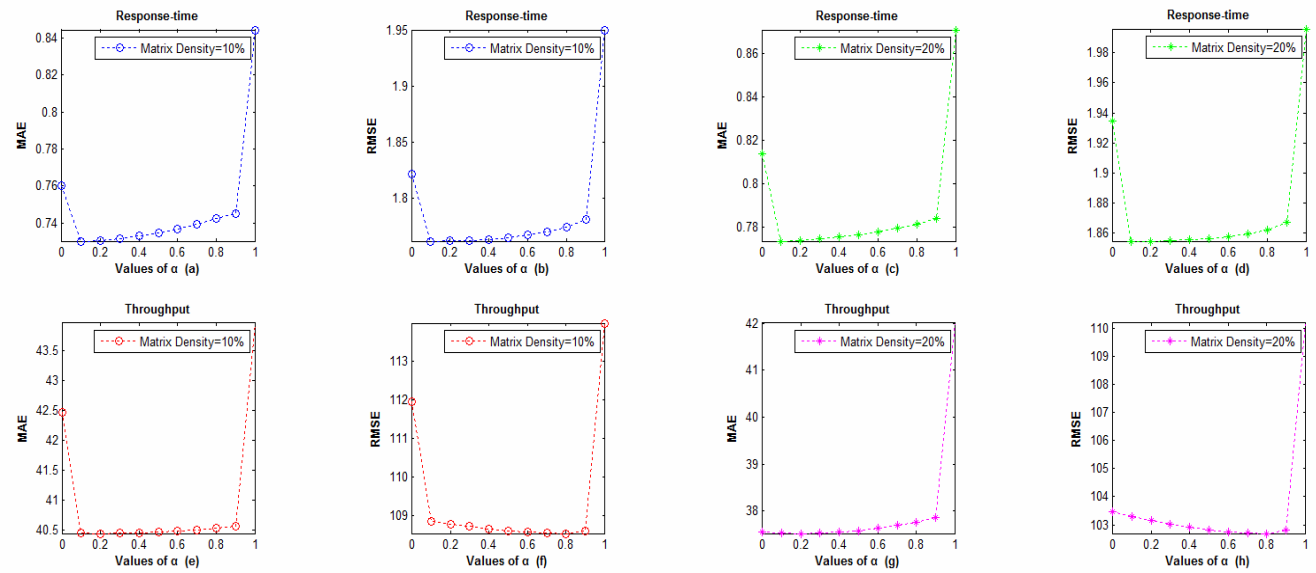

Figure 4. Impact of parameter α

From Figure 4(a) and Figure 4(b), when using user-item matrix with 10 percent density, we observe that our LAMF method achieves the best performance when is α around 0.1, while smaller values like $\alpha = 0$ or larger values like $\alpha = 0.4$ can potentially degrade the model performance. In Figure 4(c) and Figure 4(d), when using user-item matrix with 20 percent density, the optimal value of α is also around 0.1 or 0.2 for MAE and around 0.3 for RMSE. The optimal values of MAE and RMSE are different because MAE and RMSE are different metrics following different evaluation criteria. The optimal values of Figure 4(e), Figure 4(f), Figure 4(g), and Figure 4(h) are all between 0.1 and 0.8, which owes to different distribution of response-time and throughput. This observation indicates that optimally combining the two methods can achieve better prediction accuracy than purely or heavily relying one kind of method, and this is why we use as $\alpha = 0.1$ the default settings in other experiments. The same as Table 1, another observation from Figure 4 is that denser matrix provides better prediction accuracy.

3.5 Impact of Matrix Density

As shown in Table 1 and Figure 4, the accuracy of our LAMF method is influenced by the matrix density. To study the impact of the matrix density on the prediction results, we change the matrix density from 2 to 20 percent with a step value of 2 percent. We set *training user number* $= 100$, $\alpha = 0.1$ and *Dimensionality* $= 100$ in this experiment.

Figure 5 shows the experimental results, where Figure 5(a) and Figure 5(b) are the experimental results of response-time, and Figure 5(c) and Figure 5(d) are the experimental results of throughput. Figure 5 shows that when the matrix density is increased from 2 to 6 percent, the prediction accuracy of the LAMF method is significantly enhanced. With the further increase from 6 to 20 percent, the speed of prediction accuracy enhancement slows down. This observation indicates that when the matrix is very sparse, the prediction accuracy can be greatly enhanced by collecting more QoS values to make the matrix denser.

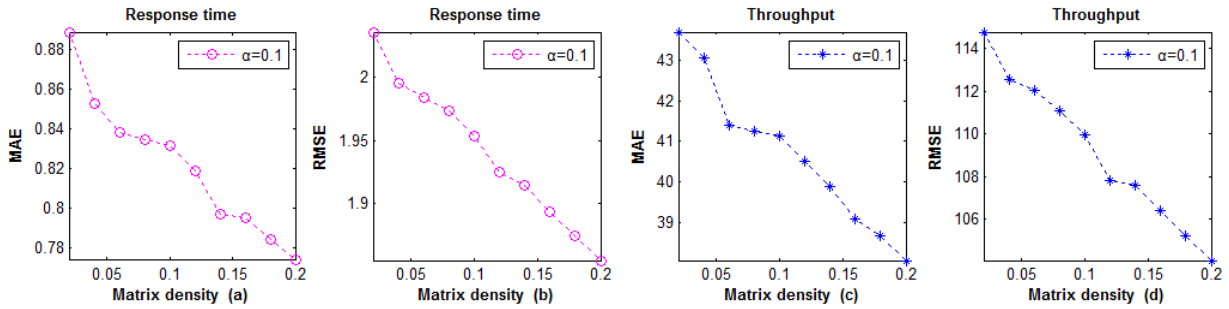

Figure 5. Impact of parameter matrix density

3.6 Impact of Dimensionality

Dimensionality determines how many factors are utilized to factorize the user-item matrix. To study the impact of the dimensionality, we vary the values of dimensionality from 10 to 100 with a step value of 10. We set *training user number* $= 100$, $\alpha = 0.1$, and *matrix density* $= 30\%$ in this experiment.

Figure 6(a) and Figure 6(b) show the experimental results of response-time, while Figure 6(c) and Figure 6(d) show the experimental results of throughput. As shown in Figure 6, the values of MAE and RMSE decrease when the dimensionality is increased from 10 to 100. These observed results coincide with the intuition that relative larger values of dimensions generate better recommendation results. However, the computational time of our LAMF approach is linear with respect to the value of dimensionality. Larger dimensionality value will require longer computation time. Moreover, the dimensionality can not be set to a very high value because it will cause the overfitting problem, which will potentially hurt the recommendation quality.

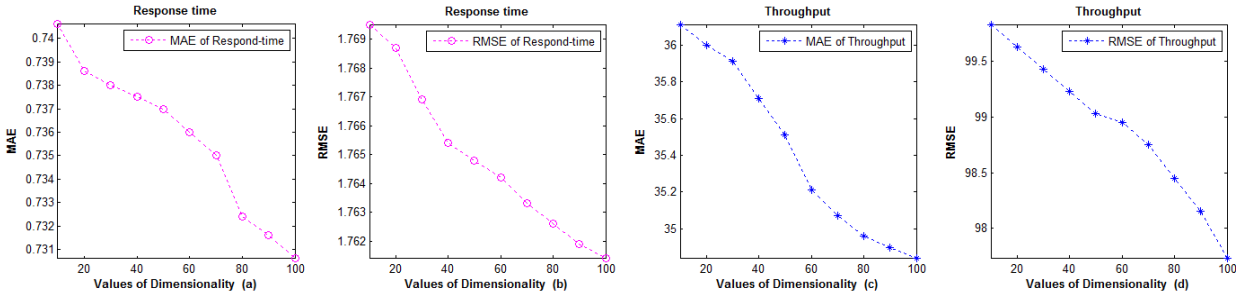

Figure 6. Impact of parameter dimensionality

4 Related Work and Discussion

Web services have been widely applied in a lot of domains (e.g., Web service composition, Web service selection, fault-tolerant Web services, and etc), which usually assume that

Web service QoS values are already known or can be easily obtained from the service providers or third-party registries. However, this is not always the case, and QoS prediction is a critical issue.

Collaborative filtering methods are widely adopted in commercial recommender systems [28, 30-32]. The first work of QoS prediction using collaborative filtering technique was conducted by Deora et al. [4]. They proposed a user-based CF algorithm to predict QoS values. Two types of collaborative filtering approaches are widely studied: neighborhood-based and model-based.

Most analyzed examples of neighborhood-based collaborative filtering include user-based approaches [26, 33], item-based approaches [34-35], and their fusion. User-based approaches predict the missing values of a user based on the values of similar users. Item-based approaches predict the missing values of a current user based on the computed information of items similar to those chosen by the current user. A hybrid user-based and item-based CF algorithm was proposed to predict QoS values by Zheng et al. [29], and carried out a series of large-scale experiments based on real Web services dataset. Neighborhood-based approaches often use the PCC algorithm [28] and the VSS algorithm [26] as the similarity computation methods. PCC-based collaborative filtering approaches generally can achieve higher prediction accuracy than the VSS-based algorithms, since PCC considers the differences of the user value characteristics.

The model-based approaches are used to learn a predefined model by training data sets. (e.g., the clustering model [36], aspect models [37], matrix factorization methods [38-41]). Matrix factorization methods focus on fitting the user-item matrix with low-rank approximations, which is engaged to make further predictions in case only a small number of factors influence the values in the user-item matrix. The neighborhood-based methods utilize the values of similar users or items (local information) for making value prediction, while model-based methods, like matrix factorization models, employ all the value information of the matrix (global information) for making value prediction.

The previous approaches employing collaborative filtering methods are limited by less to user similarity measurement. Zhang et al. [42] suggested that it was better to combine users' QoS experiences, environment factor and user input factor to predict Web services QoS values. But how to obtain environment factor and user input factor were not discussed. Chen et al. [18] were the first to recognize the influence of user location in Web services QoS prediction and proposed a novel method which group users into a hierarchy of regions according to users' locations and their QoS records. Users in a region are similar, and the method searches the regions, which may lose some information. Tang et al. [19] do the similar work named LACF, just trying different hierarchy way. The method also searches the regions which may lose some information, but avoids the assumption that the distribution of QoS values is Gaussian distribution.

Lo et al. [20] considered spatial location constraint, and appended a third regularization term at the end of the objective function of SVD-like Matrix Factorization [43]. Since the main purpose of such a kind of usage is to forbid overfitting in the learning process, it is hard to give a persuasive interpretation from the perspective of neighbors' contributions to QoS values. Xu et al. [21] developed Lo's work named WL-PMF, and SVD-like Matrix Factorization has been proved a special case of PMF model, in case that the distribution of QoS values is assumed to be the Gaussian distribution [38]. Thus, two models are the same in essence, whose difference is that WL-PMF model places spatial location constraint as weight into the prediction equation $\hat{r}_{ui}$. Furthermore, Wang et al. considered the difference between subjective and objective data [44], and performing the prediction via multi-dimensional QoS measures such as time and location [45].

Our models LAMF is also a PMF model which adds bias parameters considering unbalanced distribution of data and spatial location constraint such as WL-PMF model into the prediction equation $\hat{r}_{ui}$, however, because the position placed is different, our model LAMF 's complexity is $O(\rho_Q(k+f))$,yet WL-PMF model 's complexity is $O(\rho_Q\overline{g}kd))$.

5 Conclusion and Future Work

Based on the intuition to the effectiveness of spatial location in QoS prediction, we propose an LAMF approach for making more accurate QoS value prediction, which systematically integrates spatial location constraint and LFM approaches to achieve higher prediction accuracy, and considers unbalanced distribution of data simultaneously. The results of extensive experiments show the effectiveness of our approach.

Our approach need acquire and construct location information from mapping the IP address to longitude and latitude, which is physical spatial location. However, cyber spatial location we need should be hierarchical structure based on AS (Autonomous System) in fact. Thus, we plan to design better mechanisms to overcome the obstacle.

The LAMF approach in this paper can be applied to predict client-side QoS properties. We plan to conduct more studies to predict client-side QoS properties by integrating time factor and location factor. Client-side QoS values usually manifest trend, seasonality, periodicity and random on the time dimension due to resource-consuming of the host that Service located. Moreover, if Network traffic factor is considered, the issue will become more complicated.

Moreover, we plan to apply our approach to the cloud computing environments, where the user applications which invoke the Web services are usually deployed and running on the cloud. The change of application scenarios usually brings new issues.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2016YFB0800400), the National Program on Key Basic Research Project (973 Program) of China under Grant (No. 2014CB340400), the National Natural Science Foundation of China under grant (Nos. 61273216 and 61572371), the Natural science foundation of Hubei province in China (No. 2016CFB406) and the Humanities and Social Science Foundation of the Ministry of Education of China (No. 15YJAZH015)

References

[1] L. J. Zhang, J. Zhang, H. Cai, *Services Computing*, Springer, 2007.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A View of Cloud Computing, *Communications of the ACM*, Vol. 53, No. 4, pp. 50-58, April, 2010.

[3] M. P. Papazoglou, Service-Oriented Computing: Concepts, Characteristics and Directions, *Proceedings of the Fourth International Conferece on Web Information Systems Engineering (WISE '03)*, Rome, Italy, 2003, pp. 3-12.

[4] V. Deora, J. Shao, W. A. Gray, N. J. Fiddian, A Quality of Service Management Framework Based on User Expectations, *Proceedings of First International Conference on Service-Oriented Computing (ICSOC '03)*, Trento, Italy, 2003, pp. 104-114.

[5] E. M. Maximilien, M. P. Singh, Conceptual Model of Web Service Reputation, *ACM SIGMOD Record*, Vol. 31, No. 4, pp. 36-41, December, 2002.

[6] W. T. Tsai, X. Zhou, Y. Chen, X. Bai, On Testing and Evaluating Service-Oriented Software, *IEEE Computer*, Vol. 41, No. 8, pp. 40-46, August, 2008.

[7] G. Wu, J. Wei, X. Qiao, L. Li, A Bayesian Network Based QoS Assessment Model for Web Services, *Proceedings of International Conference on Services Computing (SCC '07)*, Salt Lake, UT, 2007, pp. 498-505.

[8] J. Hu, X. Chen, Y. Cao, L. Zhu, A Comprehensive Web Service Selection Algorithm on Just-in-Time Schedul -ing, *Journal of Internet Technology*, Vol. 17, No. 3, pp. 495-502, May, 2016.

[9] D. Ardagna, B. Pernici, Adaptive Service Composition in Flexible Processes, *IEEE Transactions on Software Engineering*, Vol. 33, No. 6, pp. 369-384, June, 2007.

[10] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q. Z. Sheng, Quality Driven Web Services Composition, *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, Budapest, Hungary, 2003, pp. 411-421.

[11] P. A. Bonatti, P. Festa, On Optimal Service Selection, *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, Chiba, Japan, 2005, pp. 530-538.

[12] V. Cardellini, E. Casalicchio, V. Grassi, F. L. Presti, Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes, *Proceedings of International Conference on Web Services (ICWS '07)*, Salt Lake, UT, 2007, pp. 743-750.

[13] L. Mei, W. K. Chan, T. H. Tse, An Adaptive Service Selection Approach to Service Composition, *Proceedings of International Conference on Web Services (ICWS '08)*, Beijing, China, 2008, pp. 70-77.

[14] T. Yu, Y. Zhang, K.-J. Lin, Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints, *ACM Transactions on the Web*, Vol. 1, No. 1, pp. 1-26, May, 2007.

[15] N. Salatge, J.-C. Fabre, Fault Tolerance Connectors for Unreliable Web Services, *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '07)*, Edinburgh, UK, 2007, pp. 51-60.

[16] L. B. Dong, T. He, L. Zhang, G. Z. Sun, J. Y. Shao, QoS Optimal Location-Based Services Architecture Based on Mobile Cloud Computing and Behavior Prediction, *Journal of Internet Technology*, Vol. 16, No. 5, pp. 895-904, September, 2015.

[17] C. F. Lai, Y. C. Chang, H. C. Chao, M. S. Hossain, A. Ghoneim, A Buffer-Aware QoS Streaming Approach for SDN-Enabled 5G Vehicular Networks, *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 68-73, August, 2017.

[18] X. Chen, X. Liu, Z. Huang, H. Sun, RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation, *Proceedings of International Conference on Web Services (ICWS '10)*, Miami, FL, 2010, pp. 9-16.

[19] M. Tang, Y. Jiang, J. Liu, X. Liu, Location-aware Collaborative Filtering for Qos-based Service Recommendation, *Proceedings of the 19th International Conference on Web Services (ICWS '12)*, Honolulu, HI, 2012, pp. 202-209.

[20] W. Lo, J. Yin, S. Deng, Y. Li, Z. Wu, Collaborative Web Service QoS Prediction with Location-Based Regularization, *Proceedings of the 19th International Conference on Web Services (ICWS '12)*, Honolulu, HI, 2012, pp. 464-471.

[21] Y. Xu, J. Yin, W. Lo, Z. Wu, Personalized Location-Aware QoS Prediction for Web Services Using Probabilistic Matrix

Factorization, *Proceedings of the Fourteenth International Conference on Web Information Systems Engineering (WISE '13)*, Nanjing, China, 2013, pp. 229-242.

[22] S. Funk, *Netflix Update: Try This at Home*, http://sifter.org/simon/journal/20061211.html.

[23] D. Belson, *The State of the Internet, Akamai*, Available: https://www.akamai.com/uk/en/about/our-thinking/state-of-the-internet-report/state-of-the-internet-connectivity-visualization.jsp.

[24] G. Huston, *BGP Routing Table Analysis Reports, BGP Home*, http://bgp.potaroo.net/.

[25] Z. Zheng, Y. Zhang, M. R. Lyu, Distributed QoS Evaluation for Real-world Web Services, *Proceedings of International Conference on Web Services (ICWS '10)*, Miami, Florida, 2010, pp. 83-90.

[26] J. S. Breese, D. Heckerman, C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, Madison, WI, 1998, pp. 43-52.

[27] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, H. Mei, Personalized QoS Prediction for Web Services via Collaborative Filtering, *Proceedings of International Conference on Web Services (ICWS '07)*, Salt Lake, UT, 2007, pp. 439-446.

[28] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, 1994, pp. 175-186.

[29] Z. Zheng, H. Ma, M. R. Lyu, I. King, QoS-Aware Web Service Recommendation by Collaborative Filtering, *IEEE Transactions on Services Computing*, Vol. 4, No. 2, pp. 140-152, April- June, 2011.

[30] R. Burke, Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction*, Vol. 12, No. 4, pp. 331-370, November, 2002.

[31] X. Su, T. M. Khoshgoftaar, X. Zhu, R. Greiner, Imputation-Boosted Collaborative Filtering Using Machine Learning Classifiers, *Proceedings of the ACM Symposium on Applied Computing (SAC '08)*, Fortaleza, Brazil, 2008, pp. 949-950.

[32] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu, F. Yang, Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems, *IEEE Transactions on Services Computing*, Vol. 8, No. 5, pp. 755-767, September- October, 2015.

[33] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, An Algorithmic Framework for Performing Collaborative Filtering, *Proceedings of the 22nd Annual International ACM SIGIR Conference Research and Development in Information Retrieval (SIGIR '99)*, Berkeley, CA, 1999, pp. 230-237.

[34] G. Linden, B. Smith, J. York, Amazon.com Recommendations: Item-to-Item Collaborative Filtering, *IEEE Internet Computing*, Vol. 7, No. 1, pp. 76-80, January/February, 2003.

[35] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, Hong Kong, China, 2001, pp. 285-295.

[36] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, Z. Chen, Scalable Collaborative Filtering Using Cluster-Based Smoothing, *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, Salvador, Brazil, 2005, pp. 114-121.

[37] T. Hofmann, Latent Semantic Models for Collaborative Filtering, *ACM Transactions on Information System*, Vol. 22, No. 1, pp. 89-115, January, 2004.

[38] R. Salakhutdinov, A. Mnih, Probabilistic Matrix Factorization, *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2007, pp. 1257-1264.

[39] H. Ma, I. King, M. R. Lyu, Learning to Recommend with Social Trust Ensemble, *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, Boston, MA, 2009, pp. 203-210.

[40] J. D. M. Rennie, N. Srebro, Fast Maximum Margin Matrix Factorization for Collaborative Prediction, *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, Bonn, Germany, 2005, pp. 713-719.

[41] R. Salakhutdinov, A. Mnih, Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo, *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, Helsinki, Finland, 2008, pp. 880-887.

[42] L. Zhang, B. Zhang, Y. Liu, Y. Gao, Z. Zhu, A Web Service QoS Prediction Approach based on Collaborative Filtering, *Proceedings of Conference on Asia-Pacific Services Computing (APSCC '10)*, Hangzhou, China, 2010, pp. 725-731.

[43] Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems, *Computer*, Vol. 42, No. 8, pp. 30-37, August, 2009.

[44] Y. Ma, S. Wang, P. C. K. Hung, C. H. Hsu, Q. Sun, F. Yang, A Highly Accurate Prediction Algorithm for Unknown Web Service QoS Values, *IEEE Transactions on Services Computing*, Vol. 9, No. 4, pp. 511-523, July-August, 2016.

[45] S. Wang, Y. Ma, B. Cheng, F. Yang, R. Chang, Multi-Dimensional QoS Prediction for Service Recommendations, *IEEE Transactions on Services Computing*, Vol. PP, No. 99, pp. 1-1, June, 2016.

Biographies

Wei Xiong is the lecturer at Hubei University of Arts and Science, China. He received his B.S. in computer science and technology from Central China Normal University in 1995, M.S. degrees in software engineering from Northwestern Poly-technical Univer- sity in 2007 and his Ph.D. degree in computer science from Wuhan University, China in 2015. His research interest

includes the development of distributed system and service computing, as well as software engineering. He participates in the IEEE Computer Society Sister Society Associate Program.

Qiong Gu is the associate professor at Hubei University of Arts and Science, China. She received the M.S. degree and the Ph.D. degree in 2006 and 2009, respectively, in China University of Geosciences. She is a member of the China Computer Federation. Her research interests include data mining, machine learning, net-mediated public sentiment, and internet of things.She is the corresponding author.

Bing Li is the professor at WuHan University. He received his Ph.D. at the Computer Science School at HUST in 2003 and worked as a post doctor researcher in The State key laboratory of software engineering (SKLSE) & School of Computer from 2003 to 2005. He is a senior member of the China computer Federation (CCF). He serves as member in three CCF technical Committee: open systems, software engineering and service computing. His research interests include software engineering and service computing.

Zhao Wu is the professor at Hubei University of Arts and Science, China. He received his B.S. in computer application from China University of Geoscience in 1999, M.S. degrees in computer application from Wuhan University of Technology in 2003 and his Ph.D. degree in computer science from Wuhan University in 2007. He is a member of the China Computer Federation. His research interests include cloud computing, service computing and internet of things.

Lei Yuan is the professor at Hubei University of Arts and Science, China. He is a member of the China Computer Federation. His research interests include cloud computing, service computing and internet of things.