The QoS Prediction of Web Service with Location Information Ensemble

Fei Peng
Institute of Acoustics, Chinese Academy of Sciences, National Network New Media Engineering Research Center, Beijing, China
Email: pengf@dsp.ac.cn

Xuewen Zeng, Haojiang Deng and Lei Liu
Institute of Acoustics, Chinese Academy of Sciences, National Network New Media Engineering Research Center, Beijing, China
Email: {zengxw, denghj, liul}@dsp.ac.cn

Abstract—**With the increasing presence and adoption of web services, accurate QoS prediction methods are becoming increasingly important. Although some QoS prediction techniques have been proposed and analyzed recently, the performance is not satisfactory, since they didn't take the relation between QoS values and users' physical locations into consideration. In order to improve the precision of web service QoS prediction, we propose a probabilistic matrix factor model that fuses the users' own properties and their physical neighbors' performance together, so as to make a comprehensive use of both the users' QoS records and their location relation. The experimental results show that our method performs better than the state-of-the-art approaches.**

Index Terms—**web service, prediction, QoS, location, PMF**

I. Introduction

A web service [1] is a programmable module with standard interface descriptions that provide universal accessibility through standard communication protocols. web services can be composed to build domain-specific applications and solutions. When service users need to implement service-oriented applications, they can use the search engines to get the services that meet their needs. They have to find the best performing one from the functionally equivalent candidates [2, 3]. However, as service users usually do not have enough information of the services' performance, it is difficult to select the optimal one. Hence, effective web service QoS prediction methods are urgently needed.

Nonfunctional performance of web services, also known as QoS (Quality-of-Service) has been considered as the key factor in service selection [4, 5]. However, it's time-consuming and resource-consuming for users to acquire all the QoS information of the services by conducting real-world web service invocations. Besides client-side evaluation, the QoS information provided by service providers or third-party communities is not reliable, because service QoS performance is highly related to the uncertain network environment and user context. Therefore, different users may observe quite different QoS performance of the same web service.

Several previous work [6-8] has applied collaborative filtering (CF) to web service recommendation. In the CF system, users propose their QoS records to the central server, and in return the server gives them useful personalized prediction results by matching together users who share similar QoS records [2]. However, it is still lack of the study on the relation between the QoS values and users' physical locations. We take some analysis on a web service dataset with location information [2]. It is obvious that some QoS properties like response time and availability are highly relate to the users' physical locations. The similarity of QoS values increases with the decrease of the distance between users, and vice versa.

According to the above description, we can model the QoS value as a mixture of both the users' own properties and their physical neighbors' performance on services. Then we can utilize both the user-service QoS matrix and the location relation network for the QoS prediction.

Inspired by the model proposed in [9], which takes the user ratings as the combination of users' own interests and their trust friends' tastes, we interpret the QoS as follows: In the users' own properties aspects, we learn the user-specific latent matrix and service-specific latent matrix by factorizing the user-service QoS matrix. For the location relation graph, we claim that the service usage records of a user's physical neighbors could reflect the user's local area's network status, so we infer and formulate the prediction problem based on the neighbors' QoS records. Then, we fuse the users and their neighbors' features together by employing a probabilistic framework [9]. Finally, we learn the user-specific and service-specific latent matrices by performing a stochastic gradient descent on the objective function. The experimental results show that our method performs better than the state-of-the-art approaches.

The remainder of this paper is organized as follows. In Section II, we provide an overview of several major approaches for QoS prediction of web services. Section

doi:10.4304/jsw.9.5.1210-1216

III presents our work on QoS prediction with location information ensemble. Experimental results are presented in Section IV, followed by the conclusion and future work in Section V.

II. Related Work

In this section, we review several major approaches for web service recommendation.

Shao et al. [12] proposed a user-based CF algorithm to predict QoS values. Zheng et al. [6] combined the user-based and item-based CF algorithm to recommend web services. However, since neither of the two approaches recognized the different characteristic between web service QoS and user ratings, the prediction accuracy of these methods was unsatisfactory. Screenath et al. [13] and Rong et al. [14] applied the idea of CF in their systems, and used MovieLens data [15] for experimental analysis. However, using the movie dataset to study web service recommendation is not convincing.

Zheng et al. [4] proposed a neighborhood-integrated matrix factorization approach for making personalized QoS value prediction. The approach fuses the neighborhood-based and model-based collaborative filtering approaches to achieve higher prediction accuracy. but the neighbors are defined as the users who have similar QoS records, not the physical neighbors described in this paper.

Zhang et al. [7] also used the matrix factorization method, and propose a model-based approach, called WSPred, for time-aware personalized QoS value prediction. Peng et al. [16] made a further step by modeling more time-effect features, and achieved better prediction accuracy.

Chen et al. [2] proposed a location-aware QoS prediction method. It employs the characteristic of QoS by clustering users into different regions. Based on the region feature, a nearest-neighbor algorithm is proposed to generate QoS prediction. However, this method just made a good start for location-aware QoS prediction, and there is sufficient room for the improvement of prediction accuracy.

In contrast, in order to improve the QoS prediction accuracy, we propose a novel probabilistic factor analysis framework, which naturally fuses the users' own properties and their physical neighbors' performance together, so as to make a comprehensive use of both the users' QoS records and their location relation.

III. Recommendation With Location Information Ensemble

The user-service QoS matrix is the only information source that traditional web service QoS prediction methods [6, 7, 8, 12, 13, 14, 17] would take into consideration. However, the location relations among users can also provide some useful information. In this section, we first present a framework for QoS prediction in Section A. Then, describe the location-aware prediction problem in Section B, and provide the solution in Sections C, D and E.

A. QoS Prediction Framework

As mentioned in Section I, the basic principle of our method is that users closely located with each other tend to have similar service invocation experience. Fig. 1 shows the QoS prediction framework of our approach, which includes the following procedures: (1) Service users provide their QoS records of service invocation and their personal meta information, especially the individual physical location information; (2) The *Input Handler* processes the input data, and store it as *User Metadata* and *User QoS Records*; (3) The system gets the *Web Service Metadata* from the UDDI registry; (4) The framework uses the provided location information to do *Location Relation Computing*; (5) The *Model Training* procedure is based on the user metadata, web service metadata, location relation and users' QoS records; (6) The *Training Result* is stored and would be updated continuously according to the model training procedure's results; (7) When service users need some prediction assists, the *Output Handler* will read the training results and recommend the optimal web services to the users.

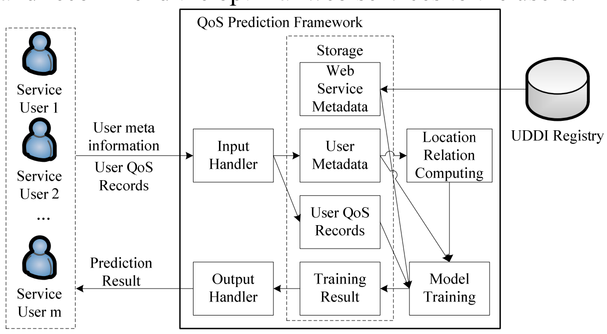
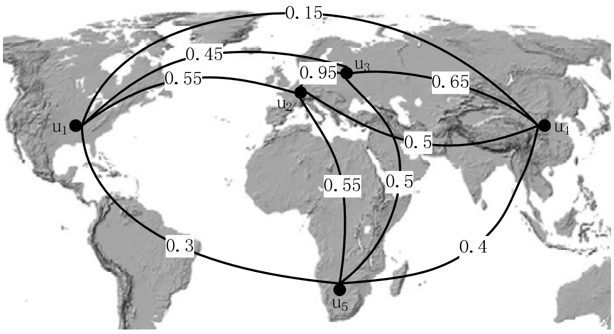


Figure 1. QoS Prediction Framework

B. Problem Description

We can build the model of prediction scenario based on the location network and the records of neighbors' QoS. The examples are presented in Fig. 2. In the location graph illustrated in Fig. 2(a), users are connected with edges, which are associated with weights in the range (0, 1]. For example, an edge with the weight R_{ij} is used to qualify the degree of distance between user u_i and user u_j in the physical location graph. The location relations in the location network are calculated by the physical longitude and latitude. Service users also provide QoS records for some services as illustrated in Fig. 2(b). In Fig. 2(b), we take the response time as the QoS values, ranging from 0 to 20 seconds. The problem we study in this paper is how to predict the missing values in the user-service QoS matrix based on the location graph and the matrix.



(a) Location Graph

	s1	s2	s3	s4	s5	s6
u1		5.611	0.8		0.146	
u2	5.93			0.202		6.039
u3			0.097			6.185
u4	0.885			7.692		
u5		0.636	0.402			8.179

(b) User-Service QoS Matrix

Figure 2. Example for Location based Prediction

C. User Features Learning

We get the user features by factorizing the user-service QoS matrix [9]. The user latent matrix $\boldsymbol{U}$ and service latent matrix $\boldsymbol{S}$ can be learned from the user-service QoS matrix $\boldsymbol{Q}$. Suppose we have m users, n services, and QoS values within the range [0, 1] in a user-service QoS matrix. We normalize the QoS values by using the function $f(x)=(x-Q_{\min})/(Q_{\max}-Q_{\min})$. Let Q_{ij} represent the QoS of user u_i for service s_j. $\boldsymbol{U}\in R^{k\times m}$ and $\boldsymbol{S}\in R^{k\times n}$ are latent matrices, with column vectors $\boldsymbol{U}_i$ and $\boldsymbol{S}_j$ representing the k-dimensional user-specific and service-specific latent feature vectors of user u_i and service s_j. According to the PMF (Probabilistic Matrix Factorization) algorithm proposed in [10], we can derive that the conditional distribution over the observed QoS is defined as:

$$p(\boldsymbol{Q}\mid\boldsymbol{U},\boldsymbol{S},\sigma_Q^2)=\prod_{i=1}^{m}\prod_{j=1}^{n}[\mathrm{N}(Q_{ij}\mid\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{S}_j,\sigma_Q^2)]^{I_{ij}^Q} \tag{1}$$

where $\mathrm{N}(x\mid\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I_{ij}^Q is the indicator function that is equal to 1 if user u_i used service s_j and equal to 0 otherwise [9]. The zero-mean spherical Gaussian priors are also placed on user and service feature vectors:

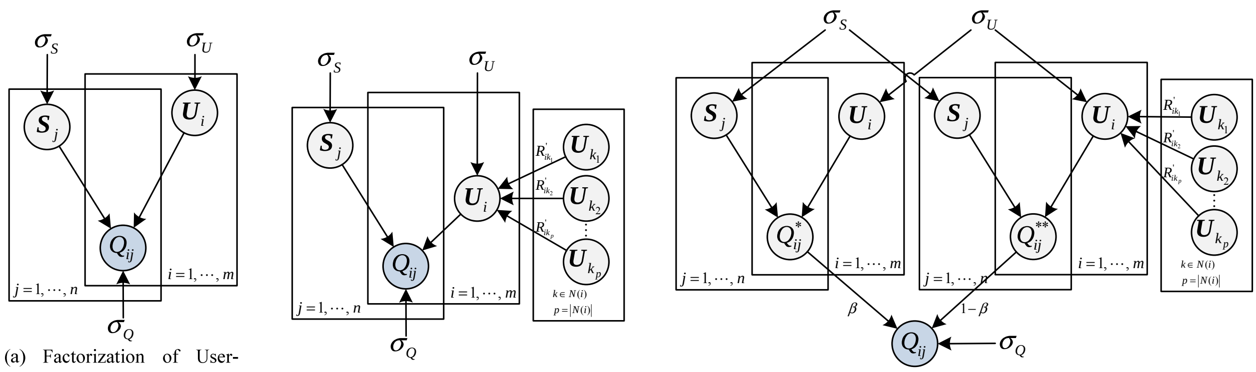
$$p(\boldsymbol{U}\mid\sigma_U^2)=\prod_{i=1}^{m}\mathrm{N}(\boldsymbol{U}_i\mid 0,\sigma_U^2\boldsymbol{I}) \tag{2}$$

$$p(\boldsymbol{S}\mid\sigma_S^2)=\prod_{j=1}^{n}\mathrm{N}(\boldsymbol{S}_j\mid 0,\sigma_S^2\boldsymbol{I}) \tag{3}$$

Hence, through a Bayesian inference, we have

$$\begin{aligned}&p(\boldsymbol{U},\boldsymbol{S}\mid\boldsymbol{Q},\sigma_Q^2,\sigma_U^2,\sigma_S^2)\\&\propto p(\boldsymbol{Q}\mid\boldsymbol{U},\boldsymbol{S},\sigma_Q^2)\,p(\boldsymbol{U}\mid\sigma_U^2)\,p(\boldsymbol{S}\mid\sigma_S^2)\\&=\prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathrm{N}\left(Q_{ij}\mid\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{S}_j,\sigma_Q^2\right)\right]^{I_{ij}^Q}\\&\times\prod_{i=1}^{m}\mathrm{N}(\boldsymbol{U}_i\mid 0,\sigma_U^2\boldsymbol{I})\times\prod_{j=1}^{n}\mathrm{N}(\boldsymbol{S}_j\mid 0,\sigma_S^2\boldsymbol{I})\end{aligned} \tag{4}$$

The graphical model of above Eq. (4) is shown in Fig. 3(a). It shows how to get the users' latent feature purely based on the user-service QoS matrix. In the next section, we will take the relation of users' locations into consideration.



(a) Factorization of User-Service Matrix (b) Recommendations by Neighbors (c) Recommendatons with Location Ensemble

Figure 3. Graphical Models

D. Prediction with Location Information

In this section, we illustrate how our location network can reflect users' local area network status, and propose a method to predict the QoS values by using the location information.

As illustrated in Fig. 2(a), the location relation graph is undirected. The nodes in the graph represent the service users, and the edges denote the location relation between them. We use a $m\times m$ matrix R to indicate the undirected graph, where m is the number of users. The matrix's element $R_{ij}\in(0,1]$ represents the weight between u_i and u_j. It also can be interpreted as how much user u_i is close to user u_j in a location graph. Note that location relation matrix is symmetric, and it can be calculated by the following equation:

$$R_{ij}=R_{ji}=\left(\frac{\cos^{-1}(\cos lat_i\cos lat_j\cos(lon_i-lon_j)+\sin lat_i\sin lat_j)}{\pi}-1\right)^2$$

lat_i and lon_i represent the real-world latitude and longitude of user u_i respectively, and are expressed in radians. lat_i is in the range $[-\pi/2, \pi/2]$, and lon_i is in the range $[-\pi, \pi]$.

As analyzed in section I, users of the same area are more reliable for us to predict the QoS of unused web services. We believe that the better QoS performance when our neighbors invoked the web services, the better performance we would have. We extract some QoS records from the dataset proposed in [2] for further explanation. As shown in Fig. 4, u_4 and u_5 are from United States, and u_{123} and u_{124} are from Singapore. The horizontal axis represents 14 distinct web services. The vertical axis represents the response time. It's obvious that the users belong to the same area tend to have quite similar QoS performance.

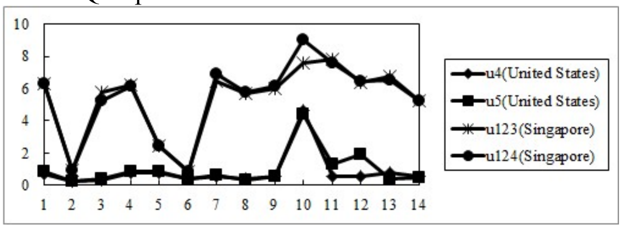


Figure 4. Example for the Connection between Location and QoS

We can take our neighbors' QoS value as a quite important reference to predict the QoS value we would have when invoking the same service. Take the case in Fig. 4 as an example, if we want to predict the QoS of u_4 when he/she uses s_1, it's obvious that we can just take the u_5's QoS record on s_1 as the predict value, and ignore u_{123} and u_{124}'s records. Whether or how much we should rely on u_5, u_{123} or u_{124}'s QoS records can be inferred from the location relation matrix R.

From the above analysis, we can use the following equation to help predicting the unknown QoS values.

$$\hat{Q}_{ij} = \frac{\sum_{k\in N(i)} R_{ik} Q_{kj}}{\sum_{k\in N(i)} |R_{ik}|} \tag{5}$$

Where $\hat{Q}_{ij}$ is the prediction value of the QoS that user u_i would have when invoking the web service s_j, Q_{kj} is the QoS that user u_k have when invoking service s_j, $N(i)$ is the set which contains the neighbors who are close to u_i. $N(i)$ is got by the equation below.

$$N(i) = \{u_k \mid u_k \in U, R_{ik} \ge \alpha\} \tag{6}$$

We can get a proper threshold α by cross-validation. In this paper, we set $\alpha = 0.65$. The denominator in Eq. (5) can be merged into the numerator, since it is the normalization term of location relation. Hence, we can rewrite the location relation equation as follow:

$$R_{ij}^{'} = \begin{cases} \dfrac{R_{ij}}{\sum_{k\in N(i)} |R_{ik}|}, & j \in N(i) \\ 0, & \text{other else} \end{cases} \tag{7}$$

Hence, the Eq. (5) can be simplified as

$$\hat{Q}_{ij} = \sum_{k\in N(i)} R_{ik}^{'} Q_{ij} \tag{8}$$

From the location network aspect, we can define the conditional distribution over the observed QoS as

$$p(Q \mid R, U, S, \sigma_Q^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathrm{N}\left(Q_{ij} \mid \sum_{k\in N(i)} R_{ik}^{'} U_k^T S_j, \sigma_Q^2\right)\right]^{I_{ij}^Q} \tag{9}$$

Hence, similar to Eq. (4), through a Bayesian inference [9], we have

$$p(U,S \mid Q,R,\sigma_Q^2,\sigma_U^2,\sigma_S^2) \propto p(Q \mid R,U,S,\sigma_Q^2)\, p(U \mid R,\sigma_U^2)\, p(S \mid R,\sigma_S^2) \tag{10}$$

In Eq. (10), we can assume that R is independent with the low-dimensional matrices U and S, then this equation can be changed to

$$\begin{aligned} & p(U,S \mid Q,R,\sigma_Q^2,\sigma_U^2,\sigma_S^2) \\ & \propto p(Q \mid R,U,S,\sigma_Q^2)\, p(U \mid \sigma_U^2)\, p(S \mid \sigma_S^2) \\ & = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathrm{N}\left(Q_{ij} \mid \sum_{k\in N(i)} R_{ik}^{'} U_k^T S_j, \sigma_Q^2\right)\right]^{I_{ij}^Q} \\ & \times \prod_{i=1}^{m} \mathrm{N}(U_i \mid 0, \sigma_U^2 I) \times \prod_{j=1}^{n} \mathrm{N}(S_j \mid 0, \sigma_S^2 I) \end{aligned} \tag{11}$$

Where $p(U \mid \sigma_U^2)$ and $p(S \mid \sigma_S^2)$ are zero-mean spherical Gaussian priors on user and service feature vectors. This equation specifies the method to predict purely based on users' location neighbors. The graphical model is shown in Fig. 3(b).

E. Location Information Ensemble

We interpreted the QoS Q_{ij} as the service s_j's performance on user u_i based on the user-service QoS matrix in section C, while in section D, we regard the QoS Q_{ij} as the performance of service s_j on user u_i's physical neighbors based on the user-service QoS matrix and the location relation network. We can see this problem more comprehensively since every user has his/her own properties and at the same time, the performance of his/her neighbors' service invoking also reflect the local area's network environment. So, we can model the QoS prediction problem more accurately by taking both of these two factors. Then, the observed QoS can be modeled as follows:

$$\begin{aligned} & p(U,S \mid Q,R,\sigma_Q^2,\sigma_U^2,\sigma_S^2) \\ & = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[\mathrm{N}\left(Q_{ij} \mid \beta U_i^T S_j + (1-\beta)\sum_{k\in N(i)} R_{ik}^{'} U_k^T S_j, \sigma_Q^2\right)\right]^{I_{ij}^Q} \\ & \times \prod_{i=1}^{m} \mathrm{N}(U_i \mid 0, \sigma_U^2 I) \times \prod_{j=1}^{n} \mathrm{N}(S_j \mid 0, \sigma_S^2 I) \end{aligned} \tag{12}$$

In Eq. (12), the user's personal properties and his/her neighbors' performance are combined by the parameter β. The parameter β controls the ratio between the two parts. We call this method Prediction with Location Ensemble (PLE). The graphical model of PLE is shown in Fig. 3(c).

The log of the posterior distribution for the prediction is given by

$$
\begin{aligned}
&\ln p(U,S\mid Q,R,\sigma_Q^2,\sigma_U^2,\sigma_S^2) = \\
&-\frac{1}{2\sigma_Q^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^Q (Q_{ij} - (\beta U_i^T S_j + (1-\beta)\sum_{k\in N(i)} R_{ik}^{'} U_k^T S_j))^2 \\
&-\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_S^2}\sum_{j=1}^{n} S_j^T S_j \\
&-\frac{1}{2}(\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^Q)\ln\sigma_Q^2 - \frac{1}{2}(ml\ln\sigma_U^2 + nl\ln\sigma_S^2) + C
\end{aligned} \tag{13}
$$

Where C is a constant that does not depend on the parameters. Maximizing the log-posterior over two latent features with hyper-parameters (i.e., the observation noise variance and prior variances) kept fixed is equivalent to minimizing the following sum-squared-errors objective functions with quadratic regularization terms [9]:

$$
\begin{aligned}
&L(Q,R,U,S) \\
&= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^Q (Q_{ij} - (\beta U_i^T S_j + (1-\beta)\sum_{k\in N(i)} R_{ik}^{'} U_k^T S_j))^2 \\
&+\frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_S}{2}\|S\|_F^2
\end{aligned} \tag{14}
$$

where $\lambda_U = \sigma_Q^2 / \sigma_U^2$, $\lambda_S = \sigma_Q^2 / \sigma_S^2$, and $\|\cdot\|_F^2$ denotes the Frobenius norm.

A local minimum of the objective function given by Eq. (14) can be found by performing stochastic gradient descent. Let us denote the prediction error, $Q_{ij} - \hat{Q}_{ij}$, be e_{ij}. We loop through all known QoS in $\boldsymbol{Q}$. For a given training case Q_{ij}, we modify the parameters by moving in the opposite direction of the gradient, yielding:

$$
\begin{aligned}
&U_i = U_i + \gamma\cdot(e_{ij}\cdot\beta S_j - \lambda_U U_i) \\
&\forall k\in N(i) \\
&\quad U_k = U_k + \gamma\cdot(e_{ij}\cdot(1-\beta)R_{ik}^{'} S_j - \lambda_U U_k) \\
&S_j = S_j + \gamma\cdot(e_{ij}\cdot(\beta U_i + (1-\beta)\sum_{k\in N(i)} R_{ik}^{'} U_k) - \lambda_S S_j)
\end{aligned} \tag{15}
$$

The meta-parameters γ (step size) and λ_U, λ_S are determined by cross-validation. We used $\gamma = 0.005$ and $\lambda_U = \lambda_S = 0.002$ in the next section. A typical number of iterations throughout the training data is 15.

IV. Experiment

In this section, we did some experiments to compare the prediction accuracy of our PLE methods with other state-of-the-art methods, and tested the effect of parameter β.

A. Dataset Description

We adopt a real-world web service QoS performance dataset for the experiment [8]. The dataset contains about 1.5 million web service invocation records of 100 web services from more than 20 countries. The QoS values include the response-time and throughput. We randomly choose 5, 10 and 20 percent QoS values of the initial training matrix to generate sparse matrices for experiments.

B. Metrics

We use two metrics, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) [11], to measure the prediction quality of our proposed approach in comparison with other collaborative filtering and location-aware prediction methods.

The metrics MAE is defined as:

$$
MAE = \frac{\sum_{(i,j)\in D_t} \left|Q_{ij} - \hat{Q}_{ij}\right|}{\left|D_T\right|} \tag{16}
$$

where Q_{ij} denotes the QoS user u_i had when invoked service s_j, $\hat{Q}_{ij}$ denotes the QoS predicted by a method, D_T denotes the training set, and $\left|D_T\right|$ denotes the size of the set. The metrics RMSE is defined as:

$$
RMSE = \sqrt{\frac{\sum_{(i,j)\in D_T} (Q_{ij} - \hat{Q}_{ij})^2}{\left|D_T\right|}} \tag{17}
$$

C. Comparision

In this section, in order to show the performance improvement of our PLE approach, we compare our method with the following approaches.

(1) PMF (Probabilistic Matrix Factorization): this method is proposed by Salkhutdinov and Minh in [10]. It only uses user-service matrix to make prediction, and is based on probabilistic matrix factorization.
(2) Loc (Location): this is the method purely uses neighbors' QoS records to make prediction. It is proposed in Section III in this paper. It is also a special case of PLE when $\beta = 0$.
(3) RBP (Region Based Prediction): this is the method proposed in [2]. It is a location-aware prediction method that clusters users into different regions, and proposed a nearest-neighbor algorithm based on the region feature.

We use different density of training data(5%, 10%, 20%) to test the algorithms. The experimental results are shown in Table I. The parameter settings of our approach are $\beta = 0.4$. The dimensions of the latent features are 10. Both the response-time and throughput values are normalized, so the range of the QoS properties is [0, 1].

TABLE I.
PERFORMANCE COMPARISONS (A SMALLER MAE OR RMSE VALUE MEANS A BETTER PERFORMANCE)

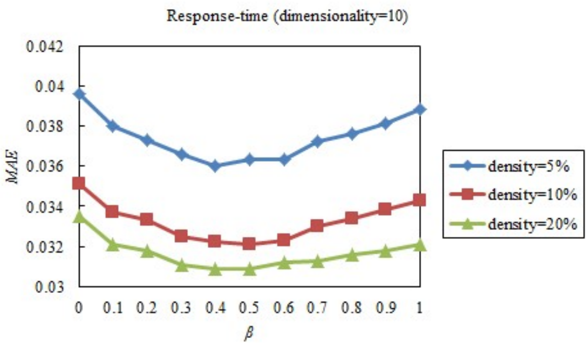
QoS Properties	Methods	Matrix Density					
		5%		10%		20%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
Response-time	PMF	0.0388	0.0866	0.0343	0.0823	0.0321	0.0789
	Loc	0.0396	0.0868	0.0351	0.0846	0.0335	0.0800
	RBP	0.0372	0.0854	0.0335	0.0812	0.0314	0.0778
	PLE	**0.0360**	**0.0839**	**0.0322**	**0.0797**	**0.0309**	**0.0761**
Throughput	PMF	0.0377	0.0842	0.0349	0.0802	0.0307	0.0752
	Loc	0.0384	0.0851	0.0356	0.0812	0.0311	0.0756
	RBP	0.0352	0.0826	0.0326	0.0794	0.0302	0.0738
	PLE	**0.0334**	**0.0813**	**0.0314**	**0.0788**	**0.0289**	**0.0725**

From table I, we can observe that the PLE methods proposed in this paper outperforms the other methods. RBP and PLE, which are location-aware prediction methods both perform better than PMF method, which ignores the location information. However, the precision of Loc method is worse than the PMF method, and we can conclude that only using neighbors' records to make prediction is not appropriate. To summarize, our PLE method achieves better performance than other methods on both MAE and RMSE. This demonstrates that our interpretation on the formation of the QoS is reasonable and feasible.

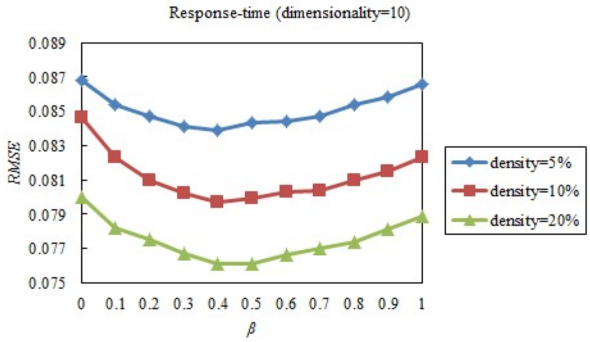
D. Impact of Parameter β

In PLE method, the parameter β balances the weight of the users' own properties and their neighbors' performance. If $\beta = 1$, we only use users' own properties, which is reflected by the user-service QoS matrix to make prediction. If $\beta = 0$, we only use the location relation graph to predict users' QoS from the neighbors nearby. In other cases, we fuse the user-service QoS matrix and the location relation graph to make probabilistic matrix factorization and predict QoS values for users.

Fig. 5 shows the impacts of parameter β on MAE and RMSE. We observe that fusing the users' own properties with their neighbors' performance greatly improves the prediction accuracy. No matter using training data with 5%, 10%, or 20% density, as β increases, the MAE and RMSE decrease (prediction accuracy increases) at first, but when β exceeds a certain threshold, the MAE and RMSE increase (prediction accuracy decreases) with further increase of the value of β. This confirms that only using the user-service QoS matrix or only using the location relation network for prediction cannot generate better performance than fusing these two factors together.



(a) MAE



(b) RMSE

Figure 5. Impact of Parameter β

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new framework to make QoS prediction for web services, which takes both the user-service QoS records and the users' physical location into consideration. The precision of our method is much better than the state-of-the-art approaches.

The physical locations are indeed an important factor when making QoS prediction for web services. However, there are still some other factors, such as the conditions of the servers, network workload that may also have influences on the QoS results. Hence, more experiments on the above issues will be conduct in our future work.

ACKNOWLEDGMENT

This work and related experiment environment is supported by the National High Technology Research and Development Program of China under Grant No. 2011AA01A102, the National Key Technology R&D Program under Grant No. 2012BAH18B04, the Key Programs of the Chinese Academy of Sciences under Grant No. KGZD-EW-103-5 and the Key Programs of the Chinese Academy of Sciences under Grant No. KGZD-EW-103-2. We are sincerely grateful to their support.

REFERENCES

[1] C. Ferris, J. Farrell, "What are Web services?," *Communications of the ACM*, vol. 46, no. 31, pp. 31, June 2003.

[2] Xi Chen, Zibin Zheng, Xudong Liu, Zicheng Huang, Hailong Sun, "Personalized QoS-Aware Web Service Recommendation an Visualization," *IEEE Transactions on Service Computing*, vol. 6, no. 1, pp. 35-47, January-March 2013.

[3] Changsong Liu, Dongbo Liu, “QoS-oriented Web Service Framework by Mixed Programming Techniques”, *Journal of Computers*, vol. 8, no. 7, pp. 1763-1770, July 2013.

[4] Tao Yu, Yue Zhang, K. –J. Lin, “Efficient Algorithms for Web Services Selection with End-to-End QoS Constrains,” *ACM Transactions on the Web*, vol. 1, no. 1, pp. 1-26, May 2007.

[5] Guangjun Guo, Fei Yu, Zhigang Chen, Dong Xie, “A Method for Semantic Web Service Selection Based on QoS Ontology,” *Journal of Computers*, vol. 6, no. 2, pp. 377-386, Feb 2011.

[6] Zibin Zheng, Hao Ma, M. R. Lyu, I. King, “WSRec: A Collaborative Filtering Based Web Service Recommender System,” In Proceedings of 2009 IEEE International Conference on Web Services, pp. 437-444, 2009.

[7] Zibin Zheng, Hao Ma, M. R. Lyu, I. King, “Collaborative Web Service QoS Prediction via Neighborhood Integerated Matrix Factorization,” *IEEE Transactions on Service Computing*, vol. pp, no. 99, pp. 1-12, 2013.

[8] Yilei Zhang, Zibin Zheng, M. R. Lyu, “WSPred: A Time-Aware Personalized QoS Prediction Framework for Web Services,” In Proceedings of IEEE 22nd International Symposium on Software Reliability Engineering, pp. 210-219, 2011.

[9] Hao Ma, I. King, M. R. Lyu, “Learning to Recommend with Social Trust Ensemble,” In Proceedings of the SIGIR ’09, pp. 203-210, 2009.

[10] R. Salakhutdinov, A. Mnih, “Probabilistic Matrix Factorization,” *Advances in Neural Information Processing Systems*, vol. 20, pp. 1257-1264, January 2008.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, January 2004.

[12] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, H. Mei, “Personalized QoS Prediction for Web Services via Collaborative Filtering,” In Proceedings of IEEE International Conference on Web Services, pp. 439-446, 2007.

[13] R. M. Screenath, M. P. Singh, “Agent-Based Service Selection,” *Web Semantics*, vol. 1, no. 3, pp. 261-279, April 2004.

[14] W. Rong, K. Liu, L. Liang, “Personalized Web Service Ranking via User Group Combining Association Rule,” In Proceedings of IEEE Conference on Web Services, pp. 445-452, 2009.

[15] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, J. Riedl, “MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System,” In Proceedings of the 8th International Conference on Intelligent User Interfaces, pp. 263-266, 2003.

[16] Fei Peng, Haojiang Deng, Lei Liu, “QoS-aware temporal prediction model for personalized service recommendation,” *Journal of XIAN University*, vol. 40, no. 4, pp. 207-213, August 2013.

[17] Xin Wan, Qimanguli Jamaliding, Toshio Okamoto, “Analyzing Learners’ Relationship to Improve the Quality of Recommender System for Group Learning Support,” *Journal of Computers*, vol. 6, no. 2, pp. 254-262, Feb 2011.

Fei Peng was born in China in 1988. He is a doctor degree candidate in Chinese Academy of Sciences. His research interests include personalized recommendation and multimedia communication.

Xuewen Zeng was born in China in 1968. He is a professor in Chinese Academy of Sciences. His research interests include multimedia communication and pattern recognition.

Haojiang Deng was born in China in 1971. He is a professor in Chinese Academy of Sciences. His research interests include multimedia communication and pattern recognition.

Lei Liu was born in China in 1980. He is an associate professor in Chinese Academy of Sciences. His research interests include multimedia communication and pattern recognition.