Collaborative Web Service QoS Prediction with Location-Based Regularization

Wei Lo, Jianwei Yin*, Shuiguang Deng, Ying Li, Zhaohui Wu
College of Computer Science & Technology
Zhejiang University
Hangzhou, China
{*spencer_w_lo, zjuyjw, dengsg, cnliying, wzh*}@*zju.edu.cn*

***Abstract*—Predicting the Quality of Service (QoS) values is important since they are widely applied to Service-Oriented Computing (SOC) research domain. Previous research works on this problem do not consider the influence of user location information carefully, which we argue would contribute to improving prediction accuracy due to the nature of Web services invocation process. In this paper, we propose a novel collaborative QoS prediction framework with location-based regularization (LBR). We first elaborate the popular Matrix Factorization (MF) model for missing values prediction. Then, by taking advantage of the local connectivity between Web services users, we incorporate geographical information to identify the neighborhood. Different neighborhood situations are considered to systematically design two location-based regularization terms, i.e. LBR1 and LBR2. Finally we combine these regularization terms in classic MF framework to build two unified models. The experimental analysis on a large-scale real-world QoS dataset shows that our methods improve 23.7% in prediction accuracy compared with other state-of-the-art algorithms in general cases.**

***Keywords*-Web Service, QoS Prediction, Matrix Factorization, Regularization**

I. INTRODUCTION

At the ongoing age of Web 2.0, Web services have enjoyed great boosts due to the strong needs of industrial companies and end users. More and more users tend to use high-quality Web services to get connected and share their ideas, and it thus makes World Wide Web more dynamic and attractive. With the growing presence of Web services, studies on Quality of Service (QoS) have recently raised the concerns of Service-Oriented Computing (SOC) researchers. A number of QoS-based paradigms have been applied to the domain of Web service selection [1], Web service automatic composition [2] and so on.

The common hypothesis of these research areas is that QoS values of all Web services are available and accurate. However, this premise sometimes may not be true for the following reasons: (1) It is time-consuming and difficult for an end user to make all Web service invocations, since there are too many Web services emerging everyday. (2) The Internet environment becomes more dynamic and vulnerable nowadays. It turns out to be impractical and impossible to acquire QoS values all the time.

* Corresponding author.

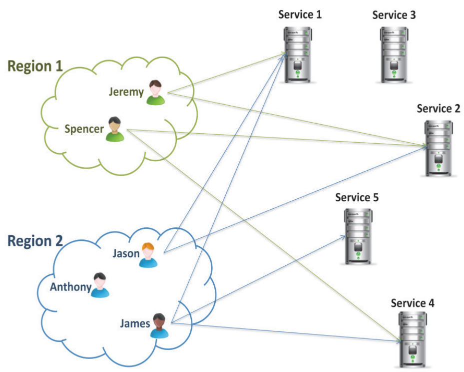
How to fill in the missing QoS values becomes crucial in this case. Inspired by the idea of user-collaboration at the Web 2.0 era, Collaborative Filtering (CF) techniques are widely used to make prediction on missing values in recommender systems [3]. These techniques have also been applied to Web services research domain recent years [4]. The main idea is to identify groups of similar users and collect their useful QoS information to a current user. The measurement of similarity is vital in the process.

Previous CF approaches on Web service studies conduct Pearson Correlation Coefficient (PCC) [3] calculations to find out a set of similar users. However, we argue that this kind of similarity measurement is inappropriate to make prediction on QoS values for the following reasons: (1) The QoS values in history records are not always accurate due to the complex nature of Internet environments. But PCC relies heavily on accurate QoS values to identify similar users groups. (2) Classic CF methods are applied to recommender systems which contain a lot of missing user ratings. Each rate represents a user's preference towards specific item. However in Web service scenarios, each QoS value is determined by the actual physical environment but not subjective judgement. This difference directly lowers the accuracy of PCC similarity calculation.

It is natural to suppose that users in the same/near area tend to enjoy the similar Web service invocation experience [5]. This idea captures our intuition since local users share the same IT infrastructures (network workloads, routers and so on), and they thus tend to receive similar objective Web service usage information. Even though the neighborhood might choose different network configurations and hence contributes diverse QoS information, we observe that these fluctuations exert far less influence than the geographical factor. Another advantage of employing geographical information is that it can contribute to the framework be more sensitive to the recent changes of QoS values. As a result, local information can be utilized to minimize the future errors with higher confidence in prediction process.

Based on the above intuition, we propose a novel collaborative QoS prediction framework with location-based regularization (LBR). We first elaborate the Matrix Factorization (MF) [6] model for missing values prediction. Then by understanding the local connectivity between Web services

978-0-7695-4752-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICWS.2012.49



(a) Real World Web Services Invocation Scenario

	S1	S2	S3	S4	S5
Jeremy	12	6			
Spencer		5.5		3	
Anthony					
Jason	2	14.2			
James	1.8			7.9	

(b) User-Service QoS Matrix

Figure 1. A Toy Example

users, LBR incorporates geographical information to identify the neighborhood rather than classic PCC manners. Based on the assumption that users in the same neighborhood tend to receive similar objective information, two location-based regularization terms are employed to revamp the classic MF framework. More specifically, in order to capture the diverse experience inside the neighborhood, local users are treated differently in the above location-based regularization terms. And then we show that the computational complexity of our proposed framework is linear in practice, and thus LBR can scale to very large datasets. The experimental analysis on a large-scale real-world QoS dataset [7] shows that LBR improves 23.7% in prediction accuracy compared with other state-of-the-art algorithms in general cases.

The major contributions of this paper include the following points:

- We elaborate the idea of utilizing the location information for the QoS prediction problem.
- We systematically discuss how to design two location-based regularization terms to capture the latent relationship inside a neighborhood.
- We revamp the Matrix Factorization model with two novel location-based regularization terms. The method on calculating two objective functions is given.

The rest of this paper is organized as follows: Section 2 describes our research problem in this paper. Section 3 details the concept of Matrix Factorization. Section 4 introduces how to revamp the Matrix Factorization model with two novel location-based regularization terms. Section 5 discusses specific issues in our framework. Section 6 presents the results of an empirical analysis. Section 7 reviews some related work on missing values prediction. Finally section 8 concludes the paper.

II. PROBLEM DESCRIPTION

The problem we study in this paper is different from traditional CF approaches in Web services domain since the latter ones only considers the user-service QoS information. In this paper, we incorporate users' geographical information to improve prediction performance. Figure 1(a) shows a real world web services invocation scenario, in which users from different geographical regions can execute Web services invocation located in other areas. This case includes two central elements in our proposed LBR framework: the neighborhood in a region and the latent connectivity among neighbors. The response time records of Web services invoked by users are shown in Figure 1(b). The gray part means this time record is unavailable.

In this toy example, there are 5 users interacting with 5 Web services. The same color shirt inside a neighborhood implies the local connectivity between users. Generally, user executes some of the Web services invocations randomly and receives them in different time intervals.

Meanwhile in the real world case, there are m users and n Web services. They contribute to an $m \times n$ user-service QoS matrix R, and each entry r_{ui} represents a QoS value recording the specific usage information of Web service i executed by the user u. The problem we study in this paper is how to predict the missing QoS values in the user-service matrix R effectively and efficiently.

III. MATRIX FACTORIZATION MODEL

Matrix Factorization model is a popular and effective tool to predict the missing values. This model maps both users and items to a joint latent factor space of a low dimensionality d, such that user-item interactions can be captured as inner products in that space. The premise behind a low-dimensional factorization technique is that there are only a few factors affecting the user-item interactions, and a user's interactive experience is influenced by how each factor applies to the user.

In this paper, we consider an $m \times n$ user-service interactive matrix R. This matrix can be approximately divided into two parts U and S with d-rank factors constraints:

$$R \approx U^T S, \tag{1}$$

where $U \in \mathbb{R}^{d\times m}$ and $S \in \mathbb{R}^{d\times n}$ with $d < \min(m, n)$ represent user feature space and service feature space respectively.

The Singular Value Decomposition (SVD) [3] technique is applied to approximate the original matrix R with U and S by minimizing the following term:

$$\min_{U,S} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} \|R_{ij} - U_i^T S_j\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ denotes the $Frobenius\ norm$. In real world cases, the original matrix R only contains a few service invocation records. This sparse issue leads to the following modification in practice:

$$\min_{U,S} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2, \tag{3}$$

where I_{ij} plays as an indicator which is equal to 1 when user u_i interacts with service s_j and is equal to 0 otherwise.

To avoid the issue of model overfitting, two regularization terms related to U and S are involved as follows:

$$\mathcal{L} = \min_{U,S} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|S\|_F^2, \tag{4}$$

Although the objective function $\mathcal{L}$ in Eq. (4) is convex in U only or S only, it is not convex in both matrixes [3]. Therefore, it is unrealistic to expect an algorithm to find the global minimum of $\mathcal{L}$. The stochastic gradient descent technique is employed to this problem as follows:

$$\begin{aligned} U_i^{'} &= U_i - \gamma_1 \frac{\partial \mathcal{L}}{\partial U_i}, \\ S_j^{'} &= S_j - \gamma_2 \frac{\partial \mathcal{L}}{\partial S_j}, \end{aligned} \tag{5}$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ are chosen as the learning rates.

IV. LOCATION-BASED REGULARIZATION

In this section, we discuss how to incorporate the geographical information as regularization terms to revamp the traditional Matrix Factorization model in detail. Subsection A introduces the neighborhood computation based on location information. And then subsection B proposes the general LBR approach. Subsection C discusses the diverse LBR method from another point of view.

A. Neighborhood Computation

The core idea of user-collaboration is to identify a set of similar users. These neighbors living in a neighborhood contribute meaningful information to improve prediction performance. How to define the term "neighborhood" becomes crucial in capturing the local connectivity.

We first calculate the Euclidean distance between users. Assuming the world is a sphere, the Euclidean distance $dist(i, j)$ between two users u_i and u_j is shown as following:

$$dist(i, j) = \sqrt{(alt(i) - alt(j))^2 + (lat(i) - lat(j))^2} \times c, \tag{6}$$

where $alt(i) \in (-180, 180]$ represents the altitude in location of u_i and $lat(i) \in (-180, 180]$ indicates the latitude in location of u_i respectively. c is a constant converting the unit of degree to meter. In this case, $c = 111, 261$.

After measuring the distance between users, the size of neighborhood needs to be identified. In practice, the neighborhood size cannot be too large since it may contain a lot of noises and thus degrade the prediction performance. For a service user i, a set of neighborhood users $G(i)$ can be defined as follow:

$$G(i) = \{j | dist(i, j) \le \theta, i \ne j\}, \tag{7}$$

where θ is a geographical threshold to control the neighborhood size. This definition implies that the neighbor relationships are symmetric since the local relationships are bidirectional. With the help of neighborhood information, we propose two location-based regularization approaches for the QoS prediction.

B. General Location-based Regularization (LBR1)

As mentioned in Section 1, it is natural to suppose that users in the same/near area tend to share similar Web service invocation experience. This intuition indicates that the difference of user feature vectors in the neighborhood should be minor. We convert this idea into the following mathematical form:

$$\min \|U_i - \frac{1}{|G(i)|}\sum_{g\in G(i)} U_g\|_F^2, \tag{8}$$

The above constraint term is used to minimize the invocation experience between a user u_i and its neighborhood to an average level. More specifically, if the neighborhood of user u_i is $G(i)$, then we can assume that u_i's invocation experience (feature vector U_i) should be close to the general experience of all neighbors in $G(i)$, which is $\frac{1}{|G(i)|}\sum_{g\in G(i)} U_g$. This representation is consistent with our intuition.

We add this regularization term in our first proposed approach to revamp the Matrix Factorization model as follow:

$$\begin{aligned} \min_{U,S} \mathcal{L}_1(R, U, S) = &\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 \\ &+ \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|S\|_F^2 \\ &+ \frac{\alpha_1}{2}\sum_{i=1}^{m}\|U_i - \frac{1}{|G(i)|}\sum_{g\in G(i)} U_g\|_F^2, \end{aligned} \tag{9}$$

where $\alpha_1 > 0$ is controlling the importance of this term. We can observe that this objective function takes all the user into consideration, and thus it is aiming at minimizing the global difference within different neighborhoods.

Similar to the traditional MF model, the global minimum of $\mathcal{L}_1$ cannot be achievable due to the nature of its inner structure [6]. We propose the gradient descent method to calculate its local minimum as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_1}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)(-S_j) + \lambda_1 U_i \\ &+ \alpha_1 (U_i - \frac{1}{|G(i)|} \sum_{g \in G(i)} U_g),\end{aligned} \tag{10}$$

$$\frac{\partial \mathcal{L}_1}{\partial S_j} = \sum_{i=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)(-U_i) + \lambda_2 S_j, \tag{11}$$

C. Diverse Location-based Regularization (LBR2)

The general location-based regularization model treats the neighbors with equal importance. However, this may not always be true in real world cases. For example, there are thousands of neighbors inside a neighborhood. Apparently those neighbors living in the same buildings exert more significant meanings in QoS information than those afar. The similarity in network configurations is somehow related to the geographical factor. Therefore, a more realistic model should assign different weights based on how similar they are.

To reweight the different neighbors, we first calculate local similarity $Lo\_Sim(i, g)$ between u_i and u_g as follow:

$$\begin{aligned}Lo\_Sim(i, g) &= 1 - \frac{1}{1 + \exp^{-dist(i,g)}} \\ &= \frac{\exp^{-dist(i,g)}}{1 + \exp^{-dist(i,g)}},\end{aligned} \tag{12}$$

where $dist(i, g)$ is defined in Eq. (6). This local similarity function Lo_Sim allows the regularization term to treat neighbors differently. If user u_i and u_g live close to each other, then the $Lo\_Sim(i, g)$ would be greater. The above equation captures this intuition and transfers the geographical distance into the local relationship.

Incorporating the above local similarity information, we propose another location-based regularization term to impose constraints in a neighborhood as follow:

$$\min \sum_{g \in G(i)} Lo\_Sim(i, g) \|U_i - U_g\|_F^2, \tag{13}$$

This constraint term has three characteristics: (1) If $Lo\_Sim(i, g)$ is large, which means the local distance is short between u_i and u_g, U_g can thus contribute more latent information to U_i. (2) This term is sensitive to those neighbors who have diverse experience. In the real world, this sensitivity plays an important role since it automatically updates the prediction consistent with the current environment. (3) This term is aiming at minimizing the latent difference between each user and its neighbors, which satisfies the need of personalized prediction.

We add this regularization term in our second proposed approach to revamp the MF model as follow:

$$\begin{aligned}\min_{U,S} \mathcal{L}_2(R, U, S) &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 \\ &+ \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|S\|_F^2 \\ &+ \frac{\alpha_2}{2} \sum_{i=1}^{m} \sum_{g \in G(i)} Lo\_Sim(i, g) \|U_i - U_g\|_F^2,\end{aligned} \tag{14}$$

where $\alpha_2 > 0$ is controlling the involvement of diverse location-based regularization factor. This objective function minimizes the global diverse difference by taking all the users into consideration.

Similar to the first model, we perform the gradient descent technique to search the local minimum of $\mathcal{L}_2$ as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_2}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)(-S_j) + \lambda_1 U_i \\ &+ \alpha_2 \sum_{g \in G(i)} Lo\_Sim(i, g)(U_i - U_g),\end{aligned} \tag{15}$$

$$\frac{\partial \mathcal{L}_2}{\partial S_j} = \sum_{i=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)(-U_i) + \lambda_2 S_j, \tag{16}$$

In the experiment section, we set $\alpha_1 = \alpha_2 = \alpha$ for simplicity.

V. DISCUSSION

The classic QoS prediction methods suffer from the "cold-start" problem, which happens frequently in user-service matrix (See Anthony's case in Figure 1(b)). Our proposed LBR approaches can solve this problem by generating user and service feature space respectively. The combination of these two parts automatically fills out all the missing values in user-service matrix.

The main computation of LBR methods is evaluating the object function $\mathcal{L}_1$ and $\mathcal{L}_2$ with their gradient parts. For LBR1 and LBR2, the computational complexities for gradients $\frac{\partial L}{\partial U}$ and $\frac{\partial L}{\partial S}$ are both $O(\rho d + |u|kd)$ and $O(\rho d)$, where ρ is the number of nonzero entries in matrix R, $|u|$ is the user population in R, k is the average population in each neighborhood and d is the dimensionality. In practice, the number of neighbors is far less than the total population of users. And it is also reasonable to assume $|u| \ll \rho$. Therefore, the total computational complexity in one iteration can be relaxed to $O(\rho d)$, which indicates that the computational time of LBR is linear with respect to the number of observations in the user-service QoS matrix. This complexity analysis shows that our proposed approaches are very efficient and can scale up to a large-scale dataset.

VI. Experiments

In this section, we conduct experiments on measuring the prediction accuracy of our LBR approaches. Our experiments are aiming at answering the following questions: (1) What are the measurement criteria? (2) How do our LBR approaches compare with other state-of-the-art collaborative filtering methods? (3) What does the impact of geographical threshold θ? (4) How does the parameter α contribute to the prediction accuracy? (5) What is the impact of the *matrix density* and *dimensionality* to our approaches?

A. Dataset Description

We have conducted our experiments on a public real world Web service QoS dataset, which is collected by Zibin Zheng et.al. It contains 1,974,675 Web service response time records. These results are collected from 339 distributed service users on 5,825 Web services. Each user record contains geographical information. More details about this dataset can be found in [7].

B. Metrics

We use the popular Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as our measurement criteria of prediction accuracy. MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i,j}|R_{ij} - \hat{R}_{ij}|, \tag{17}$$

where R_{ij} denotes the response time of Web service j observed by user i, $\hat{R}_{ij}$ is the predicted response time, and N is the number of predicted values. The MAE places the equal weight on each individual difference.

Another criterion RMSE is defined as :

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2}, \tag{18}$$

The RMSE amplifies the significance of relatively large errors. Therefore it is an efficient indicator to detect large errors.

C. Comparison

In this section, we compare our approaches with the following state-of-the-art methods.

- **UserMean**: This method uses the mean QoS value of each user to predict the missing values.
- **ItemMean**: This method employs the mean QoS value of every service to predict the missing values.
- **UPCC**: This method is a classical one that involves similar user behavior to make predictions.
- **IPCC**: This method is widely used in e-commerce scenarios. It captures similar service attributes to make predictions.
- **UIPCC**: This method [8] is a combination between UPCC and IPCC.
- **RegionKNN**: This method is proposed by Chen et al. in [5]. It incorporates the geographical information to a hybrid memory-based CF method.
- **SVD**: This method is proposed by Koren et al. in [3]. It captures the latent structure of the original data distribution.

In this section, in order to make our experiments more realistic, we randomly remove QoS values to sparse the matrix. The matrix density is thus conducted as 5%, 10%, 15%, and 20%. For example, matrix density equals 5% means we leave 5% of entries for training and the rest 95% become test ones. In this part, the above seven methods are compared with our proposed approaches given the same training and test cases. The parameter settings of our proposed approaches are $\theta = 100$, $\alpha = 0.001$, dimensionality = 10, and $\lambda_U = \lambda_S = 0.001$. Table 1 shows the comparison results, and detailed analysis on parameter tunings will be provided as follow.

From Table 1, we can find that our proposed LBR1 and LBR2 approaches obtain smaller MAE and RMSE values than others, which implies higher prediction accuracy. Meanwhile, with the increase of matrix density, the MAE and RMSE values slightly get smaller. This can be explained as more information can contribute to better prediction performance. We also observe that the MAE and RMSE values of LBR1 consistently higher than LBR2, which means assigning different weights to neighbors improves prediction performance. Among all the prediction methods, our proposed approaches improve 23.7% in MAE prediction accuracy in general cases, which indicates incorporating geographical information in Matrix Factorization model can generate better prediction performance. In the following subsections, we mainly focus on LBR2 due to the space limitation.

D. Impact of θ

In our proposed methods, the geographical threshold θ controls the size of neighborhood. If the value of θ is very small, we can only identify those neighbors living in the very short distance. If the value of θ is very large, we can incorporate neighbors to a large extent.

Figure 2 shows the impact of geographical threshold θ on the prediction accuracy. We find that as θ increases, the MAE and RMSE values decrease at first. But when θ passes over a threshold, the MAE and RMSE values soar again. This observation can be explained as when θ is smaller than a certain threshold, there are few neighbors contributing to missing QoS values predictions, which prevents user to fully absorb the wisdom of crowds. When θ is larger than a certain

Table I
ACCURACY COMPARISON(A SMALLER MAE OR RMSE VALUE MEANS A BETTER PERFORMANCE)

	Matrix Density = 5%		Matrix Density = 10%		Matrix Density = 15%		Matrix Density = 20%	
Method	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.8813	1.8601	0.8794	1.8588	0.8787	1.8586	0.8784	1.8588
IMEAN	0.7888	1.6450	0.7334	1.6198	0.6810	1.6962	0.6255	1.6078
UPCC	0.8129	1.7204	0.7412	1.6578	0.7060	1.5753	0.6834	1.5497
IPCC	0.7916	1.5563	0.7311	1.4892	0.6910	1.3843	0.6310	1.3755
UIPCC	0.7632	1.5360	0.6806	1.4442	0.6337	1.4047	0.6120	1.3864
RegionKNN	0.6782	1.5319	0.6429	1.4031	0.6021	1.3784	0.5722	1.3810
SVD	0.5691	1.5022	0.5587	1.3849	0.5437	1.3615	0.5302	1.3495
LBR1	0.5673	1.4529	0.5532	1.3911	0.5376	1.3701	0.5058	1.3396
LBR2	**0.5389**	**1.4130**	**0.5292**	**1.3481**	**0.5180**	**1.3260**	**0.4941**	**1.3147**

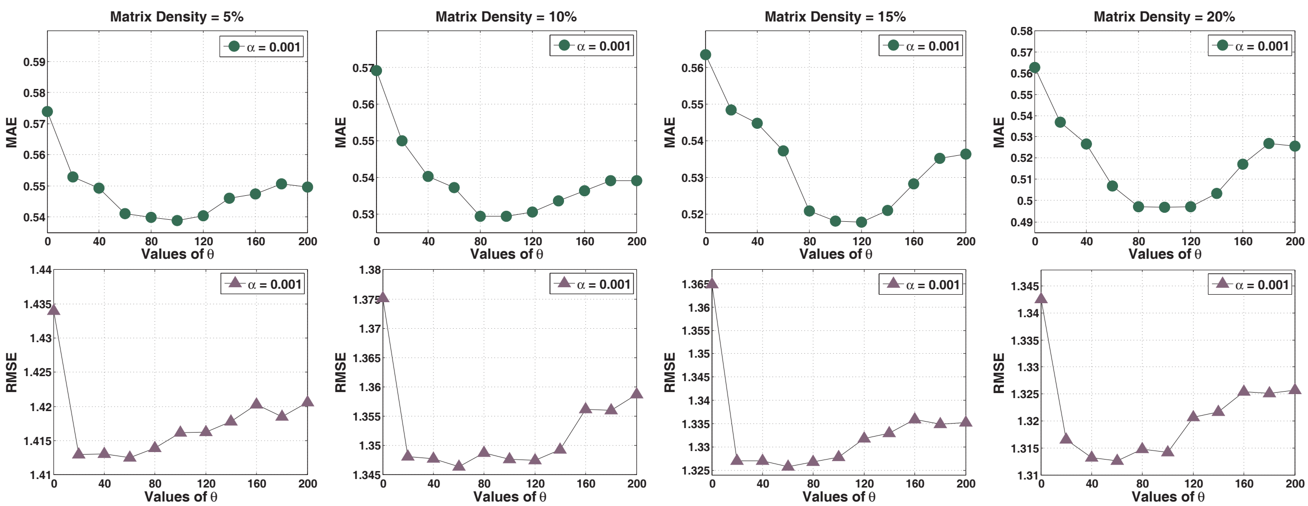


Figure 2. Impact of θ

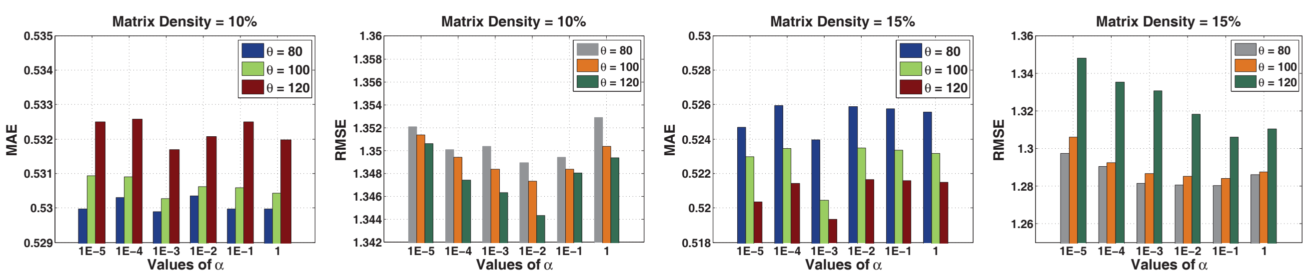


Figure 3. Impact of α

threshold, the neighbors contain much noise even though the sample size is large enough. These two cases will turn out to lower the prediction performance.

We can also observe that no matter what the matrix density is, θ around 100 contributes to the smallest MAE values, which means θ meets a threshold in this dataset. At the same time, the smallest RMSE values in all matrix density settings happen when θ is around 60. The optimal thresholds of MAE and RMSE are different since they are different criteria focusing on different aspects. The observation shows that choosing an appropriate size of neighborhood can achieve a better result.

E. Impact of α

In our LBR approaches, the parameter α controls how much the regularization terms influence to the objective functions. In the extreme case, if we set α too small, LBRs mainly focus on general MF model and underestimate the importance of location-based regularization terms. Meanwhile if we set α too large, the geographical information dominates the prediction process, which would potentially harm prediction performance. In other cases, we can tune α to combine matrix factorization and location-based reg-

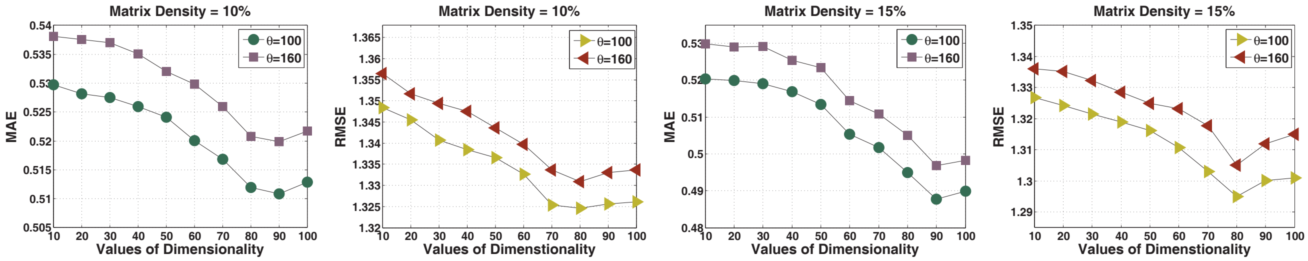


Figure 4. Impact of Dimensionality

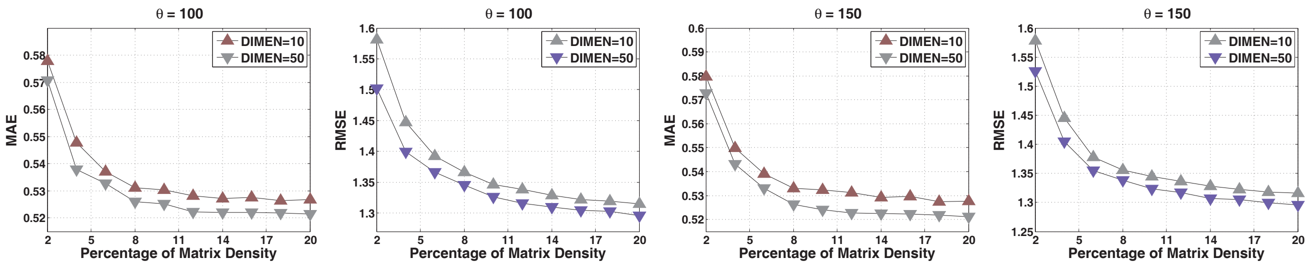


Figure 5. Impact of Matrix Density

ularization appropriately. In this section, we analyze how the changes of α can affect the prediction accuracy. We set *dimensionality* = 10. Meanwhile we tune the parameter θ from 80 to 120 and density from 10% to 15%.

Figure 3 shows that as α increases, the MAE and RMSE values decrease (the prediction accuracy increases) at first. When α goes above a certain threshold, the MAE and RMSE values increase slightly. We observe that the threshold in MAE is around $\alpha = 10^{-3}$ in all cases. Meanwhile α around 10^{-2} is the threshold in RMSE criterion. The existence of turning point confirms our intuition that solely using matrix factorization and geographical connectivity cannot contribute to better prediction accuracy rather than an appropriate combination. Moreover, we find that our framework is quite steady since it remains the similar trends in all configurations with respect to different criteria.

F. Impact of Dimensionality

In our proposed methods, dimensionality determines how many factors involve to matrix factorization. To study the impact of dimensionality, We set $\alpha = 0.001$. And we also tune the parameter θ as 100 and 160, and density from 10% to 15%.

Figure 4 shows that with the increase of dimensionality, the values of MAE and RMSE dramatically decrease at first. However, the values of MAE and RMSE increase when dimensionality goes above a certain threshold (90 for MAE and 80 for RMSE). These phenomena can be explained from two reasons: 1) The improvement of prediction accuracy confirms the intuition that a relative larger dimension generate better results. 2) When the dimensionality surpasses a certain threshold, it may cause the issue of overfitting, which turns out to degrade the prediction performance.

G. Impact of Matrix Density

To study the impact of the matrix density on MAE and RMSE, we set $\theta = 100$ and $\theta = 150$ respectively, and vary the density percentage from 2 to 20. Also we set $\alpha = 0.001$.

Figure 5 shows that when matrix density increases from 2% to 8%, both MAE and RMSE decrease sharply, which means prediction accuracy is improved significantly. With the further increase in matrix density, both MAE and RMSE decrease slowly. It shows that with more entries contributing to the training phase, our proposed approaches perform much better.

VII. Related Work

QoS has been widely studied in a number of literatures [4], [9]. The domain of Web service selection [1], and Web service composition [2] is also related to QoS research. In these areas, a popular premise is that QoS values are already known and accurate. However in real world cases, this is not always true since the complex nature of Internet environments. The demand on predicting the missing QoS values is becoming strong in SOC research domain.

Collaborative Filtering methods have been widely applied to predict the user preference in modern recommender systems [3]. Generally speaking, CF techniques can be classified into two categories: memory-based and model-based. Memory-based algorithms employ statistical analysis for computing similarity, which is used to define a set of similar users. These approaches often use Pearson Correlation Coefficient (PCC) method and Vector Space Similarity

(VSS) method [6] to compute similarity. Meanwhile, model-based approaches use history records to build a sophisticated model on latent user or item characteristics. Koren et al. [3] propose a SVD-based model to track the temporal behavior throughout the life span of the data. This matrix factorization model enjoys the great success on Netflix[1] dataset, and it thus becomes the main-stream approach in CF. The premise behind matrix factorization is that there are only a small number of factors influencing the user preference. Prediction performance with high-accuracy methods can be achieved by monitoring these latent factors delicately.

However there is limited work employing the Collaborative Filtering idea to make Web service QoS predictions. This is mainly because the lack of real world QoS values are available for academic use. Recently, Zheng et al. [7] has released a large-scale real-world QoS dataset for research usage, and it raises the concerns of SOC researchers. Shao et al. [10] propose a user-based Collaborative Filtering algorithm to predict the QoS of web services from consumers' preference. Zheng et al. [8] present a hybrid approach combining user-based and item-based approach to predict the QoS values. Liu et al. [11] extend the personalized QoS prediction approach to select the best-fit service. However these research works underestimate the importance of geographical information. Chen et al. [5] support the idea that local knowledge contributes to the prediction performance. They propose a region-based hybrid Collaborative Filtering algorithm to predict the QoS values of services. However they fail to capture the latent local relationship between users and Web services due to the use of PCC. Moreover, they cannot solve the "cold-start" problem which happens frequently in Web service invocation scenarios.

VIII. CONCLUSION AND FUTURE WORK

Based on the intuition that geographical neighborhood users share the similar Web services invocation experience, we proposed a unified Matrix Factorization framework with two novel location-based regularization terms to make missing QoS values prediction. Our LBR approaches focused on capturing geographical connectivity to identify similar users. The experimental analysis showed that our methods improve 23.7% in prediction accuracy compared with other state-of-the-art algorithms in general cases.

In this paper, we only consider user feature vector. In fact, the Web services located in the same place may provide similar facilities. Besides, social tags can be involved to identify the similarity between Web services. Therefore, we can corporate the similar Web service feature vectors to design a unified framework. More experimental studies on identifying useful heterogeneous neighborhood and designing different kinds of regularization will be conducted in our future work.

[1] http://www.netflix.com

ACKNOWLEDGMENT

This research is partially supported by National Science and Technology Supporting Program of China under grant of No.2012BAH06F02, National Key Science and Technology Research Program of China under grant of No.2011ZX01039-001-002, Research Fund for the Doctoral Program by Ministry of Education of China under grant of No.20110101110066, Science and Technology Program of Zhejiang Province of China under grant of No.2011C14004.

REFERENCES

[1] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," *Proceedings of the 18th international conference on World wide web*, 2009.

[2] J. El Hadad, M. Manouvrier, and M. Rukoz, "Tqos: Transactional and qos-aware selection algorithm for automatic web service composition," *Services Computing, IEEE Transactions on*, 2010.

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

[4] Y. Zhang, Z. Zheng, and M. Lyu, "Wsexpress: A qos-aware search engine for web services," *Web Services (ICWS), 2010 IEEE International Conference on*, 2010.

[5] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," *Web Services, 2010 IEEE International Conference on*, 2010.

[6] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, 2010.

[7] Z. Zheng, Y. Zhang, and M. Lyu, "Distributed qos evaluation for real-world web services," *Web Services (ICWS), 2010 IEEE International Conference on*, 2010.

[8] Z. Zheng, H. Ma, M. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," *Web Services, 2009. IEEE International Conference on*, 2009.

[9] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for qos-based web service composition," *Proceedings of the 19th international conference on World wide web*, 2010.

[10] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," *Web Services, 2007. IEEE International Conference on*, 2007.

[11] H. Liu, F. Zhong, and B. OuYang, "A web services selection approach based on personalized qos prediction," in *Parallel and Distributed Computing (ISPDC), 2011 10th International Symposium on*. IEEE, 2011.

[12] H. Sun, Z. Zheng, J. Chen, and M. Lyu, "Nrcf: A novel collaborative filtering method for service recommendation," *Web Services, 2011 IEEE International Conference on*, 2011.