An Extended Matrix Factorization Approach for QoS Prediction in Service Selection

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*—With the growing adoption of Web services on the World Wide Web, the issue of QoS-based service selection is becoming important. A common hypothesis of previous research is that the QoS information to the current user is supposed all known and accurate. However, the real case is that there are many missing QoS values in history records. To avoid the expensive and costly Web services invocations, this paper proposes an extended Matrix Factorization (EMF) framework with relational regularization to make missing QoS values prediction. We first elaborate the Matrix Factorization (MF) model from a general perspective. To collect the wisdom of crowds precisely, we employ different similarity measurements on user side and service side to identify neighborhood. And then we systematically design two novel relational regularization terms inside a neighborhood. Finally we combine both terms into a unified MF framework to predict the missing QoS values. To validate our methods, experiments on real Web services data are conducted. The empirical analysis shows that our approaches outperform other state-of-the-art methods in QoS prediction accuracy.**

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

I. INTRODUCTION

Web services are self-describing software applications that can be promoted, located, and used across the Internet based on a set of standards such as SOAP, WSDL and UDDI [1]. At the ongoing age of Web 2.0, Web services are widely used in business organizations to operate their processes by automatic selection. The selection process aims at discovering services which best meet their requirements in terms of QoS, i.e. performance, throughput, reliability, availability, trust, etc. With the growing number of alternative Web services that provide the same functionality but differ in quality properties, the demand of QoS-based service selection is becoming strong.

Figure 1 illustrates a real-world example of QoS-based service selection scenario. A general process contains several abstract services, which correspond to a number of concrete services located in service repositories respectively. The goal of user Jeremy is to select suitable candidate services from repositories to fit in each abstract service. He achieves this goal by maximizing his personal preference to this process.

* Corresponding author.

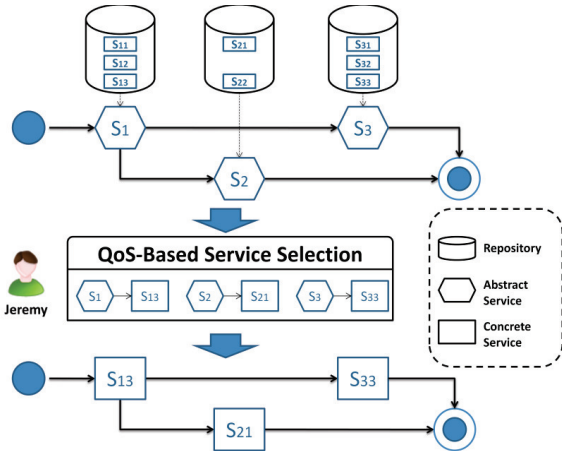

Figure 1. QoS-based Service Selection Scenario

There have been a number of works on QoS-based service selection recent years [2], [3]. A common hypothesis of these works is that all the Web services QoS values are known and accurate. However, this may not be true due to the following reasons: (1) Most Web services are owned by the companies and organizations. Therefore these services need to be charged. It is too costly to invoke all these services for scholars to get QoS values. (2) It is time-consuming for an end user to make all Web service invocations, since there are too many Web services emerging everyday. (3) The Internet environment becomes more dynamic and vulnerable nowadays. It turns out to be impractical to acquire accurate QoS values all the time. As a result, there are a large number of missing QoS values in history records.

Table I
RESPONSE TIME OF WEB SERVICES

	S_1	S_2	S_3	S_4
Jeremy	12s	*Null*	*Null*	3s
Anthony	*Null*	*Null*	*Null*	*Null*
James	*Null*	7s	*Null*	2s
Wade	1s	2s	*Null*	2s

Table 1 describes this phenomenon in an intuitive way. Each value represents the response time of Web service invoked by the corresponding user. A notation *Null* means

978-0-7695-4753-4/12 $26.00 © 2012 IEEE
DOI 10.1109/SCC.2012.36

this QoS value is missing and thus not available. From this table we can observe two points: (1) In the real world, this QoS record matrix is very sparse and contains a large number of missing values. (2) To the extreme case, user Anthony does not invoke any service in this pool. This leads to the cold-start problem, and thus it is unable to recommend any services for Anthony to choose. To execute QoS-based service selection, a necessary preprocess is to fill in the missing QoS values.

Inspired by the idea of user-collaborations in recommender systems, we propose an extended Matrix Factorization (EMF) framework to address the problem of missing QoS values prediction. The core idea of user-collaboration is to identify the neighborhood for the current user. Therefore we first apply classic Pearson Correlation Coefficient (PCC) algorithm to calculate the similarity relationship among users. After obtaining the user similarity matrix, we employ the modified Top-K technique to filter those dissimilar ones for each user, and thus the neighborhood is achieved at this step. Meanwhile, we assume that each user in the same neighborhood shares the similar Web services interactive experience. Based on this intuition, a user-based regularization term is proposed to minimize the difference of latent features among users in the same neighborhood. Finally, we incorporate this term as a constraint to revamp the traditional MF model. The MF model has been widely used as a powerful tool to solve the problem of missing values prediction [4]. By modifying the inner structure of MF framework, we extend this model to fit in Web services invocation scenarios.

This collaborative idea can be applied to the service side. However, the similarity measurement is different due to the varied scales of Web services usage information. We take this effect into account and recompose the PCC algorithm to find out a set of similar services. After identifying the neighborhood of each service, we add a service-based regularization term in an MF model to improve prediction performance.

In previous research, most works have done on either side to make QoS values predictions [5], [6]. The reason is that they do not treat both sides equally. Also the combination of both sides suffers from the high computational costs problem in these works. In this paper, we equally treat both sides and combine two corresponding regularization terms into a unified framework. Empirical analysis shows that this combination is essential in achieving high prediction performance. Meanwhile we observe that this framework is very efficient and can scale to very large datasets.

The contributions of this paper are three-fold:

- We employ different similarity calculation techniques to identify the neighborhood of user side and service side, respectively.
- We propose two corresponding regularization terms and elaborate how they work to capture the relationship in the neighborhood.
- We combine the above regularization terms into a unified Matrix Factorization framework to improve prediction performance.

The rest of this paper is organized as follows: Section 2 illustrates our proposed EMF framework for QoS-based service selection. Section 3 details the concept of low-rank Matrix Factorization model. Section 4 focuses on how to incorporate relational regularization terms to revamp the MF model. Section 5 presents the results of an empirical analysis. Section 6 reviews the related work. Finally section 7 concludes the paper.

II. ARCHITECTURE

Figure 2 illustrates the general service selection framework of our proposed EMF framework. First we collect QoS information from heterogeneous data sources. By fully utilizing the wisdom of crowds, we generate the similarity matrix by different calculations techniques. Then we input the source information and similarity relationship into our EMF engine and thus predict the missing QoS values. After observing a large number of QoS information, we filter some unrelated services to decrease the number of services for QoS-based selection. The final step automatically selects high-quality services and recommends them to the public. Due to the space limitation, we mainly introduce how the EMF QoS prediction framework works in this paper.

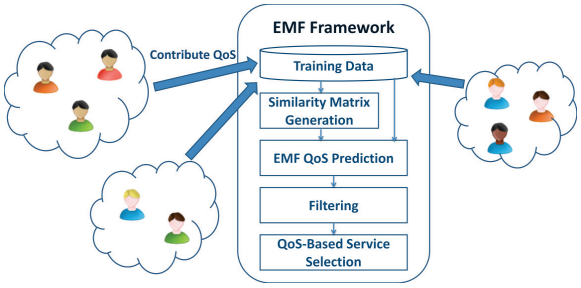

Figure 2. EMF Framework

III. LOW-RANK MATRIX FACTORIZATION

In the real world case, there are m users and n Web services. They contribute to an $m \times n$ user-service 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. As mentioned in Section 1, R is very sparse, and it thus contains a lot of missing QoS values. The problem we study in this paper is how to predict the missing QoS values of the user-service matrix R effectively and efficiently.

To address this problem, the low-rank Matrix Factorization [7] model is widely used. MF factorizes the user-item matrix and hence makes accurate prediction. The goal is to map 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 MF 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 affects the user.

In this paper, we focus on 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) [4] 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 follow:

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

where λ_1 and λ_2 are the learning rates. The optimization problem in Eq. (4) minimizes the sum-of-squared-errors objective function with quadratic regularization terms. The above form is widely used in the domain of recommender systems.

IV. RELATIONAL REGULARIZATION

In this section, we will detail how to incorporate the relational regularization terms to revamp the traditional Matrix Factorization model. Subsection A first focuses on employing users relationship in regularization part. Then subsection B discusses how to extend this idea to service side. Subsection C introduces the fusion of users and services relationship into a unified MF model.

A. User Regularization (EMF_U)

The core idea of user-collaboration is to find out a set of neighbors who behave very similarly to the current user. Similarity computation plays a vital role at the first step of prediction process.

1) Generating User Neighborhood: Existing papers on QoS values prediction use Pearson Correlation Coefficient (PCC) to compute the similarity relationship between users [5], [8]. We apply this elegant algorithm between user u_i and u_j as follow:

$$U\_Sim(i,j) = \frac{\sum_{s \in S} (r_{is} - \bar{r_{u_i}})(r_{js} - \bar{r_{u_j}})}{\sqrt{\sum_{s \in S} (r_{u_{is}} - \bar{r_{u_i}})^2} \sqrt{\sum_{s \in S} (r_{u_{js}} - \bar{r_{u_j}})^2}}, \tag{5}$$

where $S = S_{u_i} \bigcap S_{u_j}$ is the set of services both invoked by different user u_i and u_j, and $\bar{r_{u_i}}$ represents the average QoS values of different services invoked by user u_i. The U_Sim is in the range of [-1,1], where a higher value indicates higher user similarity.

After calculating the similarity relationship between users, a set of Top-K users is chosen as the neighborhood for the target user. The process of identifying the size of neighborhood is crucial to the prediction accuracy, since the dissimilar neighbors contribute useless information to make predictions and thus potentially harm this prediction accuracy. In order to choose an appropriate size, we revamp traditional Top-K algorithm to remove the dissimilar users in neighborhood as follow:

$$TU(i) = \{k | k \in \text{Top\_K\_U}(i), U\_Sim(i,k) > 0\}, \tag{6}$$

where Top_K_U(i) represents a set of the Top-K similar users ranking by similarity to user u_i, and $U\_Sim(i,k)$ is defined in Eq. (5). This modification reduces the number of dissimilar users, and it hence generates an appropriate size of neighborhood.

2) Capturing User Relationship: In practice, the interactive experience inside a neighborhood should be somehow similar. This captures our intuition since neighbors are very likely using the similar network infrastructure (network workloads, routers and so on). As a result, they contribute the similar patterns of Web Services usage information and thus are defined as neighbors.

Based on this intuition, we propose the following user relational regularization term:

$$\min \|U_i - \frac{1}{|TU(i)|} \sum_{f \in TU(i)} U_f\|_F^2, \tag{7}$$

where U_i is the feature vector of user u_i. The meaning of this term is used to minimize the interactive experience between a user u_i and its neighbors $TU(i)$. Given the neighborhood for user u_i, we assume that u_i's feature vector is similar to the average feature vector of all its neighbors in this pool.

The above constraint term holds the premise that every user's experience is close to average level of neighborhood. However, this process treats every neighbor with equal importance, which may not be true in real world cases. For example, there are thousands of neighbors inside a neighborhood. Apparently those neighbors with higher relevance should be treated more serious than others. In order to

reweight the importance inside a neighborhood, we thus change the user constraint in Eq. (7) as follow:

$$\min \|U_i - \sum_{f \in TU(i)} PU_{if} \cdot U_f\|_F^2, \tag{8}$$

This term combines different weights into the average feature vector of all neighbors. And PU_{if} is a normalized weight defined as follow:

$$PU_{if} = \frac{U\_Sim(i,f)}{\sum\limits_{g \in TU(i)} U\_Sim(i,g)}, \tag{9}$$

We incorporate this user regularization term to revamp the traditional Matrix Factorization model as follow:

$$\begin{aligned} \min_{U,S} \mathcal{L}_1(R,U,S) = & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 \\ & + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|S\|_F^2 \\ & + \frac{\alpha_1}{2}\sum_{i=1}^{m}\|U_i - \sum_{f \in TU(i)} PU_{if} \cdot U_f\|_F^2, \end{aligned} \tag{10}$$

where $\alpha_1 > 0$ is controlling the importance of this term in EMF model. 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.

Although the objective function $\mathcal{L}_1$ in Eq. (10) is convex in U only or S only, it is not convex in both matrixes [4]. Therefore, it is unrealistic to expect an algorithm to find the global minimum of $\mathcal{L}_1$. The gradient descent method is employed to find the local minimum as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial U_i} = & \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)(-S_j) + \lambda_1 U_i \\ & + \alpha_1(U_i - \sum_{f \in TU(i)} PU_{if} \cdot U_f), \end{aligned} \tag{11}$$

$$\frac{\partial \mathcal{L}_1}{\partial S_j} = \sum_{i=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)(-U_i) + \lambda_2 S_j, \tag{12}$$

B. Service Regularization (EMF_S)

The idea of utilizing the wisdom of crowds can be also applied to the service side. However the measurement of services similarity is different from the user ones. The reason is that the QoS information of each service is greatly affected by the network situation of those users. For example, due to the reason of network security and bandwidth constraints, the response time of all services invoked by user Jeremy is higher than the average level of other users. Meanwhile user Wade enjoys the high speed of network bandwidth without constraints. Thus the response time of all services is lower than others. From this example we can learn that the QoS values contains diverse knowledge to each service. In order to calculate the similarity relationship among services precisely, we slightly modify the PCC algorithm to fit in the service side as follow:

$$S\_Sim(i,j) = \frac{\sum_{u \in U}(r_{ui} - \bar{r_u})(r_{uj} - \bar{r_u})}{\sqrt{\sum_{u \in U}(r_{ui} - \bar{r_u})^2}\sqrt{\sum_{u \in U}(r_{uj} - \bar{r_u})^2}}, \tag{13}$$

where $U = U_{s_i} \bigcap U_{s_j}$ is the set of users who both have invoked service s_i and s_j. $\bar{r_u}$ means the average QoS values of service invoked by u. In Eq. (13), we remove the impact of different QoS scale by using $(r_{ui} - \bar{r_u})$.

After understanding the similarity relationship among services, we filter those dissimilar service neighbors by utilizing Top_K strategy as follow:

$$TS(i) = \{k | k \in \text{Top\_K\_S}(i), S\_Sim(i,k) > 0\}, \tag{14}$$

Similarly, we assume the difference in latent features of each service inside a neighborhood tends to minor. We transfer this assumption into a service regularization term and involve it to revamp traditional MF model as follow:

$$\begin{aligned} \min_{U,S} \mathcal{L}_2(R,U,S) = & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 \\ & + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|S\|_F^2 \\ & + \frac{\alpha_2}{2}\sum_{j=1}^{n}\|S_j - \sum_{h \in TS(j)} PS_{jh} \cdot S_h\|_F^2, \end{aligned} \tag{15}$$

where $\alpha_2 > 0$ and PS_{jh} is a normalized weight defined as follow:

$$PS_{jh} = \frac{S\_Sim(j,h)}{\sum\limits_{l \in TS(j)} S\_Sim(j,l)}, \tag{16}$$

A local minimum of $\mathcal{L}_2$ is can be calculated by performing the gradient descent method as follows:

$$\frac{\partial \mathcal{L}_2}{\partial U_i} = \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)(-S_j) + \lambda_1 U_i, \tag{17}$$

$$\begin{aligned} \frac{\partial \mathcal{L}_2}{\partial S_j} = & \sum_{i=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)(-U_i) + \lambda_2 S_j \\ & + \alpha_2(S_j - \sum_{h \in TS(j)} PS_{jh} \cdot S_h), \end{aligned} \tag{18}$$

C. Fusion Regularization (EMF_F)

In previous research, most works have done on one side to make predictions [5], [6]. There are 2 reasons to explain this phenomenon: (1) These works do not treat user side and service side symmetrically, and both sides thus could not be combined into a unified model. (2) The complexities of the previous algorithms lowers the possibility of combination.

From the previous sections, we can see each step in our proposed EMF framework is symmetrically: similarity calculation, neighborhood generation, regularizaion combination. Taking the different natures between both sides into consideration, we employ two algorithms to measure the similairity. Also we choose different Top_K values on both

sides due to the varied population. We later show that our EMF framework is very efficient since the computation time is linear with respect to the matrix density.

To sum up, we propose a unified framework by fusing user-side and service-side regularization terms as follow:

$$\begin{aligned}\min_{U,S} \mathcal{L}_3(R,U,S) = &\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)^2 \\ &+ \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|S\|_F^2 \\ &+ \frac{\alpha_1}{2}\sum_{i=1}^{m}\|U_i - \sum_{f\in TU(i)} PU_{if}\cdot U_f\|_F^2 \\ &+ \frac{\alpha_2}{2}\sum_{j=1}^{n}\|S_j - \sum_{h\in TS(j)} PS_{jh}\cdot S_h\|_F^2,\end{aligned} \tag{19}$$

A local minimum of $\mathcal{L}_3$ is can be calculated by performing the gradient descent method as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_3}{\partial U_i} = &\sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T S_j)(-S_j) + \lambda_1 U_i \\ &+ \alpha_1(U_i - \sum_{f\in TU(i)} PU_{if}\cdot U_f),\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_3}{\partial S_j} = &\sum_{i=1}^{m} I_{ij}(R_{ij} - U_i^T S_j)(-U_i) + \lambda_2 S_j \\ &+ \alpha_2(S_j - \sum_{h\in TS(j)} PS_{jh}\cdot S_h),\end{aligned} \tag{21}$$

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

D. Complexity Analysis

The main computation of EMF framework is evaluating the object function with their gradient parts. In this part, We focus on the EMF_F approach since it is a combination between other methods. For EMF_F, the computational complexities for gradients $\frac{\partial L}{\partial U}$ and $\frac{\partial L}{\partial S}$ are both $O(\rho d + |u|kd)$ and $O(\rho d + |s|kd)$, where ρ is the number of nonzero entries in matrix R, $|u|$ and $|s|$ are the population of both sides, 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 services. And it is also reasonable to assume both population is less than the total density of matrix R. Therefore, the total computational complexity in one iteration can be relaxed to $O(\rho d)$, which indicates that the computational time of EMF_F is linear with respect to the number of observations in the user-service QoS matrix. This complexity analysis shows that our proposed framework is very efficient and can scale up to a large-scale dataset.

The process of our proposed EMF approaches is quite general. It only requires the information from the QoS matrix but not other heterogenous data sources. As a result, they can be extended to other QoS invocation scenarios without any modification.

V. EXPERIMENTS

In this section, we conduct experiments on measuring the prediction accuracy of our EMF approaches. Our experiments are aiming at answering the following questions: (1) What is the measurement criterion? (2) How does our proposed EMF framework compare with other state-of-the-art methods? (3) What is the impact of TOP_K thresholds on both sides? (4) 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. More details about this dataset can be found in [9].

B. Metric

We use the popular Mean Absolute Error (MAE) as our measurement criterion of prediction accuracy. MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i,j}|R_{ij} - \hat{R}_{ij}|, \tag{22}$$

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.

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.
- **SVD**: This method is proposed by [4] in Collaborative Filtering area. 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 from 5% to 20% with ascending rate as 5%. Matrix density equals 5% means we leave 5% of entries for training and the rest 95% become test ones. In this part, the above six methods are compared with our proposed EMFs given the same training

Table II
ACCURACY COMPARISON(A SMALLER MAE VALUE MEANS A BETTER PERFORMANCE)

	Density = 5%	Density = 10%	Density = 15%	Density = 20%	Density = 25%	Density = 30%
Method	MAE	MAE	MAE	MAE	MAE	MAE
UMEAN	0.8813	0.8794	0.8787	0.8784	0.8753	0.8749
IMEAN	0.7888	0.7334	0.6810	0.6255	0.6078	0.5910
UPCC	0.8129	0.7412	0.7060	0.6834	0.6697	0.6504
IPCC	0.7916	0.7311	0.6910	0.6310	0.5937	0.5563
UIPCC	0.7632	0.6806	0.6337	0.6120	0.5736	0.5486
SVD	0.5691	0.5587	0.5437	0.5302	0.5222	0.5205
EMF_U	0.5571	0.5432	0.5391	0.5176	0.5070	0.4858
EMF_S	0.5409	0.5329	0.5192	0.5091	0.4976	0.4831
EMF_F	**0.5189**	**0.5103**	**0.5022**	**0.4981**	**0.4718**	**0.4632**

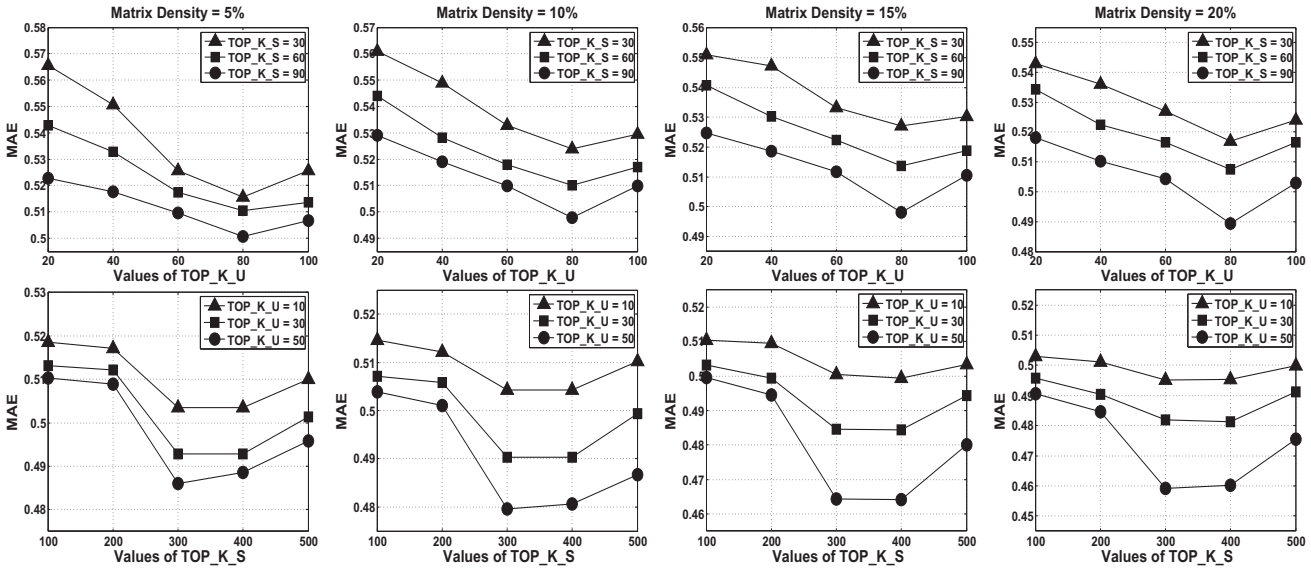

Figure 3. Impact of Neighborhood Size

and test cases. The parameter settings of our proposed approaches are TOP_K_U=60, TOP_K_S=300, α=0.001, dimensionality=10. Table 2 shows the comparison results, and detailed analysis of parameter tunings will be provided in the following subsections.

From Table 2, we can observe that our proposed EMF approaches obtain smaller MAE values than others, which implies higher prediction accuracy. Meanwhile, the MAE values slightly get smaller with the increase of matrix density. This can be explained as more information can contribute to better prediction performance. We also find that EMF_F consistently performs better than EMF_U and EMF_S, which means a combination of both sides can generate a better prediction result. We observe that the MAE values of EMF_S are lower than EMF_U in general cases. The reason is that the number of services is approximately 6 times to the users, and hence more useful information is collected on service side. Among all the prediction methods, our proposed approaches generally achieve lower prediction errors, which indicates incorporating relational constraints in Matrix Factorization can generate better prediction accuracy. In the following subsections, we mainly focus on EMF_F due to the space limitation. Meanwhile we treat EMF_F as EMF for short.

D. Impact of Neighborhood Size

In our EMF approach, the parameter TOP_K_U and TOP_K_S directly control the size of neighborhood respectively. In the extreme case, if we set these values too small, EMF only listen to the advice from few neighbors. If we set these values too high, EMF generates a large size of neighborhood which contains varied noises.

Figure 3 shows the impact of TOP-K values on the prediction accuracy. On the user side, we observe that as TOP_K_U increases, the MAE values decrease at first. But when TOP_K_U passes over a threshold, the MAE values soar again. The similar phenomena happen with respect to service side. This observation can be explained when TOP_K 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.

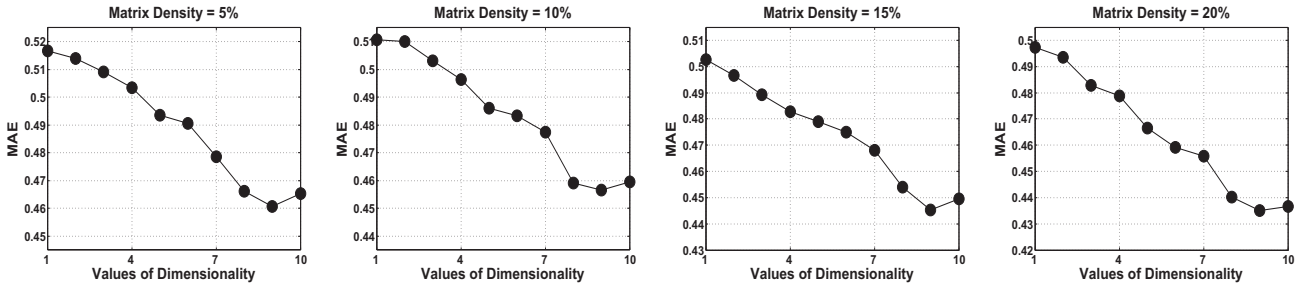

Figure 4. Impact of Dimensionality

When TOP_K is larger than a certain 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, TOP_K_U around 80 contributes to the smallest MAE values, which means TOP_K_U meets a threshold in this dataset. At the same time, the smallest MAE values in all matrix density settings happen when TOP_K_S is around 300. The optimal thresholds of TOP_K_U and TOP_K_S are different since the population on both sides is varied. This observation shows that choosing an appropriate size of neighborhood can achieve a better prediction result.

E. Impact of Dimensionality

In our proposed method, dimensionality directly determines how many factors involves to matrix factorization. To study the impact of dimensionality, we set TOP_K_U=10, TOP_K_S=100 and tune the matrix density.

Figure 4 shows that with the increase of dimensionality, the values of MAE dramatically decrease at first. However, the values of MAE increase when dimensionality goes above a certain threshold (around 90 for MAE). 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.

F. Impact of Matrix Density

To study the impact of the matrix density on MAE, we set TOP_K_S=100 and dimensionality=10. and we vary the TOP_K_U as 5, 10 and 15.

Figure 5 shows that when matrix density increases from 5% to 15%, the MAE values consistenly decrease, which means prediction accuracy is improved significantly. With the further increase in matrix density, MAE values slowly decrease. It shows that with more entries contributing to the training phase, EMF performs much better.

Another observation is that if we set Top_K_U=5, the MAE values decrease sharply when the density is low.

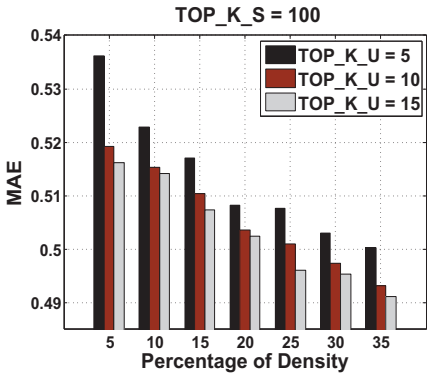

Figure 5. Impact of Matrix Density

However, the MAE values decrease slowly when the density surpasses 20. This can be explained that in this configuration when the training sample is small, our framework is sensitive to the the global information. Nevertheless when the global information is quite abundant, the main power of improving prediction accuracy is the inner structure of our framework.

VI. RELATED WORK

The problem of QoS-based service selection has been widely studied in a number of literatures during recent years. In [10], Ran et al. suggest the use of "QoS certifier". This term is used to certify the QoS claims made by service providers on their related services. The QoS values are involved into the "UDDI" registry to utilize more appropriate service selection. Zeng et al. [11] first transfer QoS-based service selection into an optimization problem. And then they present a middleware platform to select web services for the specific objectives. Alrifai et al. [2] first use Mixed Integer Programming (MIP) to transfer the global QoS constraints into local ones. And then they use distributed local selection to find the best web services that satisfy these local constraints. Kang et al. [12] develop a mechanism to optimize the service selection process for multiple related service requesters. They also propose a extended Skyline method to speed up the selection process. .

A common hypothesis of previous research is that the

QoS information is known and accurate. However, as we discussed above, there are many missing QoS values of services to the consumer in the real situations. Therefore, a fundamental process before QoS-based service selection is to predict the missing QoS values.

There are a few works aiming at making missing QoS values prediction. In [6], Shao et al. propose a user-based Collaborative Filtering algorithm to make similarity mining and predict the QoS of web services from consumers' experiences. Liu et al. [13] extends the personalized QoS prediction approach to select the best-fit service. Zheng et al. [8] present a hybrid approach which combines user-based and item-based approach together to predict the QoS of web services. More specifically, they employ two confidence weights to balance these two predicted values. Chen et al. [14] utilize several techniques to fill in the missing QoS values for service selection process. Chen et al. [5] discover the great influence of user's location to the accuracy of prediction and propose a region-based hybrid Collaborative Filtering algorithm to predict the QoS of services.

Different from previous research, we propose a unified framework with relational regularization terms. This extended Matrix Factorization framework can tackle the cold-start problem happens in previous works. Moreover, the complexity analysis shows that our model is quite effective and scale to the large dataset. The experiments show that our approach outperforms the existing approaches in prediction accuracy.

VII. CONCLUSIONS

In this paper, we proposed an extended Matrix Factorization framework with relational regularization to make QoS values prediction. Our EMF approaches adopted different similarity measurement techniques to identify neighborhood on user and service side. And we also introduced two novel relational regularization terms to revamp classic MF model into a unified framework. The empirical analysis showed that our approach outperforms other state-of-the-art methods in QoS prediction accuracy.

In our future work, more relational regularization terms will be applied to solve the problem of QoS prediction. The other parts in our EMF service selection framework will be explored extensively.

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. Papazoglou and D. Georgakopoulos, "Service-oriented computing," *Communications of the ACM*, 2003.

[2] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," *Proceedings of the 18th international journal on World Wide Web (WWW)*, 2009.

[3] T. Yu, Y. Zhang, and K. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, 2007.

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

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

[7] Y. Koren and R. Bell, "Advances in collaborative filtering," *Recommender Systems Handbook*, 2011.

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

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

[10] S. Ran, "A model for web services discovery with qos," *ACM Sigecom exchanges*, 2003.

[11] L. Zeng, B. Benatallah, A.H.H.Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Trans. on Software Engineering*, 2004.

[12] G. Kang, J. Liu, M. Tang, X. Liu, and K. Fletcher, "Web service selection for resolving conflicting service requests," 2011.

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

[14] L. Chen, Y. Feng, J. Wu, and Z. Zheng, "An enhanced qos prediction approach for service selection," 2011.

[15] Q. Sun, S. Wang, H. Zou, and F. Yang, "Qssa: A qos-aware service selection approach," *International Journal of Web and Grid Services*, 2011.