Alleviating Data Sparsity in Web Service QoS Prediction by Capturing Region Context Influence

Zhen Chen[1(✉)], Limin Shen[1(✉)], Dianlong You[1], Feng Li[2], and Chuan Ma[1]

[1] School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
ysucz0815@163.com, {shenllmm,youdianlong}@sina.com
[2] School of Computer Science and Engineering, Northeastern University, Shenyang 110000, China
{ysu_lifeng,tianyi_mc}@126.com

Abstract. With the advent of service computing paradigm, Web service QoS prediction has become a necessity to support high quality service recommendation and reliable Web-based system building. However, the inherent data sparsity issue and potentially strong but inconspicuous relation between users or Web services and their neighborhoods under the context of region information are overlooked in previous studies. In this paper, we propose a unified matrix factorization model by capturing the influences of region contexts from both user and service sides in an integrated way. Different from previous researches, our approach capitalizes on the advantages of latent feature and neighborhood approaches systematically so as to achieve accurate QoS prediction. Experimental results have shown the proposed approach outperforms its competitive methods with respect to accuracy efficiently, thereby demonstrating the positive effect that incorporation of explicit region context can have on alleviating the concerned data sparsity issue.

Keywords: Web service · QoS prediction · Data sparsity · Region context

1 Introduction

Web services are software systems designed to support interoperable machine-to-machine interaction over a network [1]. Due to the advantages of dynamic binding, loosely coupling and across platform, it facilitates the delivery of business applications as Web services are accessible to anyone, anytime, at any location and using any platform. Additionally, Web services give benefits to both users and providers in such a way that users get what they expect for their paid electronic solutions [2], while providers can concentrate on the core competencies of their business without devoting too many precious people to develop the specific functions [3].

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017
S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 540–556, 2017.
DOI: 10.1007/978-3-319-59288-6_53

With the increasing amount and wide application of Web services, it becomes very challenging for users to find appropriate Web services with equivalent functionality [4]. For example, the number of available identification card inquiring services in Baidu API Store[1] is up to 30, as of May 2016. The wealth of Web services brings about a problem rather than a solution because users are drowning into the sea of service selection. Quality of service (QoS), characterizing the no-functional attributes of Web services, such as response time and throughput, is then introduced to solve this challenge by differentiating the performance differences of Web services [5]. However, it is often required to predict QoS values of Web services due the following facts: (1) the number of available Web services is large and is still growing, and a user has accessed a few Web services and the vast majority of QoS values are unknown, a problem known as *data sparsity*; (2) QoS data is related to the specific situation, users in different region areas might have dramatically different QoS experiences; (3) Testing all candidate Web services is not practical in application, because it is very time and resource consuming and some Web services need to be paid for a function call.

To tackle the problem of Web service QoS prediction, researchers have devised a number of collaborative filtering (CF) based methods, which could be divided into two main categories: neighborhood-based and model-based methods. Neighborhood-based methods exploit similar neighbors' ratings directly to make prediction. Shao et al. first introduce CF to make Web service QoS prediction [6]. They make similarity mining and prediction from users' neighborhood. Zheng et al. propose a hybrid QoS prediction method by combining user-based and service-based CF methods [7]. Wu et al. adopts a similar fusion method by designing an adjusted-cosine-based similarity to remove the impact of QoS scale [8]. Although neighborhood-based methods are intuitive and easy to be implemented, they suffer from the unavoidable data sparsity issue in CF method, and this will cause a major performance degradation when there are new users and services added to the system, a problem known as *cold-start*. Moreover, users can be neighbors only when they have co-invoked some services, while this is not always true because users in the same local region may also have positive correlations. These impede the generalization of neighborhood-based methods in real scenario.

Different from neighborhood-based method making prediction directly from similar neighbors, model-based methods exploit the known QoS ratings to infer users' and services' latent features that characterize the behavior of user invoking service. Matrix factorization (MF) is one of the most successful implementation techniques of latent feature model. In [9], the authors improve prediction accuracy by extending MF with user similar neighbors. Yin et al. believe users inside a local neighborhood share similar invocation experiences, and they extend MF with a location-based regulation term and make prediction with a combination of pre-processing results of classic MF and extended MF [10]. These methods improve prediction performance by considering user side neighborhood only, while ignoring the role of neighborhood from service side. Moreover, the inherent *data sparsity* and *cold start* issues remain fully unaddressed.

[1] http://apistore.baidu.com/.

Recently, researchers have introduced location information to improve QoS prediction accuracy further. Tang et al. exploit location information for neighbor selection in their neighborhood-based CF prediction method [11]. E et al. propose a similar method with [11] and improve neighbor selection method by calculating the geographical distance with user latitude and longitude information [12]. Besides location factor, Yu et al. in [13] introduce time factor to improve Web service QoS prediction. Region context is inherently existed in user-service rating system. In this paper, the main motivation for exploiting region context in Web service QoS prediction stems from the observations that the ideas we are exposed to and the choices we make are significantly influenced by our region neighborhoods. Different from previous researches, we study the benefit of region context and propose a **RE**gion context-aware **M**atrix **F**actorization (REMF) approach, which not only integrates the cooperation ideas behind collaborative filtering but also exploits region contexts of users and services to capture the local preferences and as well as alleviates the data sparsity problem to a large extent.

Our main contributions are summarized as follows.

(1) The insights that users and services are positively correlated under the context of region information are observed based on real-world dataset analysis.
(2) REMF was proposed specifically to model user and service region context influences, producing accurate predictions and alleviating data sparsity issue.
(3) Comprehensive evaluations using real-world dataset demonstrate the effectiveness and efficiency of REMF.

The remainder of this paper is organized as follows: Sect. 2 defines our problem formally and analyzes the underlying correlations between users or services and their neighbors under the context of region information. Section 3 introduces the details of our proposed approach. Section 4 presents experimental results and Sect. 5 concludes the work.

2 Problem Statement and Observations

2.1 Problem Statement

We first introduce the notations used in this paper. Let $U = \{u_1, u_2, ..., u_m\}$ and $S = \{s_1, s_2, ..., s_n\}$ denote the sets of users and Web services, respectively, where m is the number of users and n is the number of Web services. First, users invoked Web services on the Internet through standard protocols, such as XML, SOAP and WSDL. Second, if user u used service s and $r_{u,s}$ is the observed QoS rating, otherwise we use *null* denotes the unknown rating from u to s. Third, u submits the observed rating $r_{u,s}$ to the universal description, discovery and integration system, thus m users will contribute an $m \times n$ user-service QoS rating matrix $R = \{r_{u,s}\}_{m \times n}$. As we discussed before, there is a large number of Web services on the Internet, and most users would have accessed only a small fraction of the large universe of available Web services. Consequently, the matrix R is very

sparse, and the limited available QoS data will degrade the prediction accuracy and hinder the application of Web service recommendation.

Different from QoS ratings which are explicitly contributed by users, region information is implicitly existed in every rating-based system. Let $URC = \{LonLat_u, AS_u, Country_u\}$ and $SRC = \{Provider_s, AS_s, Country_s\}$ denote the region contexts of user and service, respectively, where $LonLat_u, AS_u, Country_u$ be a set of users in the same longitude and latitude($LonLat$), autonomous system[2] (AS) and country, respectively; $Provider_s, AS_s, Country_s$ be a set of services belonging to the same provider, autonomous system and country, respectively. Figure 1 shows the elements relation of region contexts URC and SRC: $LonLat_u \subseteq AS_u \subseteq Country_u$, $Provider_s \subseteq AS_s \subseteq Country_s$.

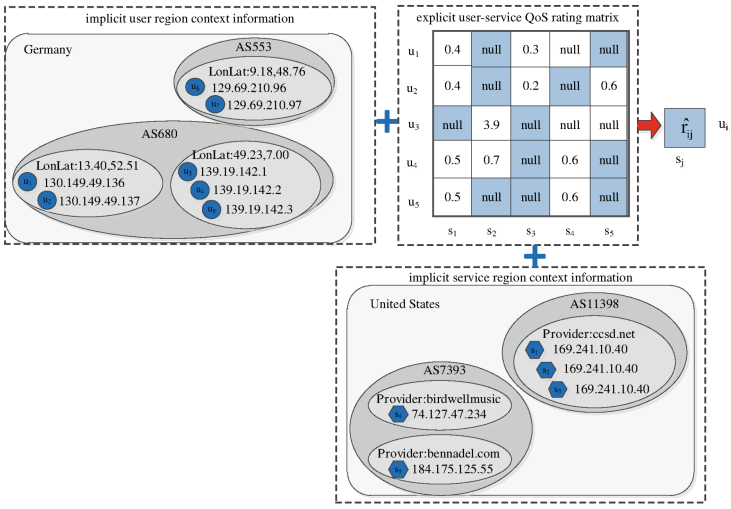


Fig. 1. Leveraging explicit QoS ratings and implicit region context for Web service QoS prediction

It is nature to assume that users with smaller region, such as in Fig. 1 u_1 and u_2 are in the same $LonLat$, will have a more similar QoS experiences on the Web services they co-invoked. This is true because users in the same $LonLat$ share the same network facilities. Moreover, services with smaller region, such as s_1 and s_2 belong to the same service provider, will give a more similar QoS performances to users as they share the same server load. Therefore, we argue that region context contains complementary information and motivates us to capture the significant influence of region context to support more accurate QoS prediction.

With the notations and motivations above, our problem can be stated as: given the known QoS ratings in R and the implicit region contexts including user region context URC and service region context SRC. We aim to predict

[2] An autonomous system (AS) is a collection of connected Internet protocol routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain that presents a common.

the unknown QoS values by using the explicit known QoS ratings in R and implicit user and service region contexts $\{URC, SRC\}$.

2.2 Observations

We collect publicly available real-world dataset WSDream for this study [14]. It contains 1,974,675 response time records collected by 339 users from 31 countries on 5,825 web services from 74 countries. Country information are collected in the dataset, while user *LonLat* and *AS* information, service *Provider* and *AS* information are not available, so we adopted an IP2Location[3] service to identify these missing region information for the purpose of prediction. Some statistics of WSDream dataset are shown in Table 1, where *AS* information of 938 Web services are not identified due to the failure of 817 $WSDL$s to IP address conversion and 121 IPs are not recorded in the known ASs.

Table 1. Statistics of WSDream dataset

User	Value	Web service	Value
Num. of users	339	Num. of web services	5,825
Num. of user *LonLat*s	159	Num. of web service providers	2,699
Num. of user *AS*s	137	Num. of web service *AS*s	1,032
Num. of user Countries	31	Num. of web service countries	74

The motivation in this study is that users'/services' QoS ratings are similar to or influenced by users/services that they are regionally related to. In this subsection, we investigate the influences of region contexts via studying the correlation between users/services and their corresponding region neighborhood. Specifically, with user and service region contexts $\{URC, SRC\}$, we ask the two questions: (1) Are users/services have the abilities to cluster a specified number of region neighbors? (2) Are users/services within smaller region more similar in terms of their region neighbors' ratings?

To answer the first question, we first give the definition of region neighbor. Let LN_u, AN_u, CN_u and PN_s, AN_s, CN_s be the set of neighbors of user u and service s at their corresponding region contexts URC and SRC, where $LN_u = \{\forall v | v \in LonLat_u\}$, $ANu = \{\forall v | v \in AS_u \wedge v \notin LonLat_u\}$, $CN_u = \{\forall v | v \in Country_u \wedge u \notin AS_u\}$, and $PN_s = \{\forall t | t \in Provider_s\}$, $AN_s = \{\forall t | t \in AN_s \wedge t \notin Provider_s\}$, $CN_s = \{\forall t | t \in Country_s \wedge t \notin AN_s\}$. Based on the definitions, the region neighbors of user u_1 in Fig. 1 can be clustered as: $LN_{u_1} = \{u_2\}, AN_{u_1} = \{u_3, u_4, u_5\}, CN_{u_1} = \{u_6, u_7\}$.

We conduct statistical distribution of users and services with different number of region neighbors, and Fig. 2 plots the statistical results. It can be found that the larger region scope, the more possibility to cluster the specified number of region neighbors, such as in *Country* region, the proportion of users and services

[3] https://www.ip2location.com.

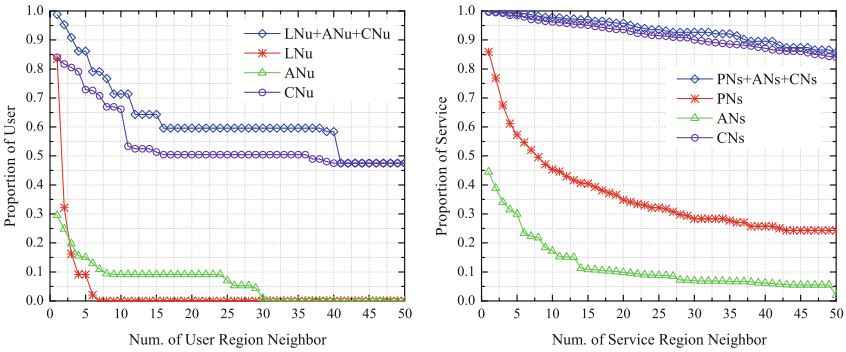


Fig. 2. Distribution of user and service with different number of region neighbors

having 30 neighbors is about 50% and 90% respectively, which is higher than the other region contexts. Moreover, by combining different region contexts, most users and services have the ability to obtain their region neighbors. Thus, the results from Fig. 2 suggest a positive answer with Observation 1 to the first question.

Observation 1. *Users and services have the abilities to obtain the specified number of region neighbors, and the larger region scope, the more likelihood of obtaining the specified number of region neighbors.*

For the second question, we want to investigate whether users and services are positively correlated with their region neighbors. In this work, we adopt the widely used Pearson Correlation Coefficient (PCC) similarity of users' and services' rating vector to measure their rating similarity [15]. With PCC, we calculate the similarity for each user/service and his region neighbors, and use the average PCC to evaluate correlation between them. For example, with the similarity PCC(u,v) of user u and v, $v \in LN_u$, the average PCC of region neighbors is calculated as: $Avg.PCC(LN_u) = \frac{1}{|LN_u|} \sum_{v \in LN_u} PCC(u, v)$, which indicates the overall correlation of user u and his $LonLat$ region neighbors LN_u.

Since the distribution of users and services is decrease with the increasing of region neighbors. We vary the number of neighbor from 0 to 50 under different region contexts, and Fig. 3(a) and (b) plot the average PCC of users and services with different number of region neighbor, respectively. We observe that the average PCC of users/services and their region neighbors is 0.52 and 0.3, respectively, indicating that the QoS ratings of users and services are positively correlated with their region neighbors. Moreover, the smaller region scope, the higher average PCC. This suggests us that when modeling region context, we should give priority to more local region neighbors. With the above analysis, we have Observation 2.

Observation 2. *With smaller region scope, users' and services' rating are more correlated with their region neighbors, and vice versa.*

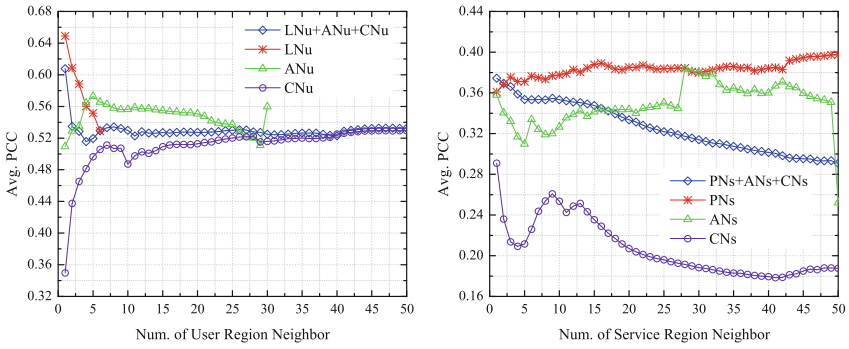


Fig. 3. Average of users and services PCC with different number of region neighbors

Positive observations to both questions provide the evidence of the significance of region contexts. With the verification of underlying relationship of region contexts and QoS ratings, we are ready study how to exploit the region context to alleviate the concerned data sparsity challenge in QoS prediction.

3 Our Method

3.1 Baseline Predictor

Our previous research demonstrates that Web service QoS data exhibit large user and Web service effects [16]. Specifically, some users tend to perceive higher response time than others due to the poor network while some Web services tend to perform lower response time than others due to the less server load. Thus, in order to adjust the QoS data by accounting for these effects, we suggest a baseline predictor. It estimates an unknown rating $\hat{r}_{0u,s}$ and accounts for user and service effects as follow.

$$\hat{r}_{0u,s} = b_a + b_u + b_s \tag{1}$$

where b_a is the global average of all ratings in matrix R, b_u and b_s represent the observed biases of u and s from average, respectively. To estimate b_u and b_s, we can solve the following optimization problem.

$$\ell = min \frac{1}{2} \sum_{u=1}^{m} \sum_{s=1}^{n} I_{u,s} (r_{u,s} - \hat{r}_{0u,s})^2 + \frac{\lambda_1}{2} \|b_u\|_F^2 + \frac{\lambda_2}{2} \|b_s\|_F^2 \tag{2}$$

where $I_{u,s}$ plays as an indicator which is equal to 1 when u has interacted with s, and is equal to 0 otherwise. $\|\|_F^2$ denotes the Frobenius norm. The first term in Eq. (2) tries to adjust bu and bs close to the real rating, and the other two regularization terms are added to avoid the over-fitting problem. Considering these effects is effective to improve prediction accuracy in our later experiments.

3.2 Matrix Factorization

Before modeling region context, we adopt a state-of-the-art prediction method based Matrix Factorization (MF) as our basic model. The idea behind MF is that user QoS rating behaviors are influenced by a reduced latent features and it performs a low-rank MF on matrix R. Let $p_u \in \mathbb{R}^{d\times m}$ and $q_s \in \mathbb{R}^{d\times n}$ denotes user and service feature vector respectively, for the missing rating $\hat{r}_{1u,s} = p_u^T q_s$, low-rank MF strives to make $\hat{r}_{1u,s}$ as close as possible to the ground truth $r_{u,s}$ by minimizing the following objective function.

$$\ell = min \frac{1}{2} \sum_{u=1}^{m} \sum_{s=1}^{n} I_{u,s}(r_{u,s} - \hat{r}_{1u,s})^2 + \frac{\lambda_3}{2} \|p_u\|_F^2 + \frac{\lambda_4}{2} \|q_s\|_F^2 \tag{3}$$

There are several advantages of MF method [17]. (1) data sparsity issue can be alleviated by factorizing the user-service rating matrix, it allows to obtain meaningful relations between pairs of users or services, even though these users have invoked different services, or these services were accessed by different users; (2) MF can be solved by a simple optimization problem and a local optimal solution can be found by an gradient based method; (3) MF is extensible and enables us to integrate other meaningful sources of side information, such as user and service region contexts in the following subsection.

3.3 Modeling User Region Context Influence

The region context of user perspective reveals the rating correlation of users and their region neighborhood. The observations in Subsect. 2.2 suggest that users are able to cluster a specified number of positively correlated region neighbors. With user region context, users with closer region scope are more likely to have similar QoS experiences. Specifically, the QoS experience of a user u is influenced by the region neighbors that are in the same region area. Thus, the feature vector of u can be defined as a combination of the feature vectors p_v to the region neighbors RN_u of u.

$$\hat{p}_u = |RN_u|^{-\frac{1}{2}} \sum_{v \in RN_u} p_v \quad \hat{r}_{2u,s} = \hat{p}_u^T q_s = (|RN_u|^{-\frac{1}{2}} \sum_{v \in RN_u} p_v)^T q_s \tag{4}$$

The combination for $\hat{p}_u$ is based on the fact that similar users will capture similar latent features, consequently transferring the knowledge from user region neighbors. Equation (4) indicates how the user region context is modeled in MF, and region neighbor RN determines whom are used to build the model. Observation 2 shows that users in smaller region have more positive correlation with their neighbors, and the neighbor selection order is suggested as follows: $LonLat$, AS and $Country$. Thus, region neighbors RN_u of u can be clustered as:

$$RN(u) = \begin{cases} \{v | v \in K_l(LN_u)\} & if\ |LN_u| \geq UK \\ \{v | v \in LN_u + K_a(AN_u)\} & if\ |LN_u| \leq UK \wedge |PN_u| + |AN_u| \geq UK \\ \{v | v \in LN_u + AN_u + K_c(CN_u)\} & otherwise \end{cases} \tag{5}$$

where UK denotes the number of user u's region neighbors.

3.4 Modeling Service Region Context Influence

The service side region context reveals the rating correlation of services and their region neighborhood. Previous observations also show that services' region context have a positive relationship with QoS ratings, so we adopt a similar method like the modeling of user context to capture the influence of service region context. Similarly, the feature vector of s can be defined as a combination of feature vectors q_t to the region neighbors RN_s of s.

$$\hat{q}_s = |RN_s|^{-\frac{1}{2}} \sum_{t \in RN_s} q_t \quad \hat{r}_{3u,s} = p_u^T \hat{q}_s = p_u^T (|RN_s|^{-\frac{1}{2}} \sum_{t \in RN_s} q_t) \tag{6}$$

The clustering method of region neighbors RN_s is suggested as follows:

$$RN(s) = \begin{cases} \{t | t \in K_p(PN_s)\} & if\ |PN_s| \geq SK \\ \{t | t \in PN_s + K_a(AN_s)\} & if\ |PN_s| \leq UK \wedge |PN_s| + |AN_s| \geq SK \\ \{t | t \in PN_s + AN_s + K_c(CN_s)\} & otherwise \end{cases} \tag{7}$$

where SK denotes the number of service s's region neighbors.

3.5 Ensemble Method

In the above subsections, we introduce our solutions to consider the bias effects and to model region contexts mathematically. With these solutions, we proposed an ensemble method REMF by leveraging bias information and region context systematically. The basic idea of REMF is that users/services with a same region context should have similar QoS experience/performance, and an unknown rating $\hat{r}_{u,s}$ can be predicted as a linear combination of ratings from the baseline predictor, MF and region neighbors aware methods.

$$\hat{r}_{u,s} = \alpha \hat{r}_{0u,s} + \beta \hat{r}_{1u,s} + \gamma \hat{r}_{2u,s} + \theta \hat{r}_{3u,s} \quad \alpha + \beta + \gamma + \theta = 1 \tag{8}$$

In Eq. (8), the first term is the bias effect, the second term is basic MF, and the last two terms are the influences of user and service region contexts. Here, we suggest a unified ensemble model REMF that improves prediction accuracy by capitalizing on the advantages of bias, neighborhood and latent feature methods. REMF is a post-processing method that user and service bias effects, region information and latent feature information are built systematically, rather than a simple combination of pre-processing factorization results. To factorize the assemble model, we aim to solve the following problem.

$$\begin{aligned} \ell = min \frac{1}{2} \sum_{u=1}^{m} \sum_{s=1}^{n} I_{u,s} (r_{u,s} - \hat{r}_{u,s})^2 + \frac{\lambda}{2} (|S_u|^{-\frac{1}{2}} \|b_u\|_F^2 + |U_s|^{-\frac{1}{2}} \|b_s\|_F^2 + \\ |S_u|^{-\frac{1}{2}} \|p_u\|_F^2 + |U_s|^{-\frac{1}{2}} \|q_s\|_F^2 + \sum_{v \in RN_u} |S_v|^{-\frac{1}{2}} \|p_v\|_F^2 + \sum_{t \in RN_s} |U_t|^{-\frac{1}{2}} \|q_t\|_F^2) \end{aligned} \tag{9}$$

where S_u be a set of services invoked by user u, and U_s be a set of users who access service s. RN_u and RN_s be a set of region neighbors clustered by Eqs. (5) and (7), respectively.

The gradients of ℓ with respect to variables in set $Para = \{b_u, b_s, p_u, q_s, RN_u, RN_s\}$ are:

$$\nabla b_u = -\alpha e_{u,s} + \lambda \left|S_u\right|^{-\frac{1}{2}} b_u \tag{10}$$

$$\nabla b_s = -\alpha e_{u,s} + \lambda \left|U_s\right|^{-\frac{1}{2}} b_s \tag{11}$$

$$\nabla p_u = -e_{u,s}(\beta q_s + \theta(\left|RN_s\right|^{-\frac{1}{2}} \sum_{t \in RN_s} q_t)) + \lambda \left|S_u\right|^{-\frac{1}{2}} p_u \tag{12}$$

$$\nabla q_s = -e_{u,s}(\beta p_u + \gamma(\left|RN_u\right|^{-\frac{1}{2}} \sum_{v \in RN_u} p_v)) + \lambda \left|U_s\right|^{-\frac{1}{2}} q_s \tag{13}$$

$$\forall v \in RN_u : \nabla p_v = -e_{u,s}\gamma \left|RN_u\right|^{-\frac{1}{2}} q_s + \lambda \left|S_v\right|^{-\frac{1}{2}} p_v \tag{14}$$

$$\forall t \in RN_s : \nabla q_t = -e_{u,s}\theta \left|RN_s\right|^{-\frac{1}{2}} p_u + \lambda \left|U_t\right|^{-\frac{1}{2}} q_t \tag{15}$$

where $e_{u,s} = r_{u,s} - \hat{r}_{u,s}$. An optimal solution of the objective function ℓ in Eq. (9) can be obtained by an stochastic gradient descent method, which iterates every non-*null* ratings in matrix R until convergence. The detailed algorithm is shown in Algorithm 1.

Algorithm 1. The proposed region context aware matrix factorization REMF

Input: rating matrix R, the region neighbor RN; parameters UK, SK, lr, d;

Output: user bias feature vector B_u, service bias feature vector B_s, user feature matrix U, service feature matrix S, user neighbor feature matrix UN, service neighbor feature matrix SN;

```
Initialize b_u, b_s, U, S, UN, SN randomly;
while not convergent do
    Calculate ∇b_u, ∇b_s, ∇p_u, ∇q_s;
    Update b_u = b_u − lr∇b_u;
    Update b_s = b_s − lr∇b_s;
    Update p_u = p_u − lr∇p_u;
    Update q_s = q_s − lr∇q_s;
    for each non-empty v ∈ RN_u do
        calculate ∇p_v;
        Update p_v = p_v − lr∇p_v;
    end for
    for each non-empty t ∈ RN_s do
        calculate ∇q_t;
        Update q_t = q_t − lr∇q_t;
    end for
end while
```

Algorithm 1 first initializes model parameters randomly. Then an iteration procedure is loaded for learning. Parameter lr is the learning rate that controls the speed of iteration. After learning all parameters, the unknown QoS ratings can be predicted with Eq. (8).

4 Experiments

In this section, we conduct experiments to answer the followings questions: (1) how does the proposed method REMF perform compared to the well-known predictors in both warm-start situation and cold-start situations? (2) how do the user region and service region contexts affect the prediction accuracy of REMF? (3) what is the impact of region neighborhood size? (4) how is the efficiency of REMF?

4.1 Experimental Settings

Publicly dataset WSDream is adopted and we choose response time QoS attribute to evaluate our proposed method, the region information are extracted by an IP2Location service. Some statistics of the dataset are presented in Table 1. In practical, the user-service QoS matrix R is very sparse, thus we choose $x\%$ ratings as the origin data to training the data and the remaining $1 - x\%$ as the testing data to predict. In the following experiment, x is conducted as 2.5, 5, 7.5, 10.

Two well-known metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), are adopted to evaluate the prediction performance. The metric MAE is defined as:

$$MAE = \frac{1}{N} \sum |r_{u,s} - \hat{r}_{u,s}| \tag{16}$$

where N is the number of ratings in the testing set. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N} \sum |r_{u,s} - \hat{r}_{u,s}|^2} \tag{17}$$

A smaller RMSE or MAE value means better performance. Note that previous work demonstrated that small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-k recommendation [18].

We estimate the latent factors of REMF with Algorithm 1 and use a learned model for QoS prediction. We use training data to estimate REMF's parameters and select the one provide the optimal results. Table 2 presents the parameters used to make prediction.

4.2 Comparison of Different Predictor

In this section, we compare our REMF with various representative prediction methods as follows: GMEAN, UMEAN, IMEAN, UPCC, IPCC, WSRec [7], MF [15], NIMF [9] and Colbar [10]. GMEAN employs the global average to make

Table 2. Experimental parameter settings

λ	lt	α	β	γ	θ	UK	SK	Dimensionality d	Num. of iteration
0.02	0.012	0.1	0.3	0.3	0.3	10	20	11	17

prediction, which is equals to b_a in baseline predictor. UMEAN and IMEAN employ the mean QoS value of user and service as results respectively. UPCC, IPCC, WSRec are the neighborhood based CF, UPCC and IPCC use the similar neighbors of users and services to make collaborative prediction, respectively, and WSRec is the combination of them. MF conducts a basic matrix factorization on the QoS matrix, and NIMF extend MF by incorporating user neighborhood information, both of them can be realized through REMF by setting $\alpha = 0, \beta = 1, \gamma = 0, \theta = 0$ and $\alpha = 0, \beta = 0.5, \gamma = 0.5, \theta = 0$, respectively. Colbar integrates user location-aware neighbors in MF and make prediction by fusing the results of basic MF and Extended MF.

Tables 3 and 4 illustrates the compared results under $warm - start$ scenario (each user invokes at least one service, and each service has been accessed at least once) and $cold - start$ scenario (there exist 34 $cold - start$ users who do not invoke any services and 583 $cold - start$ services that do not accessed by

Table 3. Comparisons in warm-start scenario

Metric	MAE				RMSE			
Density	2.5%	5%	7.5%	10%	2.5%	5%	7.5%	10%
GMEAN	1.1099	1.0466	1.0258	1.0191	1.9758	1.9691	1.9676	1.9652
UMEAN	0.8686	0.8680	0.8703	0.8725	1.8565	1.8552	1.8527	1.8505
IMEAN	0.8666	0.7649	0.7245	0.7152	1.6680	1.5731	1.5487	1.5358
UPCC	0.7148	0.6948	0.6836	0.6712	1.5668	1.4966	1.4697	1.4209
IPCC	0.7180	0.7142	0.7074	0.6963	1.6059	1.5298	1.5032	1.4520
WSRec	0.7040	0.5999	0.5478	0.5265	1.5140	1.3618	1.3202	1.2925
MF	0.7389	0.6470	0.6358	0.5777	1.9502	1.6088	1.4379	1.3534
NIMF	0.6996	0.5584	0.5143	0.4953	1.5796	1.3523	1.2858	1.2415
Colbar	0.6720	0.5449	0.5136	0.4914	1.4790	1.3232	1.2691	1.2338
REMF	**0.6328**	**0.5320**	**0.5016**	**0.4805**	**1.4277**	**1.2812**	**1.2351**	**1.1995**

Table 4. Comparisons in cold-start scenario

Metric	MAE				RMSE			
Density	2.5%	5%	7.5%	10%	2.5%	5%	7.5%	10%
GMEAN	1.1071	1.0432	1.0424	1.0130	1.9752	1.9696	1.9648	1.9690
MF	0.7528	0.6824	0.6424	0.5977	1.7910	1.6770	1.5025	1.4912
REMF	**0.6645**	**0.5772**	**0.5374**	**0.5219**	**1.5851**	**1.4188**	**1.3650**	**1.3146**

any users), respectively. Observing from the above results we have the following observations: (1) REMF consistently obtains the lowest MAE and RMSE under all settings, especially compared with NIMF and Colbar, REMF obtains better results than them, this is because of the incorporating of service neighbors under region context. These demonstrate that exploiting region contexts can significantly improve prediction performance. (2) The accuracy of REMF is increased along with the increasing of matrix density. The major reason is that much more QoS rating data can afford more information on learning a more accurate model, which suggests us encouraging users to share their observed QoS rating to UDDI for better prediction. (3) In the pure $cold-start$ scenario, only GMEAN and MF methods can work, and the proposed REMF outperforms the compared methods. The average performance of REMF is reduced by 7.31% and 10.43% in terms of MAE and RMSE which is acceptable in practice.

With the above observations, we can answer the first question: by capturing the influence of region contexts, the proposed method is superior to other methods and alleviates the data-sparsity issue with better accuracy.

4.3 Impact of Assemble Weights

Assemble weights α, β, γ and θ are used to balance the effects of bias information, latent factor, user region context and service region context. To investigate the impact of these assemble weights, we train them with 6 combinations under different matrix densities and the results are shown in Table 5.

Table 5. Impact of assemble weights

Metric	MAE				RMSE			
Density	2.5%	5%	7.5%	10%	2.5%	5%	7.5%	10%
$\alpha = 1, \beta = 0, \gamma = 0, \theta = 0$	0.6693	0.6107	0.5519	0.5162	1.4654	1.3553	1.2904	1.2481
$\alpha = 0, \beta = 1, \gamma = 0, \theta = 0$	0.6934	0.6690	0.5904	0.5164	1.8603	1.5337	1.3812	1.2837
$\alpha = 0, \beta = 0, \gamma = 1, \theta = 0$	0.6904	0.6144	0.5790	0.5435	1.5930	1.4505	1.3807	1.3101
$\alpha = 0, \beta = 0, \gamma = 0, \theta = 1$	0.7668	0.6432	0.6059	0.5559	1.6747	1.3997	1.3581	1.2692
$\alpha = 0.25, \beta = 0.25, \gamma = 0.25, \theta = 0.25$	0.6192	0.5317	0.5006	0.4818	1.4241	1.2875	1.2383	1.1992
$\alpha = 0.1, \beta = 0.3, \gamma = 0.3, \theta = 0.3$	**0.6180**	**0.5309**	**0.4968**	**0.4767**	**1.4127**	**1.2775**	**1.2361**	**1.1960**

It is observed in Table 5 that the last 2 combinations have a relatively better performance compared with the first 4 combinations, which illustrates that users' and services' bias information and region context will all contribute positive influences to improve prediction performance. This shows the effectiveness of our assemble method and also explains why we choose $\alpha = 0.1, \beta = 0.3, \gamma = 0.3, \theta = 0.3$ in our experiment settings.

4.4 Impact of Neighborhood Size UK and SK

Parameters UK and SK determine the number of neighbors that will be used to transfer the knowledge from region contexts. When UK and SK are at near-zero values, REMF is thus degenerated to a basic MF model with biases. On the other hand, when UK and SK are set to be very large values, this indicates region contexts are fully considered in REMF. Table 6 shows the experimental results of REMF with different number of UK and SK.

Table 6. Impact of neighborhood size

Density	SK	MAE			RMSE			Time/Iteration (unit: s)		
		UK = 0	UK = 10	UK = 20	UK = 0	UK = 10	UK = 20	UK = 0	UK = 10	UK = 20
5%	0	*0.6648*	0.6343	0.6368	*0.5016*	0.4966	0.4954	*0.6300*	1.8171	2.7681
	10	0.6151	0.6176	0.6139	0.4934	0.4922	0.4916	2.0631	3.1700	4.1732
	20	0.6189	**0.6174**	0.6143	0.4934	**0.4921**	0.492	3.5102	**4.5630**	5.6463
	30	0.6160	0.6139	0.6139	0.4888	0.4892	0.4917	4.9072	5.9600	7.0034
10%	0	*1.6464*	1.5101	1.5055	*1.2444*	1.2331	1.2353	*1.3260*	3.6262	5.4513
	10	1.4187	1.4176	1.4121	1.2054	1.2033	1.2085	4.0842	6.3753	8.1864
	20	1.4196	**1.4127**	1.4096	1.2055	**1.2027**	1.2016	6.8393	**9.1235**	10.8000
	30	1.4148	1.4086	1.4100	1.2038	1.2021	1.2003	9.5575	11.8606	13.6727

From Table 6, we observe some interesting observations: (1) When UK = SK = 0 without considering region contexts, REMF have a relatively lager MAE and RMSE, and they decrease with the increase of UK and SK. This verifies that incorporating region context in REME can indeed contribute valuable information and improve performance. (2) When UK $>$ 10 and SK $>$ 20, varying UK and SK has little improvement on accuracy under different matrix density, and the accuracy even starts to fluctuate because too many neighbors may conclude lower correlated neighbors that will introduce noise and do harm for the performance. (3) More neighbors mean more computation time, the configuration, UK = 10, SK = 20, gives us an acceptable prediction accuracy and computation time, thus we set UK = 10, SK = 20 in the other experiments.

4.5 Efficiency Analysis

The theoretical time complexity of REMF is mainly based on the learning of $Para = \{b_u, b_s, p_u, q_s, RN_u, RN_s\}$, which is $O(d*|R|*K)$, where d is the number of latent feature, K is the max value of UK and SK, $K = max(UK, SK)$. Due to d and $K \ll |R|$, REMF is scalable and linear with respect to the number of observed QoS in matrix R. To evaluate the efficiency our proposed approach, we conduct experiments to study the average time per iteration in compared methods. The experiment results are shown in Table 7.

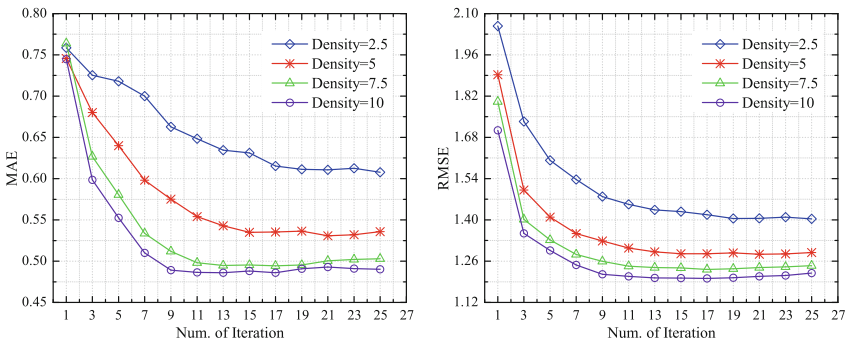
Table 7 shows that MF requires the least time in one iteration, NIMF and Colbor need about 2 s for each iteration when density is 10%, and our REMF

Table 7. Average time/iteration comparisons (unit: s)

Density	MF	NIMF	Colbor	REMF
5%	0.0630	1.0170	1.0930	4.4925
10%	0.1270	2.0351	2.1431	9.0148

obtains the largest computation time in one iteration. This because REMF integrates not only 10 user neighbors but also 20 service neighbors, while MF does not consider any neighbor information and NIMF and Colbor incorporate 10 user neighbors without considering service side information. The more neighbors mean the longer training time to learn these latent features. Note that experimental results show that there is higher predication accuracy of REMF than other methods in both warm-start and cold-start case. This shows a tradeoff between collaborative neighbors (accuracy) and efficiency.

Fortunately, Fig. 4 shows the performance of REMF with the number of iteration varying from 1 to 27. It is observed that both MAE and RMSE decrease with the increasing number of iteration, and arrives at a convergence when iteration number is about 17. The results reflect REMF have a fast convergence speed and good performance. It is worth noting in Table 7 when density = 10% the average time for one iteration is 9.0148 s, thus the overall leaning time of REMF is no more than 3 min. This indicates that REMF is applicable in practice.



(a) (b)

Fig. 4. Performance with different number of iteration

5 Conclusions and Future Work

This paper proposed a novel Web service QoS prediction approach by capturing region context influence, which makes contribution to alleviate the data sparsity issue with the following conclusions. (1) Publicly real-world Web service QoS data analysis shows that there exists a positive rating correlating between

users/services and their region neighborhoods, which suggests that region context contains complementary information and provides necessary technical support for QoS prediction. (2) Our proposed method systematically integrates region contexts from both user and service sides into a unified model, which provides an effective solution for alleviating data sparsity issue and cold-start issue. (3) Comprehensive experimental results show that region context can efficiently and significantly improve the performance of Web service QoS prediction. For future research, we intend to further improve the proposed method by considering other relevant context such as social and time information.

Acknowledgements. This work is supported by National Natural Science Foundation of China (61272466, 61300193), Hebei Provincial Natural Science Foundation (F2016203290) and Colleges and Universities in Hebei Province Science and Technology Research Project (QN2016073).

References

1. Zhang, L., Zhang, J., Cai, H.: Services Computing. Springer & Tsinghua University Press, Beijing (2007)
2. Angelov, S., Grefen, P.: The business case for B2B e-contracting. In: Proceedings of the 6th International Conference on Electronic Commerce, pp. 31–40. ACM, New York (2004)
3. Moghaddam, M., Davis, J.G.: Service selection in web service composition: a comparative review of existing approaches. In: Bouguettaya, A., Sheng, Q.Z., Daniel, F. (eds.) Handbook on Web Services: Web Services Foundations, pp. 321–346. Springer, New York (2014)
4. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: state of the art and research challenges. IEEE Comput. **40**(11), 38–45 (2007)
5. Ran, S.: A model for web services discovery with QoS. ACM SIGecomExchanges **4**(1), 1–10 (2003)
6. Shao, L., Zhang, J., Wei, Y.: Personalized QoS prediction for web service via collaborative filtering. In: Proceedings of IEEE Conference on Web Services, Salt Lake City, pp. 439–446. IEEE (2007)
7. Zheng, Z., Ma, H., Lyu, M.R.: QoS-aware web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(5), 140–152 (2011)
8. Wu, J., Chen, L., Feng, Y., Zheng, Z.: Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Trans. Syst. Man Cybern. Syst. **43**(2), 428–439 (2013)
9. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
10. Yin, J., Lo, W., Deng, S., Li, Y., Wu, Z., Xiong, N.: Colbar: a collaborative location-based regularization framework for QoS prediction. Inf. Sci. **265**, 68–84 (2014)
11. Tang, M., Jiang, Y., Liu, J., Liu, X.: Location-aware collaborative filtering for QoS-based service recommendation. In: Proceedings of the 19th International Conference on Web Services, Miami, pp. 202–209. IEEE (2012)
12. E, H., Tong, J., Song, M., Song, J.: QoS prediction algorithm used in location-aware hybrid web service. J. China Univ. Posts Telecommun. **22**(1), 42–49 (2015)

13. Yu, C., Huang, L.: A web service QoS prediction approach based on time- and location-aware collaborative filtering. SOCA **10**(2), 135–149 (2016)
14. Zheng, Z., Zhang, Y., Lyu, M.R.: Distributed QoS evaluation for real-world web services. In: Proceedings of the 8th International Conference on Web Services, Miami, pp. 83–90. IEEE (2010)
15. Benesty, J., Chen, J., Huang, Y.: Pearson correlation coefficient. In: Benesty, J., Chen, J., Huang, Y., Cohen, I. (eds.) Noise Reduction in Speech Processing, vol. 2(2), pp. 1–4. Springer, Heidelberg (2009)
16. Shen, L., Chen, Z., Li, F.: Service selection approach considering the uncertainty of QoS data. Comput. Integr. Manuf. Syst. **19**(10), 2652–2663 (2013)
17. Menon, A.K., Elkan, C.: Link prediction via matrix factorization. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011. LNCS, vol. 6912, pp. 437–452. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23783-6_28
18. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD, pp. 426–434. ACM, New York (2008)