Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080553

Topic Model based Collaborative QoS Prediction

Jian Wu[*], *Lichuan Ji*[*], *Tingting Liang*[*] *and Liang Chen*[*]

College of computer science and technology, Zhejiang University, Hangzhou, China

Received: 25 Sep. 2013, Revised: 23 Dec. 2013, Accepted: 24 Dec. 2013
Published online: 1 Sep. 2014

Abstract: With the increasing development and growth of Web services on the World Wide Web, the demand of appropriate Web service selection approaches are unprecedentedly strong, and Quality-of-Service (QoS) based service computing is becoming an important issue of service-oriented computing. In most of previous works, the QoS values of services to users are all conceived to be known, however, lots of them are unknown in practice application. Recently, lots of literatures aiming at predicting such missing QoS values are published, they all consider the unknown QoS values prediction as a fundamental step for the QoS-based service computing. Looking through existing works, we discover that the online cold-start scenario, in which some new coming Web services haven't been involved even once, hasn't been considered carefully. In this paper, we utilize a collaborative framework by integrating matrix factorization with probabilistic topic model to predict QoS values. Specifically, the basic idea of the proposed approach is collaborative filtering via matrix factorization, while the cold-start problem is handled by employing probabilistic topic model based on WSDL (Web Service Description Language) documents. The experiment are based on two real-world datasets (one contains 100 users and 150 Web services, and the other contains 339 users and 2344 Web services), and the results demonstrate the prediction accuracy of the proposed approach.

Keywords: Keywords-Web service, QoS prediction, collabrative filtering, topic model.

1 Introduction

There are lots of studies [1],[2] about QoS-based service selection these years. These methods have the common premise that all the Web services QoS values must be known and accurate. Obviously, accurate QoS values of Web services guarantee the QoS-based approaches work well. However, this hypothesis is not always true in reality. For example, there are too many Web services emerging on the Internet every day, so that users cannot invoke all the services. For some types of services, many users may have never need the type of function before. In addition, the environment on the Internet is dynamic and users' hosts have different settings, these make the evaluation of QoS values more difficult.

So it is obvious to be impractical to acquire accurate QoS values in reality. Table 1 shows this phenomenon in a simple example. The numeric values in this table represent responding time for each user to invoke the corresponding services, and the notation "–" means that the user has never used this Web service or he cannot succeed in invoking this service. For example, when $user_1$ send the request message to the service ws_1 , he will receive the respond after 1.3 seconds. But for service $ws3$, he has not historical record and this gives rise to the missing data problem: (1) the QoS historical matrix is sparse and contains many missing values; (2) some Web service such as $ws3$ is a new comer and have never been invoked by any user.

Table 1: RESPONSE TIME OF WEB SERVICE

	$user_1$	$user_2$	$user_3$
ws_1	1.3s	–	2s
ws_2	1.2s	–	–
ws_3	–	–	–
ws_1	1s	2s	2.3s

For the first problem, the common methods can address it. For example, we can draw user i and service j representative vector u_i and v_j by matrix factorization, then predict the QoS value through this formula: $\hat{r}_{i,j} = u_i^T v_j$. However, for the second problem, because the new Web service has none QoS historical record in the system, cold start phenomenon makes matrix factorization impossible. To execute QoS-based service

* Corresponding author e-mail: {wujian2000,jilichuan, cliang, liangtt}@zju.edu.cn

```
<schema targetNamespace="http://example.com/stockquote.xsd"
        xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
        <complexType>
            <all>
                <element name="StockCode" type="string"/>
            </all>
        </complexType>
    </element>
    <element name="TradePrice">
        <complexType>
            <all>
                <element name="price" type="float"/>
            </all>
        </complexType>
    </element>
</schema>
```

Fig. 1: An example of WSDL file

selection, the missing data, especially ones of new services, has to be filled before the selection process.

LDA[3] (Latent Dirichlet Allocation) is a powerful approach to extract topic distribution of documents in text corpus. WSDL (Web Service Description Language) is an XML format file for describing Web services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information [4] . Besides the operating message, we also can find description text from the interface names. Figure 1 is a simple example of WSDL file.

We can extract words reflecting the Web service function to a certain degree from element names in WSDL files. For example, from the WSDL file of Fig. 1, we can extract words “stock”, “price” and “trade”, and these words imply this Web service is about “business”. LDA is based on bag of words model and it does not care the ordinal relation of words in documents. Obviously, words extracted from WSDL file can compose a description document of this Web service. According to these description documents we can extract the topic distribution of the Web services.

Inspired by the collaborative topic regression model proposed by Wang [5], we propose a collaborative QoS prediction model called CQP. Our model uses the WSDL files and QoS value historical record together and integrates the matrix factorization with topic model. Our model can solve the second problem above effectively.

In particular, this paper has three-fold contributions:

–To improve the accuracy of QoS prediction, we propose a collaborative model named CPQ which utilizes WSDL files itemand QoS matrix conjunctively.
–Our method can make QoS prediction for new Web service not invoked by anyone effectively.
–We experimentally evaluate CPQ model by employing two real-world datasets.

The rest of this paper is organized as follows: Section II reviews the related work. Section III illustrates the background about matrix factorization and probabilistic topic model. Section IV presents our collaborative QoS prediction model. Section V describes our experiment and Section VI concludes the paper.

2 Related Work

The problem of QoS-based service selection has widely attracted the attention of many researchers in recent years. There have a number of literatures about this problem. Zeng et al. [6] have built the dynamic Web service composition with five generic QoS properties. Ran et al. [7] have proposed a service discovering model with “QoS certifier”. This model certifies the QoS claims made by service providers on their corresponding services. In [8], Zeng et al. first consider QoS-based service selection as an optimization problem and present a middleware platform to select Web services through maximizing user satisfaction measured by QoS attributes while satisfying the constraints.Yu et al. [2] have designed a broker-based architecture to facilitate the selection of QoS-based services.The target of service selection is to maximize an application-specific utility function under the end-to-end QoS constraints.In [9], Liu et al. have presented a open, fair and dynamic QoS computation model for web services selection through implementation of and experimentation with a QoS registry in a hypothetical phone service provisioning market place application.Serhani et al. [10] have put forward a two-phase verification technique that is performed by a third party broker in order to achieve the goal which is supporting the client in selecting web services based on his/her required QoS.

A common latent premise of previous research is that the QoS values of Web services to users are all known and accurate or can be easily obtained from the service providers. However, as we discuss above, there are often missing data in reality and these methods encounter restrictions in real-world practical applications. Therefore, a preprocessing before QoS-based service selection is to predict the missing QoS values.

To predict QoS values, limited approaches have been proposed these years. Shao et al. [11] have proposed a user-based collaborative filtering algorithm to predict the QoS values of Web services from consumers’ experiences. Zheng et al. [12] have proposed a hybrid approach named WSRec that combines user-based and item-based methods together to predict the QoS values of Web services. Chen et al. [13] have propose a region-based hybrid collaborative filtering algorithm to predict the QoS values taking advantage of the great influence of user’s location to the accuracy of prediction. Zibin’ NIMF (neighborhood integrated matrix factorization) model takes advantage of the past Web service usage experience of service users to predict Web service QoS value for users [14]. Wei lo et al. [15] have proposed an extended matrix factorization framework,

this model is quite effective and scales to the large dataset.In [16],Chen et al. have proposed an enhanced QoS prediction approach, which uses A-cosine equation for similarity calculation to remove the impact of different QoS scale and adds a data smoothing process to improve the prediction accuracy , to predict the missing QoS values. Sergio et al. [17] have investigated the Markovian Arrival Processes (MAP) and the related MAP/MAP/1 queueing model as a tool for performance prediction of servers deployed in the cloud.

In addition, all these previous works does not consider the WSDL text content, and ignore the fact that topic distribution of Web service have great influence to predict missing data. Our approach called CQP integrates the matrix factorization with topic model. It can make prediction for any service efficiently no matter whether this service is invoked by users.

	s_1	s_2	s_3	s_4
u_1	1	7	1	2
u_2	2	0	0	1
u_3	1	6	0	0
u_4	3	4	2	0

(a) Warm prediction

	s_1	s_2	s_3	s_4
u_1	5	7	3	0
u_2	2	0	0	0
u_3	0	5	0	0
u_4	3	4	2	0

(b) Cold prediction

Fig. 2: Two tasks of prediction

3 Background

In this section, we describe the traditional matrix factorization to predict QoS value, and review Latent Dirichlet Allocation (LDA) for topic model on text corpora.

3.1 Two tasks of prediction

In the observed QoS values matrix, each dimension stands for I users or J items respectively. In our paper, users are Web service consumers and items are Web services. The element $r_{i,j}$ in the users $\times$ items matrix denotes the QoS value the j_{th} service costs i_{th} user. If the i_{th} user has not invoked the j_{th} Web service, we set $r_{i,j} = 0$ simply. Our target is just to predict the QoS value where the Web service has not been used by any user.

There are two types of prediction scenarios: ***warm prediction*** and ***cold prediction*** as Fig. 2 presents. Fig 2(a) illustrates warm prediction which is the scenario predicting QoS value of Web service consumed by at least one user. This can be addressed with the traditional matrix factorization, and we call this scenario as ***warm prediction***. Fig. 2(b) illustrates cold prediction. In this scenario, some Web services have been used by none user. For example,$r_{i,4} = 0(i = 1,2,3,4)$ denotes that the fourth Web service is a new item in this system and has not been used by any user. This is a cold starting problem. We name this prediction task as ***cold prediction***.

The second scenario is important in practical online system, because the new Web services are released continually, the Web service system are encountering the cold starting problem. Traditional matrix factorization cannot solve this cold prediction problem.

3.2 Probabilistic Matrix Factorization

As a model-based collaborative filtering algorithm, matrix factorization, a latent factor model, performs well in predicting missing data. With the matrix factorization model, the users and items can be presented in a low dimensional space. According to our need, the dimension length of the expressive space can be set as K, this means the i_{th} user and the j_{th} item are represented as $u_i \in R^K$ and $v_j \in R^K$ separately. The missing QoS value can be predicted by the below equation:

$$\hat{r}_{i,j} = u_i^T v_j \tag{1}$$

In general, we can use an approximate framework to gain $U = (u_i)_{i=1}^I$ and $V = (v_j)_{j=1}^J$. In this framework, we need minimize the regularized squared error loss function:

$$min_{U,V} \sum_{i,j} \left(r_{i,j} - u_i^T v_j\right)^2 + \lambda_u \|u_i\|^2 + \lambda_v \left\|v_j\right\|^2 \tag{2}$$

where λ_u and λ_v are regularization parameters avoiding over fitting, and $r_{i,j}$ is the real QoS value.

Ruslan [18] has proposed a probabilistic framework named PFM (probabilistic matrix factorization) for matrix factorization. In Ruslan's framework, matrix factorization can be seen as a generative process.

–For Web services, choose latent vector space V following the probabilistic formula (3)
–For users, choose latent vector space U following the probabilistic formula (4)
–For QoS value matrix generated when services are invoked by users, obtain R following formula (5)

$$p(V|\lambda_v) = \prod_{i=1}^{N} \mathcal{N}\left(V_i|0,\lambda_v^{-1} I_K\right) \tag{3}$$

$$p(U|\lambda_u) = \prod_{i=1}^{N} \mathcal{N}\left(U_i|0,\lambda_u^{-1} I_K\right) \tag{4}$$

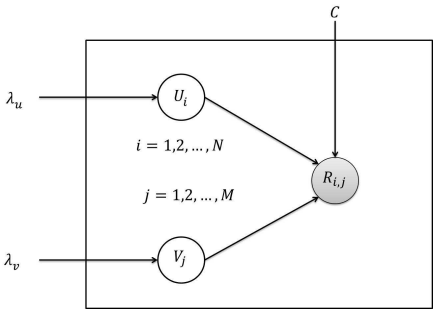


Fig. 3: Probabilistic graphical model of PMF

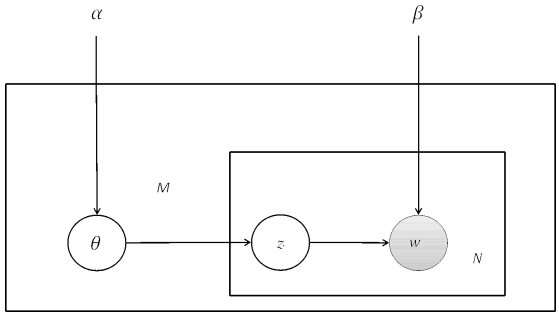


Fig. 4: Graphical representation of LDA

$$P(R|U,V,C) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}\left(R_{i,j}|U_i^T V_i, \left(c_{i,j}^{-1}\right)\right)\right]^{I_{i,j}} \quad (5)$$

Where I_K is a K-dimensional identity matrix, and $\lambda_u^{-1}I_K$, $\lambda_v^{-1}I_K$ and C^{-1} are covariance matrix. This process also can be presented as Fig. 3. In formula (5), Where $c_{i,j}^{-1}$ is the precision parameter for $r_{i,j}$. The larger $c_{i,j}$ is, the more we trust QoS value $r_{i,j}$. Obviously, we trust the QoS value known by us than the missing ones, so we can choose parameters a and b satisfying this inequality$a > b \geq 0$, and set $c_{i,j} = a$ when service j is used by user i, and $c_{i,j} = b$ otherwise.

3.3 Probabilistic Topic Model

Topic models are statistical algorithms aiming to analyze the words of original corpus and discover the "topic" through a large collection of documents [19]. Topic modeling algorithms do not require any prior knowledge and is an unsupervised approach. These models enable us to use the interpretable low-dimension topic vector to express the document topic distribution. These algorithms can be adapted to many fields such as document cluster, information retrieve and social network analysis. The most classical algorithm in probabilistic topic models is Latent Dirichlet Allocation (LDA) [2],[19]. Assume there are K topics, and the vocabulary distribution on each topic k can be represented as a V (the amount of vocabulary) dimensions vector β_k. LDA is a generative model and can be easily described by generative process of one document as below:

1.Choose the document length $N \sim Posisson(\xi)$
2.Choose topic proportions $\theta_j \sim Dirichlet(\alpha)$
3.For each word n:
(a)Choose a topic $z_{j,n} \sim Mult(\theta_j)$
(b)Choose a word $w_{j,n} \sim Mult\left(\beta_{z_{j,n}}\right)$

This statistical model reflects the fact that all the documents share the same topic, but each document can exhibit multiple topics in different proportion. We formally define the document length to be N and a *topic* to be a distribution over a fixed vocabulary. The topics over each document have different proportion from other documents. Each word in each document is drawn through the topic proportion θ_j and vocabulary proportion β.

LDA is very effective and simple to deploy. It does not need prior information and assume the documents are not labeled with any keyword. For a given corpus of documents, we can learn the topics and other parameters with variational EM methods. In our model, we use LDA to learn the topic distribution over each WSDL file to gain a warm start.

4 Collaborative QoS Prediction

4.1 Evaluation Metric

In many previous papers, authors always used the MAE (Mean Absolute Error) as the evaluation metric of missing value estimation. The MAE metric is as follow:

$$MAE = \frac{\sum_{U,S}|r_{u,s} - \hat{r}_{u,s}|}{N}$$

Where $\hat{r}_{u,s}$ is the predicted QoS value of Web service s invoked by user u, the corresponding $r_{u,s}$ is the real QoS value, and N stands for the total number of predicted QoS value.

Because the QoS value range of different services may differ very tempestuously, the MAE in QoS prediction is not rational enough. Wu et al. [20] have used a more impartial metric normalizing the differences range of MAE named NMAE:

$$NMAE = \frac{MAE}{\sum_{U,S}\frac{r_{u,s}}{N}}$$

We use NMAE to evaluate the result in our experiment. The model which has smaller NMAE performs better.

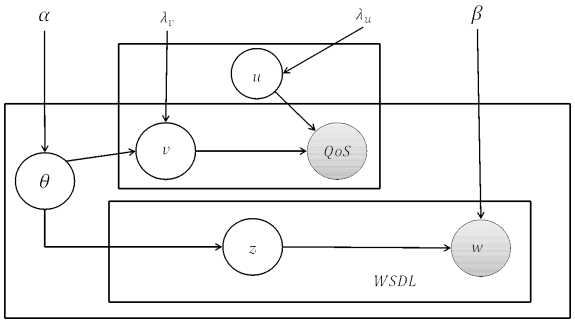


Fig. 5: Graphical representation of CQP

4.2 Architecture

In this section, we describe the collaborative model which utilizes the WSDL files and the observed QoS values.

In order to improve the prediction accurate of matrix factorization, Shan [21] has combined the topic model and matrix factorization, he substituted the item latent vector v_j with the topic proportion θ_j in the equation below:

$$r_{i,j} \approx N\left(u_i^T \theta_j, c_{i,j}^{-1}\right)$$

This model makes use of the topic feature but not item features from matrix factorization and cannot distinguish the importance the topic feature plays in prediction task. For example, there are two Web services, one is about sport ads and the other is about sport news, they have the similar topic feature θ_α and θ_β. As we know, the difference between the QoS values one user spends on two Web services always is big, even very great.

Wang [5] has proposed the collaborative topic regression named CTR, this model can integrate the topic model and matrix factorization together.Wang has applied CTR model to recommend scientific articles to researchers and received excellent recommendation results. The core procedure in CTR is how the article latent vector generates. CTR assumes that this vector follows $v_j = \varepsilon_j + \theta_j$ where θ_j is the topic proportion and ε_j is offset which reflects the degree the latent vector relay on content.

Inspired by CTR from Wang [5], we propose a QoS prediction model integrated matrix factorization and topic model, and we call it CQP (Collaborative QoS Prediction) model. This model can be described as a generative process as following and Fig. 5 is a graphical model example.

1.For each user i, we can draw user latent vector $u_i \sim N\left(0, \lambda_u^{-1} I_K\right)$
2.For each Web service,
(a)Draw topic proportions $\theta_j \sim Dirichlet\left(\alpha\right)$
(b)Draw item latent offset $\varepsilon_j \sim N\left(0, \lambda_v^{-1} I_K\right)$ and set the Web service latent vector as $v_j = \varepsilon_j + \theta_j$
(c)For each meaningful word $w_{j,n}$ in WSDL files,
i.Draw topic assignment $z_{j,n} \sim Mult\left(\theta\right)$
ii.Draw word $w_{j,n} \sim Mult\left(\beta_{z_{j,n}}\right)$
3.For each user-Web service pair (i , j), draw the QoS:

$$r_{i,j} \sim N\left(u_i^T v_j, c_{i,j}^{-1}\right)$$

where the item latent vector v_j is generated by $v_j = \varepsilon_j + \theta_j$, and $r_{i,j}$ stands the Qos value. We can see $v_j = N\left(\theta_j, \lambda_v^{-1} I_K\right)$ from above process because of $\varepsilon_j \sim N\left(0, \lambda_v^{-1} I_K\right)$.

In our model, we assume the Web service latent vector v_j is close to the topic distribution θ_j, and ε_j make v_j also can diverge from θ_j.Thus, we can obtain $r_{i,j}$ from the below formula:

$$\mathbb{E}\left[r_{i,j} | u_i, \theta_j, \varepsilon_j\right] = u_i^T \left(\varepsilon_j + \theta_j\right)$$

Through the experiment, when we add a iterative control condition $NMAE_{new} > NMAE_{old}$, we can reduce the iterative cost and gain good NMAE value.

Algorithm 1 CQP Algorithm

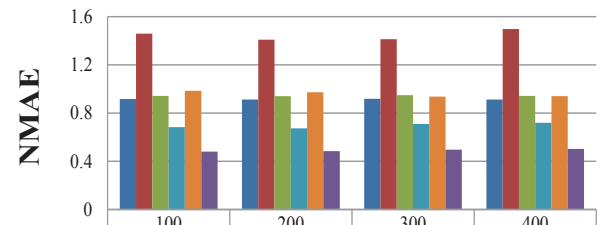
```
Initialize U,V,MaxIterate and ε
Get α,β, and θ from LDA
Compute ℒ_old,NMAE_old
Repeat I = 1 to MaxIterate :
    U_old ← U,V_old ← V
    u_i ← (V C_i V^T + λ_u I_K)^{-1} V C_i R_i
    v_j ← (U C_j U^T + λ_v I_K)^{-1} (V C_j R_j + λ_v θ_j)
    Compute ℒ_new
    Compute NMAE_new
    If abs((ℒ_old − ℒ_new)/ℒ_old) < ε
        Break
    If NMAE_new > NMAE_old
        U ← U_old,V ← V_old
        NMAE_new ← NMAE_old
        Break
    mathcalL_old = mathcalL_new
Return U,V and NMAE_new
```

4.3 Learning the Parameters

From above analysis, we can set $\alpha = 1$ and draw the likelihood function as below:

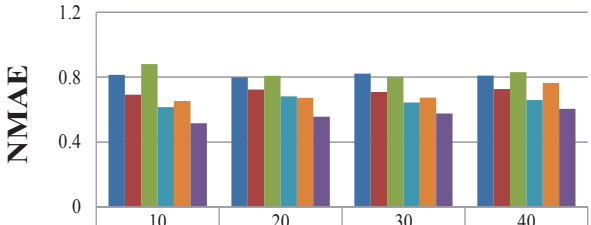
$$\begin{aligned}\mathscr{L} = &-\frac{\lambda_u}{2}\sum_i u_i^T u_i - \frac{\lambda_v}{2}\sum_j \left(v_j - \theta_j\right)^T \left(v_j - \theta_j\right) + \\ &\sum_j \sum_n \log\left(\sum_k \theta_{j,k} \beta_{k,w_{j,n}}\right) - \sum_{i,j} \frac{c_{i,j}}{2}\left(r_{i,j} - u_i^T v_j\right)^2\end{aligned} \tag{6}$$

Maximization of the posterior is equivalent to maximizing Formula (6). Such a nice globally optimal



	100	200	300	400
UERmean	0.9163	0.9125	0.9184	0.9127
ITEMmean	1.4591	1.4088	1.4123	1.4964
LDA-MF	0.9427	0.9403	0.9475	0.9417
User-CF	0.6828	0.6735	0.7098	0.7201
MF	0.9841	0.9725	0.9355	0.9403
Collo-QP	0.4811	0.4837	0.4963	0.5031

(a) NMAE values of warm prediction on big dataset



	10	20	30	40
UERmean	0.8133	0.7992	0.8214	0.8089
ITEMmean	0.691	0.7234	0.708	0.7269
LDA-MF	0.8795	0.8082	0.802	0.8301
User-CF	0.6149	0.6804	0.6431	0.6588
MF	0.6525	0.6723	0.6734	0.7639
Collo-QP	0.5152	0.55632	0.5753	0.6045

(b) NMAE values of warm prediction on small dataset

Fig. 6: NMAE values of warm prediction

solution cannot be directly obtained, based on Wang [5], we can optimize this function following this iterative process:

$$u_i \leftarrow \left(VC_iV^T + \lambda_u I_K\right)^{-1} VC_iR_i \tag{7}$$

$$v_j \leftarrow \left(UC_jU^T + \lambda_v I_K\right)^{-1} \left(VC_jR_j + \lambda_v \theta_j\right) \tag{8}$$

Where C_i is a diagonal matrix with $c_{i,j}\,(j = 1,\ldots,J)$ as diagonal elements and $R_i = (r_{i,j})_{j=1}^{J}$ for u_i. Through the experiment, we find that computing β and θ cost high, and when setting them as fixed values from the result of LDA, we also can obtain comparable performance. This setting also contributes to choosing best parameters faster. The algorithm framework is described as Algorithm 1.

5 Experiment

In this section, we demonstrate our model by analyzing two real-world Web service datasets both of which insist of WSDL files and QoS values.

5.1 Datasets

The QoS values of Web services can be expressed as a users $\times$ Web services matrix. Elements in this matrix indicate the QoS value generated through users' invoking services. The topic model in our model is based on bag of words assumption. We can extract the valuable words which contribute to indicating the Web service purpose from corresponding WSDL file, and the set of words in every WSDL can be regarded as a description file based on bag of words model.

Zibin's dataset [22],[23] is popularly used in a number of papers [15],[20], this is a real-world dataset which only consists of two types of QoS but not WSDL files. In order to discover the topic information of Web services, we crawl the corresponding WSDL files whose URLs (Uniform Resource Locator) have been given in Zibin's dataset from Internet. In this paper, our datasets primarily consist of the users-Web services responding time QoS matrix and WSDL files of corresponding services. In our experiment, the larger dataset consists of 339 users and 2344 Web services, and the smaller one which is a part of the larger dataset consists of 100 users and 150 Web services.

5.2 Evaluation Metric

In many previous papers, authors always used the MAE (Mean Absolute Error) as the evaluation metric of missing value estimation. The MAE metric is as follow:

$$MAE = \frac{\sum_{U,S} |r_{u,s} - \hat{r}_{u,s}|}{N}$$

Where $\hat{r}_{u,s}$ is the predicted QoS value of Web service s invoked by user u, the corresponding $r_{u,s}$ is the real QoS value, and N stands for the total number of predicted QoS value.

Because the QoS value range of different services may differ very tempestuously, the MAE in QoS prediction is not rational enough. Wu et al. [20] have used a more impartial metric normalizing the differences range of MAE named NMAE:

$$NMAE = \frac{MAE}{\sum_{U,S} \frac{r_{u,s}}{N}}$$

We use NMAE to evaluate the result in our experiment. The model which has smaller NMAE performs better.

5.3 Parameter setting

In our model, we set $a = 1, b = 0.01$ empirically. The precision parameter λ_v implies the importance of topic proportion θ_j to draw the Web service representation vector.

In our experiment, we set the precision parameter of $\lambda_u \in \{0.01, 0.1, 1\}$,$\lambda_v \in \{0.01, 0.1, 1, 10, 100, 1000\}$ .As discussed in Section IV, the larger λ_v is, the more strongly v_j relies on θ_j .

Different from Wang [5]' CTR, we simply fix θ and β as the result from LDA. Through this simplification, our model can save much computation cost and is more efficient.

For traditional matrix factorization prediction model, we set $K = 50, 80, 100$ for the larger dataset in our experiment, and find that $K = 80$ gives the best performance. In a similar way, when $K = 10, 30, 50, 80, 100$ for smaller dataset, $K = 10$ performs best. In order to make the comparison between our algorithm and other methods perform well, we set the number of topics $K = 80$ for the experiment on the larger dataset and $K = 10$ for the experiment on the smaller dataset in our approach.

5.4 Comparisons

In this section, we compare our proposed approach with the following representative prediction algorithm:

- **–USERmean.** This is a classical method that utilizes similar user behavior to make prediction.
- **–ITEMmean.** This method captures similar service attributes to make predictions.
- **–LDA-MF.** LDA-MF approach replaces v_j with θ_j for all Web service. As discussed in previous section, LDA-MF captures the user vector from MF and item vector from LDA.
- **–User-based CF**. User-based collaborative filtering is a traditional prediction method, Chen [20] has proposed an improved CF algorithm named neighborhood CF. This method also cannot solve the cold prediction problem.
- **–MF.** Traditional matrix factorization is a model-based collaborative filtering, and can generate user and item latent vector directly in warm prediction.

For the warm prediction, in order to make our experiment more realistic, we vary the number of missing data to be predicted of every user $M = 100, 200, 300, 400$ for the larger dataset which consists of 2344 Web services, and $M = 10, 20, 30, 40$ for the smaller dataset which consists of 150 Web services.

For the cold prediction, the number of new Web services not invoked by any user has little impact on experiment result, so we fix this number is equal to 50 for the larger dataset and 15 for the smaller dataset, and set the terminal convergence is equal to 0.005 empirically. The comparative approaches and our algorithm are conducted on the same datasets and share the same common parameters.

We use coordinate descent method to get the optimal solution, considering iteration speed, so we set the max iterations be 200 for MF and our model named Collo-QP.

5.5 Warm Prediction

We conduct the warm prediction with six methods: User-based CF, LDA-MF, USERmean, ITEMmean, MF, and our Collo-QP model.

Fig. 6 shows the result of warm prediction in the two datasets (Fig. 6(a) for the larger dataset and Fig. 6(b) for the smaller one). We can observe that our approach Collo-QP performs best, its NMAE value is always smaller than others in large scale in four cases.

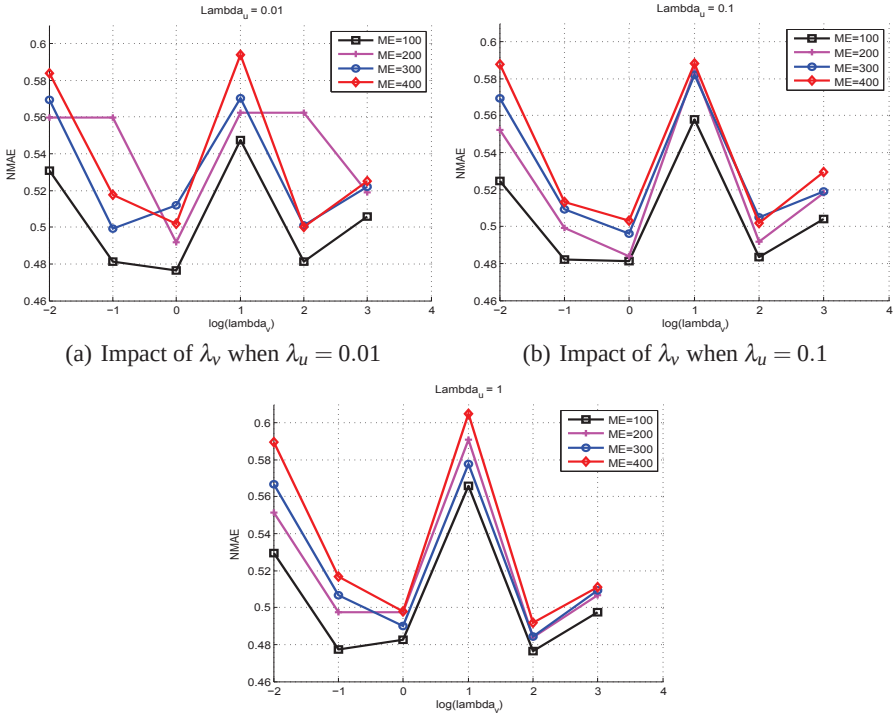
USERmean method uses the mean QoS value based on all records of each user, when the number of Web services this user invoked is larger than the number he does not invoked, its NMAE value will keep stable, for example, in the larger dataset with missing elements increasing, NMAE value is around 0.9150, and in the smaller dataset NMAE value is around 0.8100. Because of the litter noise invoked history in smaller dataset, we can find that this method performs better in smaller dataset.

Similarly, ITEMmean method uses the average QoS value based on all records of each Web service to make prediction. NMAE value of this approach is high than any other algorithm in the experiment on the larger dataset, because the difference of QoS between users for the same service is large. While in the experiment on the smaller dataset, this method also performs bad. This means that the average value cannot represent the real value of every service.

LDA-MF method is rough and ignores the influence from QoS records to Web service representative vector. It takes topic distribution from WSDL file as service vector simply. As we all know, the number of words in WSDL file is very small, for example, the number of many Web services' effective vocabulary are not more than 20, these Web services' topic distribution cannot represent Web services suitably. From Fig. 6, we can observe that NMAE value of LDA-MF is very high, and its performance is even worse than USERmean. This approach also keeps stable along with missing elements increasing.

In the five contrast approaches, User-based CF performs best. This approach uses top-k similar users selection algorithm, and predicts missing data with records of similar users.

Basic Matrix Factorization algorithm cannot perform well in our experiment. From Fig. 6, the NMAE value of MF method is high than LDA-MF, User-based CF and our Collo-QP model in most cases.



(a) Impact of λ_v when $\lambda_u = 0.01$

(b) Impact of λ_v when $\lambda_u = 0.1$

(c) Impact of λ_v when $\lambda_u = 1$

Fig. 7: Impact of λ_u and λ_v On Big Dataset

Table 2: NMAE VALUES OF DIFFERENT λ_u AND λ_v ON BIG DATASET

λ_v	missing elements = 100			missing elements = 200			missing elements = 300			missing elements = 400		
	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$
0.01	0.5307	0.5245	0.5294	0.5598	0.5525	0.5513	0.5694	0.5693	0.5666	0.584	0.5877	0.5896
0.1	0.4813	0.4823	0.47721	0.5598	0.4991	0.4974	0.4993	0.5092	0.5067	0.5175	0.5132	0.517
1	0.4766	0.4811	0.4827	0.4917	0.4837	0.4976	0.5118	0.4963	0.4902	0.5017	0.5031	0.4978
10	0.5476	0.5581	0.5658	0.5624	0.5844	0.5908	0.5702	0.5823	0.5777	0.5942	0.5884	0.605
100	0.4814	0.4834	0.4765	0.5624	0.4917	0.4839	0.5008	0.5049	0.4842	0.5001	0.5018	0.4918
1000	0.5059	0.5041	0.4977	0.5189	0.5183	0.5066	0.5219	0.519	0.5093	0.5252	0.5293	0.511

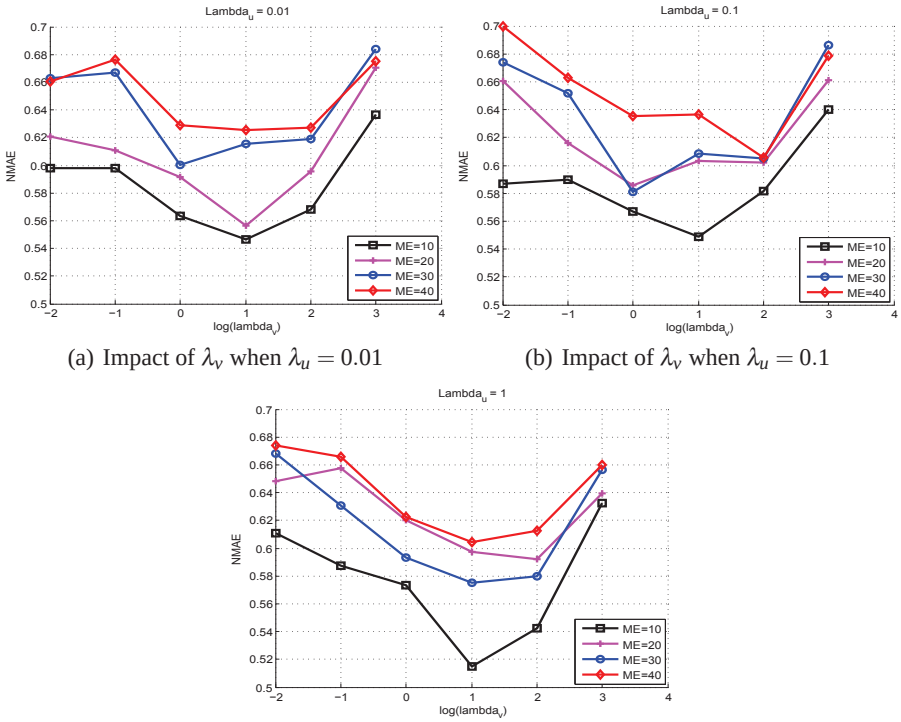
Table 3: NMAE VALUES OF DIFFERENT λ_u AND λ_v ON SMALL DATASET

λ_v	missing elements = 10			missing elements = 20			missing elements = 30			missing elements = 40		
	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$	$\lambda_u = 0.01$	$\lambda_u = 0.1$	$\lambda_u = 1$
0.01	0.5982	0.5869	0.6106	0.621	0.6607	0.648	0.6629	0.674	0.6682	0.6604	0.6995	0.6741
0.1	0.5982	0.5896	0.5877	0.611	0.6158	0.6577	0.6668	0.6515	0.6309	0.6762	0.6631	0.6658
1	0.5634	0.5668	0.5733	0.5918	0.5854	0.6199	0.6005	0.5808	0.5933	0.6288	0.6354	0.6226
10	0.5467	0.5486	0.5152	0.5563	0.603	0.5972	0.6157	0.6083	0.5752	0.6254	0.6367	0.6045
100	0.5682	0.5817	0.5424	0.5954	0.602	0.5923	0.619	0.6052	0.5799	0.6274	0.6057	0.6125
1000	0.6366	0.6403	0.6323	0.6707	0.6612	0.6393	0.6838	0.6863	0.6567	0.6753	0.6784	0.6599

It is obvious that our algorithm obtains best performance. In Fig. 6(a), we set $\lambda_u = 0.1$ and $\lambda_v = 1$, and in Fig. 6(b), we set $\lambda_u = 1$ and $\lambda_v = 10$. Because we find that in these setting our algorithm can obtain good performance empirically.

Table 2 and table 3 show that Collo-QP' NMAE values change with λ_v increasing when λ_u is equal to 0.01, 0.1 and 1, and Fig. 7 and Fig. 8 also describe these results respectively. From these figures, we can see that our algorithm can obtain best performance where $\lambda_v = 1$ in the larger dataset and $\lambda_v = 10$ in the smaller dataset.

From table 2 and figure 7, we can see that if λ_v is fixed, NMAE values is stable when we vary $\lambda_u = 0.01, 0.1, 1$. From table 3 and figure 8, we can also



(a) Impact of λ_v when $\lambda_u = 0.01$

(b) Impact of λ_v when $\lambda_u = 0.1$

(c) Impact of λ_v when $\lambda_u = 1$

Fig. 8: Impact of λ_u and λ_v On Small Dataset

observe the stability of λ_u when λ_v is fixed. but when λ_u is fixed in each case, NMAE values have much difference among different λ_v values. The larger λ_v is, the more strongly Web service representative vector relies on topic distribution. As we discussed above, these results prove that QoS records and topic distribution affect Web service expression together.

From figure 7 and figure 8, we can also find that the λ_v and λ_u values which perform best in two experiment differ badly. This shows that λ_v and λ_u is an empirical parameter, and in order to gain good performance, we should try many times to decide these parameters.

5.6 Cold Prediction

Because the new services are not invoked by any user, there are no records to help us predict their QoS values. We extract Web service representative vector from WSDL and use topic distribution to represent Web service.

In order to compare the influence of user representative vector generating process when we use topic distribution θ_j replace λ_v, we make cold prediction with LDA-MF and Collo-QP method. In this contrast experiment, we set $\lambda_u = 0.1$ and $\lambda_v = 1$ for the larger dataset and $\lambda_u = 1$ and $\lambda_v = 10$ for the smaller dataset, because we have known our model can perform well generally in these setting .

Table 4 and Table 5 show the results from the larger dataset and the smaller one respectively. we can find that our model can perform better than LDA-MF in both dataset. The user vectors of LDA-MF are from matrix factorization, however, Collo-QP' user vector is generating process considering matrix factorization and topic model. Formula (7) obviously exhibit this process. From the two tables, we also can find that both of LDA-MF and Collo-QP perform better in the larger dataset than the smaller dataset. The reason causes these results is that the larger dataset consists more words, so the topic distribution on the larger dataset is more accurate.

Table 4: NMAE VALUES OF COLD PREDICTION ON LARGER DATASET

method	ME=100	ME=200	ME=300	ME=400
Collo-QP	1.0909	1.0812	1.0887	1.0953
LDA-MF	1.1141	1.1121	1.1138	1.1122

6 Conclusion

In this paper, we propose a collaborative QoS prediction framework named CQP integrating matrix factorization

Table 5: NMAE VALUES OF COLD PREDICTION ON SMALLER DATASET

method	ME=10	ME=20	ME=30	ME=40
Collo-QP	1.2527	1.2547	1.2183	1.1739
LDA-MF	1.3637	1.3531	1.3737	1.3879

with topic model. The proposed model extracts topic distribution form WSDL files and can make effective prediction in cold-start scenario. To validate our methods, some classical approaches and our algorithm are conducted on two real-world datasets which consisting of QoS records and WSDL files of corresponding Web services. The empirical experiment and analysis show that the proposed CQP outperforms other methods in QoS prediction accuracy.

In the future work, we will integrate influence of Web services tags and categories of users and services with our framework.

Acknowledgment

This research was partially supported by the National Technology Support Program under grant of 2011BAH16B04, the National Natural Science Foundation of China under grant of 61173176, National Key Science and Technology Research Program of China 2013AA01A604.

References

[1] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," in *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pp. 881–890, 2009.

[2] 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)*, vol. 1, no. 1, 2007.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, no. 3, pp. 993–1022, 2003.

[4] W3C, "Web services description language (wsdl) 1.1." http://www.w3.org/TR/wsdl, 2001.

[5] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 448–456, 2011.

[6] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality driven web services composition," in *Proceedings of 12th International Conference on World Wide Web (WWW)*, pp. 411–421, 2003.

[7] S. Ran, "A model for web services discovery with qos," *ACM SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.

[8] 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*, vol. 30, no. 5, pp. 311–327, 2004.

[9] Y. Liu, A. H. Ngu, and L. Z. Zeng, "Qos computation and policing in dynamic web service selection," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters*, pp. 66–73, 2004.

[10] M. Serhani, R. Dssouli, A. Hafid, and H. Sahraoui, "A qos broker based architecture for efficient web services selection," in *Proceedings of the 2005 IEEE International Conference on Web Service*, pp. 113–120, 2005.

[11] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction for web services via collaborative filtering," in *Proceedings of International Conference on Web Service*, p. 439C446, 2007.

[12] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," *IEEE Trans. Service. Computing*, vol. 4, no. 2, p. 140C152, 2011.

[13] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proceedings of International Conference on Web Service(ICWS)*, pp. 9–16, 2010.

[14] Z. Zheng, H. Ma, M. Lyu, and I. King, "Collaborative web service qos prediction via neighborhood integrated matrix factorization," *IEEE Trans. Service. Computing*, vol. 1, no. 1, 2012.

[15] W. Lo, J. W. Yin, S. Deng, Y. Li, and Z. H. Wu, "An extended matrix factorization approach for qos prediction in service selection," in *Proceedings of IEEE 9th International Conference on service computing*, pp. 162–169, 2012.

[16] L. Chen, Y. Feng, J. Wu, and Z. Zheng, "An enhanced qos prediction approach for service selection," in *Proceedings of the 2011 IEEE International Conference on Services Computing*, pp. 727–728, 2011.

[17] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson, "Markovianworkload characterization for qos prediction in the cloud," in *Proceedings of the 2011 IEEE International Conference on Cloud Computing*, pp. 147–154, 2011.

[18] S. R and M. A, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1257–1264, 2008.

[19] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[20] J. Wu, L. Chen, Y. P. Feng, Z. B. Zheng, M. C. Zhou, and Z. H. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Transactions on Systems,Man,and Cybernetics:Systems*, vol. 43, no. 2, pp. 428–439, 2013.

[21] H. Shan and A. Banerjee, "Generalized probabilistic matrix factorizations for collaborative filtering," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, p. 1025C1030, 2010.

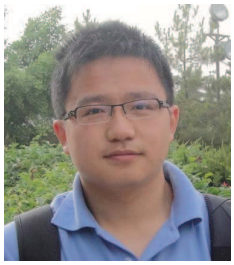
[22] Z. B. Zheng, Y. L. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *Proceedings of the 8th International Conference on Web Services (ICWS)*, pp. 83–90, 2010.

[23] Y. L. Zhang, Z. B. Zheng, and M. R. Lyu, "Exploring latent features for memory-based qos prediction in cloud computing," in *Proceedings of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pp. 1–10, 2011.

Jian Wu received his B.S. and Ph.D. Degrees in computer science from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. He is currently an associate professor at the College of Computer Science, Zhejiang University, and visiting professor at University of Illinois at Urbana-Champaign. His research interests include service computing and data mining. He is the recipient of the second grade prize of the National Science Progress Award. He is currently leading some research projects supported by China National Natural Scientific Foundation and National High-tech R&D Program of China (863 Program).

Lichuan Ji received the B.S. degree in information and computer science from Huazhong Agricultural University, Wuhan, China, in 2012. He is currently working toward the Master degree in the College of Computer Science, Zhejiang University.His research interests include recommender system, service computing and topic model. He is a ACM student member now.

Tingting Liang received the B.S. degree in the school of science, Jiangnan University, Wuxi, China, in 2013. She is currently working toward the Ph.D degree in the College of Computer Science, Zhejiang University, Hangzhou. Her research interests include service computing, machine learning and data mining.



Liang Chen received the B.S. degree in computer science from Zhejiang University, Hangzhou, China, in 2009. He is currently working toward the Ph.D. degree in the College of Computer Science, Zhejiang University. His publications have appeared in some well-known conference proceedings and international journals. He also served as a Reviewer for some international conferences and journals (CIKM, TSMC, TSC, ICWS, etc). His research interests include service computing and data mining. He received the award of ”Excellent Intern” from Microsoft Research Asia in 2010.