

# Multi-user web service selection based on multi-QoS prediction

Shangguang Wang · Ching-Hsien Hsu · Zhongjun Liang ·  
Qibo Sun · Fangchun Yang

Published online: 4 October 2013  
© Springer Science+Business Media New York 2013

**Abstract** In order to find best services to meet multi-user's QoS requirements, some multi-user Web service selection schemes were proposed. However, the unavoidable challenges in these schemes are the efficiency and effect. Most existing schemes are proposed for the single request condition without considering the overload of Web services, which cannot be directly used in this problem. Furthermore, existing methods assumed the QoS information for users are all known and accurate, and in real case, there are always many missing QoS values in history records, which increase the difficulty of the selection. In this paper, we propose a new framework for multi-user Web service selection problem. This framework first predicts the missing multi-QoS values according to the historical QoS experience from users, and then selects the global optimal solution for multi-user by our fast match approach. Comprehensive empirical studies demonstrate the utility of the proposed method.

**Keywords** Web services · Service selection · QoS · Multi-user · QoS prediction

---

S. Wang (✉) · Z. Liang · Q. Sun · F. Yang  
State Key Laboratory of Networking and Switching Technology,  
Beijing University of Posts and Telecommunications,  
Beijing 100876, China  
e-mail: sgwang@bupt.edu.cn

Z. Liang  
e-mail: kkljiangjun@163.com

Q. Sun  
e-mail: qsun@bupt.edu.cn

F. Yang  
e-mail: fcyang@bupt.edu.cn

C.-H. Hsu  
Department of Computer Science and Information Engineering,  
Chung Hua University, Hsinchu 707, Taiwan  
e-mail: chh@chu.edu.tw

## 1 Introduction

Web services are designed as computational components to build service-oriented distributed systems (Zeng et al. 2003). The rising development of service-oriented architecture makes more and more alternative Web services offered with equivalent or similar functions. Therefore, how to select a feasible service to meet the demands of different users has become a popular research area.

In order to improve the customer's satisfaction, many studies (Zeng et al. 2003; Zeng et al. 2004; Canfora et al. 2005; Alrifai and Risse 2009; Alrifai et al. 2010) focused on the Quality of Service (QoS) optimization. However, most existing selection methods are proposed on the single request condition, which cannot be directly used on multi-user condition, due to the unavoidable challenges in this problem are shown as follows.

- 1) **Effect.** As the ability of a service is limited, the service may not be able to hold the guarantee as it provides, when the requests are overloading. Specially speaking, if a single service instance takes requests more than its ability threshold, then its QoS values may decline rapidly, which will lead requesting users a bad QoS experience. Furthermore, due to the various locations and communication links, different users could have different QoS experiences when invoking even the same Web service, and the consuming time and cost make it expensive to get all information by invocation. In real case, there are always many missing QoS values in history records; this will increase the difficulty of the selection. While existing methods always assume the QoS information for users are all known and accurate, which might be something different with actual scenarios. Therefore, how to efficiently select the Web service for multi-user is the first challenge in our problem.
- 2) **Efficiency.** In practical world applications, users require less time-consumption in Web service selection. However, the time complexity of existing methods for

multi-user Web services selection problem is exponential with the number of multi-user and candidate services. While with the extensive applications of Web services, the two numbers grow bigger and bigger, which leads the expensive time cost. Therefore, how to efficiently select the Web service for multi-user is the second challenge in our problem.

- 3) **Simultaneity.** In Web service selection applications, the users want to obtain multi-QoS at same time. However, most existing methods cannot deal with this case, as they had to predict each individual QoS value one by one, and QoS- prediction is the foundation issue of Web service selection. Therefore, how to simultaneously calculate the multi-QoS for multi-user is the third challenge in our problem.

In this paper, we propose a new framework for multi-user Web service selection problem, which aims to effectively and efficiently select feasible services to meet the demands of different users. In this framework, we first propose a multi-QoS prediction method, which can predict personalized QoS values according to the historical QoS experience collected from different users, and then we propose a fast match approach to efficiently select the global optimal solution for multi-user. The contributions are listed as follows.

- 1) We propose a new framework for the multi-user Web service selection problem, which aims to effectively and efficiently select feasible services to meet the demands of different users. Using this framework, each user can be provided with his/her suitable service, to avoid overload of selected services in a short time.
- 2) We design a multi-QoS prediction method, to simultaneously predict the multi-QoS according to the historical QoS experience from different users. In order to guarantee the accuracy effect, this method takes the relationship between QoS attributes into consideration. Specially, it first normalizes the QoS attribute values, then employs the Non-negative Matrix Factorization to extract the feature of users from all QoS attributes together, and last predicts the missing multi-QoS value observed from different users via Multi-output Support Vector Regression algorithm.
- 3) We design a fast match approach (FMA for short) to select the global optimal solution for the multi-user. It first respectively calculates the optimal matching Web service for each user, then checks the load conditions in this local selection plan. If any Web service could be overloaded, then we transform this problem to a maximum weight matching problem and employ the Kuhn-Munkres algorithm (KM for short) to select the global optimal solution. Its complexity of the time can be  $O(n \times m)$  in the best situation, where  $n$  and  $m$  represent the number of users and Web services respectively.

- 4) We conduct several experiments on real-world Web service dataset and synthetic dataset. Comprehensive empirical studies demonstrate the utility of the proposed method.

The rest of paper is organized as follows: Section 2 highlights the related work. Section 3 formulates the problem, and describes our framework. Section 4 details the multi-QoS prediction method and Section 5 details the fast match approach. Section 6 presents the experimental result. We conclude the paper in Section 7.

## 2 Related work

In this section, we will briefly introduce the related work in Web service selection area. There is a line of work focused on the Quality of Service (QoS) optimization and efficient Web service selection methods, aims to improve the customers satisfaction.

Back to a decade ago, Zeng et al. in (Zeng et al. 2003; Zeng et al. 2004) first transferred the QoS-based service selection problem into an optimization problem. They used (mixed) linear programming techniques to find the optimal selection of component services. However, it suffered from poor scalability, due to the exponential time complexity of the applied search algorithms. Motivated by real-time selection with large number of service, Canfora et al. in (Canfora et al. 2005) employed GA to solve Web service selection problem. Alrifai et al. in (Alrifai and Risse 2009) combined the global optimization with local selection in order to find a close-to-optimal selection efficiently. After that, they in (Alrifai et al. 2010) proposed a new method to select and use representative skyline services for the composition. However, all above works are mainly focusing on service selection for a single user, when too many requester select the same best Web service, the conflicts may occur.

Recently, Shahand et al. (Shahand et al. 2010) and Dyachuk and Deters (Dyachuk and Deters 2006) considered the multiple requests waiting in a queue, and Kang et al. (Kang et al. 2011) used Euclidean distance with weights to measure degree of matching of services based on QoS, they employed a 0–1 integral programming model to solve problem of optimal service selection for multiple requesters (GOSSMR). However, the exponential time complexity of the applied search algorithms could influence the customers satisfaction. In addition, it supposed the QoS information for requesters are all known and accurate, and the real case is that it is very hard to get all requesters' QoS information from users' perspective (Lo et al. 2012).

To get personalized QoS, a few works attempted to predict the missing values. In (Shao et al. 2007), Shao et al. first employed collaborative filtering technique to predict the QoS of Web service. They proposed a user-based CF algorithm to predict QoS values. Zheng et al. (Zheng et al. 2009)

proposed a hybrid user-based and item-based CF algorithm to predict the personalized QoS, and they carried out a series of large-scale experiments based on real Web services dataset.

In (Jiang et al. 2011), Jiang et al. proposed that the influence of personalization of Web service items should be taken into account when computing degree of similarity between users. That is, more popular services or services with more stable QoS from user to user should contribute less to user similarity measurement. Zhang et al. (Zhang et al. 2010) suggested that it was better to combine users' QoS experiences, environment factor and user input factor to predict Web services QoS values. But how to obtain environment factor and user input factor were not discussed. Chen et al. (Chen et al. 2010) discovered the great influence of user's location to the accuracy of prediction and proposed a region based hybrid Collaborative Filter algorithm to predict the QoS of services. This method groups users into a hierarchy of regions according to users' locations and their QoS records, so that the users in a region are similar. When identifying similar users for a target user, instead of searching the entire set of users, the method only searches the regions that the target user belongs to.

Tang et al. (Tang et al. 2012) proposed a hybrid location-aware QoS prediction method, which not only consider the location of users but also the services' relationship. Lo et al. (Lo et al. 2012) proposed an extended Matrix Factorization framework with relational regularization to make missing QoS values prediction. They added a user-based regularization term and a service-based regularization term in the minimizing objective function. However, existing algorithms make QoS prediction without considering the relationship between QoS attributes. And they predicts value individual QoS one by one, which cannot scale to multi-user service selection problem.

Different from previous research, we propose a new framework for multi-user Web services selection problem. In this framework, we first propose a prediction method by considering the relationship between QoS attributes, which can predict multi-QoS at same time, then propose a fast match approach to select the global optimal solution for multi-user.

### 3 Proposed framework

In this section, we present the problem formulation and our framework for multi-user Web service selection problem.

#### 3.1 Problem formulation

The goal of multi-user Web service selection is to select feasible services to meet the demands of different users, which not only do the global selection solution ensure that each user get his suitable service, but it also avoids QoS declining caused by requests overloading. To better describe the multi-

user Web selection problem, we first introduce some terminology, and then define the problem.

- 1) **QoS**. QoS is the quantitative non-functional properties of web services, which can be used to describe the quality criteria of a web service, such as response time, price and availability etc..
- 2) **QoS Aggregation**. QoS aggregation is to evaluate the utility for the user. It can map multi-dimension attributes into one single real value from user's perspective. In this paper, we employ an aggregation function to calculate the utility value by the following:

$$Utility(u_i, s_j) = \sum_{k=1}^m w_{ik} * q_j^k \tag{1}$$

where  $\sum_{k=1}^m w_{ik}=1$ ,  $q_j^k$  is a QoS attribute value of the web service  $s_j$ ,  $w_{ik}$  denotes for the weight of  $k$ -th QoS attributes on user  $u_i$ 's preference.

- 3) **Processing Capacity**. Processing capacity of Web service  $pc_j$  is the maximum request number that the Web service  $s_j$  can be able to process at the present.

Assume multiple users  $U = \{u_1, u_2, \dots, u_n\}$  request for the same functional web service  $S$  with different QoS preference at same time, and there are  $m$  candidate Web service satisfied that functional requirement, the multi-user Web service selection problem is to select feasible services for the requestors based on the personalized QoS values, and its objective is to maximize the following global utility function.

$$\max \sum_{i=1}^n \sum_{j=1}^m Utility(u_i, s_j) * x_{ij} \tag{2}$$

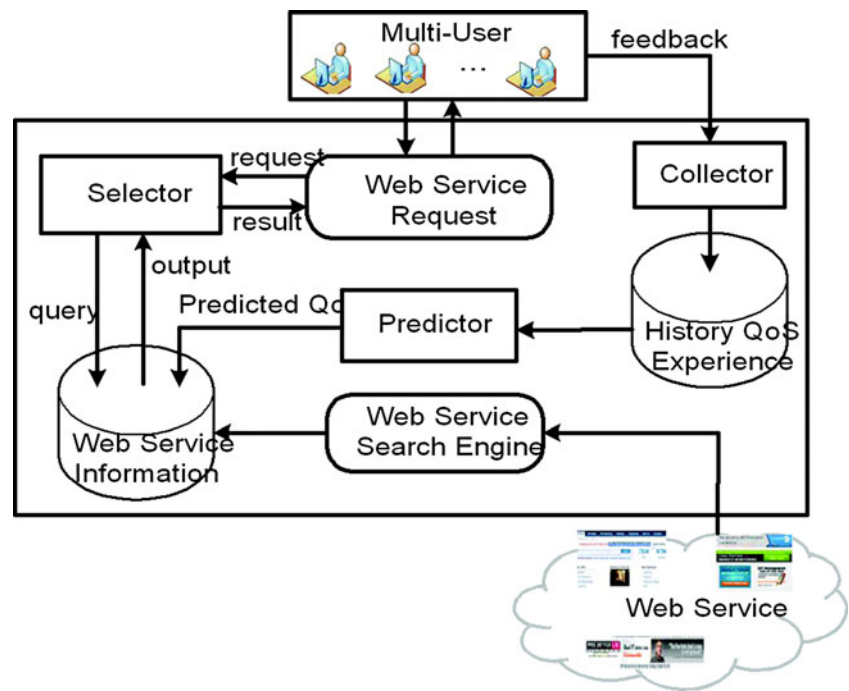
where  $x_{ij}$  represents the select plan for user  $u_i$ , if user select the Web service  $s_j$ , then  $X_{ij} = 1$ ; otherwise  $X_{ij} = 0$  ( $\sum_{i=1}^n X_{ij} \leq pc_j$ ).

#### 3.2 Our framework

As presented in the previous section, determining the best candidate services for different users is an optimization problem, which is based on the known multi-QoS values. However this is unlikely in practical, different users will have different QoS experiences when invoking even the same Web service, due to their various locations and communication links, and it is expensive to got all their QoS information from the users' perspective (Lo et al. 2012).

To address this problem, we propose a multi-user Web service selection framework, which can predict missing multi-QoS values to support the Web service selection. As shown in Fig. 1, our framework includes following modules: Web service search engine, Collector, Predictor and Selector . The Web service search engine searches Web services from the Internet and adds them into the Web service database. The Collector collects the local QoS records feedback from different Web services users and stores them in the historical QoS

**Fig. 1** Our framework for multi-user Web service selection



experience database. The Predictor predicts different users' personalized multi-QoS values. The selector gives the global optimal select plan for multiple requester, which can not only give the global solution for multi-users, but also avoid overload of selected services.

In our framework, the Predictor and Selector are the key modules to improve user experience, their performance are depend on the prediction algorithm and search algorithm. Next, we will give the detail in Section 4 and 5.

#### 4 Multi-QoS prediction

In multi-user Web service selection, the selector always wants to obtain multi-QoS value at same time since the global optimal solution are given based on many attributes. However, existing QoS predicted algorithms are all focused on predicting individual QoS one by one, which cannot scale to the need of our selection. To dress this problem, we propose a new prediction algorithm which can not only predict multi-QoS value at same time, but also can provide a good predicting precision for different users.

The personalized QoS experience, obtained from a user after the invocation, is co-determined by the feature of corresponding invoked Web service and the user. Therefore, for a target Web service, the different feature of these users is the main reason for the different QoS experience. In this paper, we predict the missing multi-QoS based on the extracted feature of users and historical QoS experience of users. Figure 2 shows the simplified procedure of our multi-QoS prediction

approach. It contains QoS normalization step, feature extraction step, and QoS prediction step.

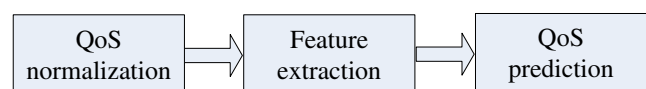
QoS normalization step, which aims to normalize the QoS attribute values by Gaussian method (Ortega et al. 1997). This step can map the QoS values to the interval  $[0,1]$ , which prepares for feature extraction.

Feature extraction step, which aims to extract the feature of users from existing multi-QoS values. This step can handle the acquired sparse multi-QoS information.

QoS prediction step, which aims to predict the missing Multi-QoS values via Multi-output Support Vector Regression algorithm. This step finally solves our problem based on the extraction feature of users and the other users' historical QoS experience for a target Web service.

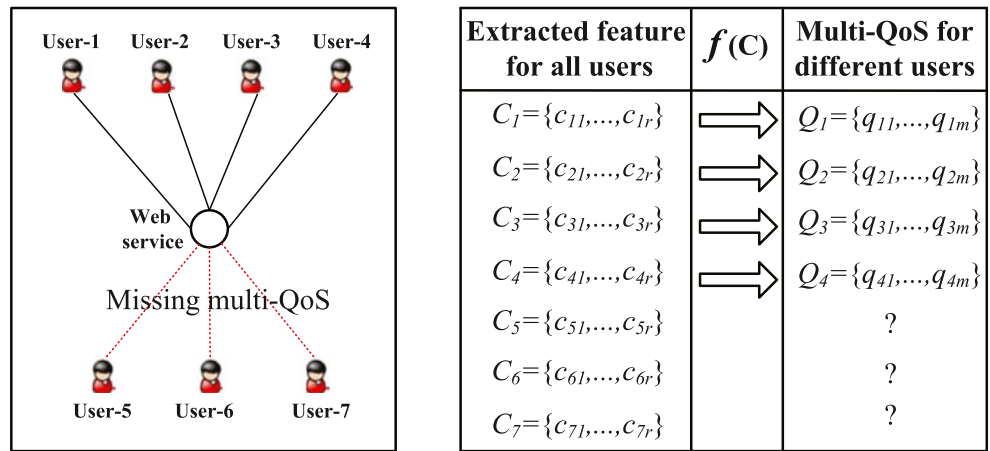
##### 4.1 QoS normalization

To map the different QoS attributes to a same range of values, we employ Gaussian method (Ortega et al. 1997) to normalize the QoS attribute values for feature extraction. It can not only map the QoS values to the interval  $[0,1]$ , but also can better avoid the influence of abnormal values (such as high value or low value) compared with other normalization methods (such as Extremum Regularization Method). The detail can be described as follow.



**Fig. 2** Procedure of multi-QoS prediction approach

Fig. 3 Prediction example



First, we normalize the Multi-QoS attribute values in each dimension by the following:

$$Q_{ij}^k = 0.5 + \frac{q_{ij}^k - \hat{q}_{ij}^k}{2 \times 3\sigma} \quad (3)$$

Where  $Q_{ij}^k$  is normalized value of  $k$ -th QoS attribute,  $\hat{q}_{ij}^k$  denotes the average value of  $k$ -th QoS attribute obtain from all users,  $\sigma$  is the standard deviation of  $k$ -th QoS attribute.

Second, assign  $Q_{ij}^k$  with 0 or 1 based on the distance with 0 or 1, if it is out of the  $[0,1]$  interval. It has been proved that 99% data can be directly mapped to the  $[0,1]$  interval (Ortega et al. 1997).

#### 4.2 Feature extraction

Since the multi-QoS attributes reflect the feature of corresponding invoked Web service and the user, we extract the feature based on the historical QoS experience. To better reflect the feature of user, we extract it with multi-QoS attributes together. Due to the existing QoS matrix is always sparse, we employ matrix factorization to extract the features of users. The main idea of this method is to derive a high-quality low-dimensional matrix to approximate the historical

QoS data, which represent the feature of Web service and user respectively.

Consider multi-QoS matrix  $Q = \{Q^1, Q^2, \dots, Q^m\}$  ( $Q \in \mathbb{R}^{1 \times m \times n}$ ) is consisting of  $l$  users,  $n$  Web services and  $m$  QoS attributes. The extraction step is to factorize  $Q$  into two matrices  $C \in \mathbb{R}^{1 \times m \times r}$  and  $W \in \mathbb{R}^{r \times n}$ , which satisfy the following:

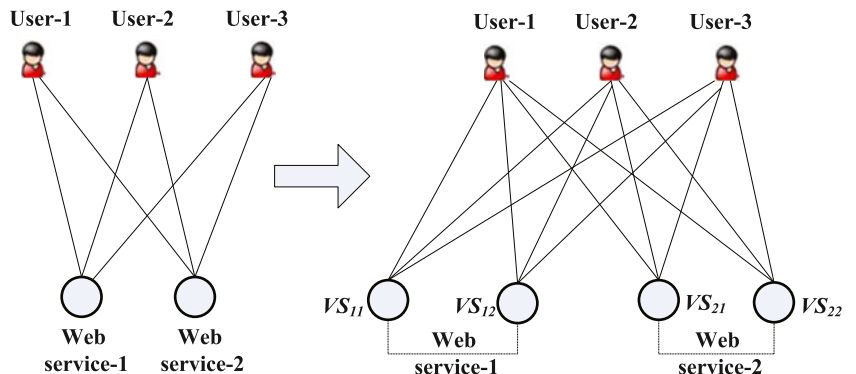
$$Q_{ij} = \{Q_{ij}^1, Q_{ij}^2, \dots, Q_{ij}^m\} \approx \left\{ \sum_{k=1}^r c_{ik} w_{jk}^1, \sum_{k=1}^r c_{ik} w_{jk}^2, \dots, \sum_{k=1}^r c_{ik} w_{jk}^m \right\} \quad (4)$$

Where  $Q_{ij}$  is the multi-QoS performance of Web service  $s_j$  observed by user  $u_i$ , vector  $c_i = [c_{i1}, c_{i2}, \dots, c_{ir}]$  reflects the feature of user  $u_i$ , matrix  $w_j = [w_j^1, w_j^2, \dots, w_j^m]$  reflects the feature of Web service  $s_j$ , vector  $w_j^d = [w_{ji}^d, w_{j2}^d, \dots, w_{jr}^d]$  is its sub feature,  $r$  is the rank of matrix  $C$  and  $W$ , it generally satisfies  $(l \times m + n) \times r < m \times l \times n$ .

Based on the following equation, we put multi-QoS attribute values together and use NMF algorithm (Lee and Seung 1999) to learn the feature of Web services. The detail can be described as follow:

First, we put normalized multi-QoS attribute values together to form a supermatrix  $Q^* = [Q^1, Q^2, \dots, Q^m]$

Fig. 4 Transform example



**Table 1** Detail of QoS data set

Statistics	Response time	Throughput
Scale	1-20s	1-1000 kbps
Mean	0.910 s	47.386 kbps
Num. Users	339	339
Num. Web Services	5,825	5,825

Second, we decompose the  $Q^*$  to extract the matrix  $C$  and  $W$  with the objective function  $Minf(C, W)$  by the following:

$$Min f(C, W) = \sum_{h=1}^{l \times m} \sum_{j=1}^n I_{ij} [Q_{ih}^* \times \log(CW)_{ih} - (CW)_{ih}] \quad (5)$$

where  $I_{ij}$  is the indicator function that is equal to 1, if user  $u_i$  invoked service  $s_j$  and 0 otherwise. Then, an incremental gradient descent method is employed to extract the feature of Web service by the following:

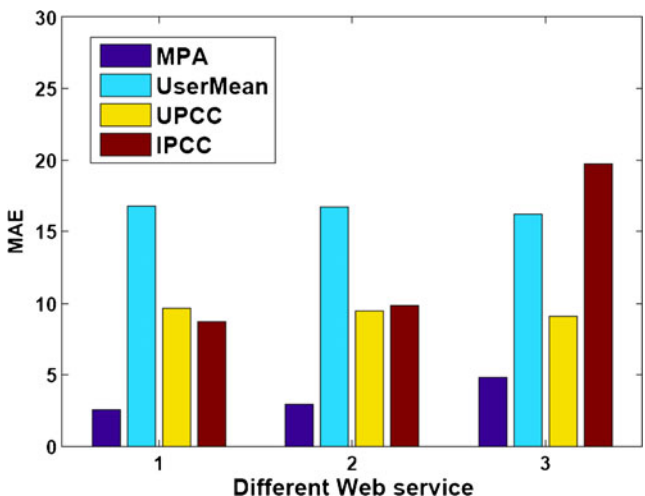
$$C_{ia} = C_{ia} \sum_h W_{ah} \frac{Q_{ih}^*}{(CW)_{ih}} \quad (6)$$

$$C_{ia} = \frac{C_{ia}}{\sum_k C_{ka}} \quad (7)$$

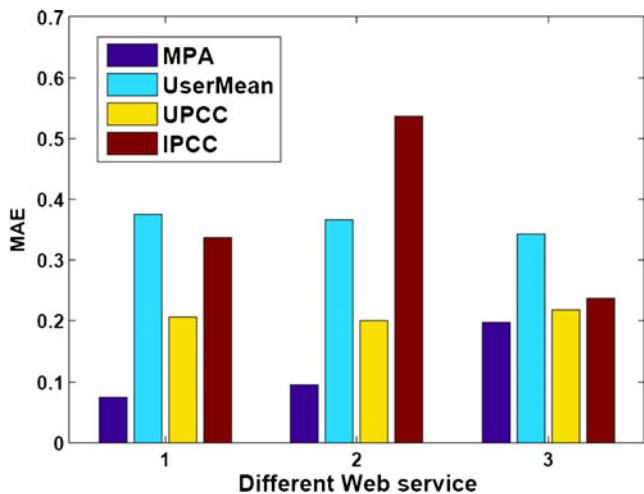
$$W_{ah} = W_{ah} \sum_i C_{ia} \frac{Q_{ih}^*}{(CW)_{ih}} \quad (8)$$

4.3 QoS prediction

In this step, we predict multi-QoS for different users based on the extracted feature of users and their historical QoS experience. Consider we have got the extracted



**Fig. 5** MAE of throughput from different Web services

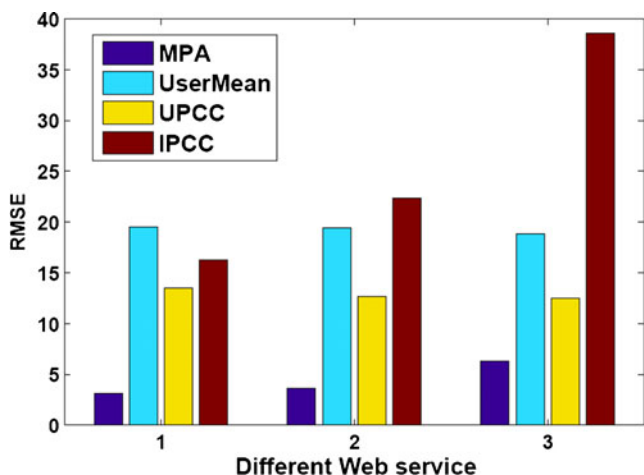


**Fig. 6** MAE of response time from different Web services

feature of users and some personal QoS experience about a target Web service, the key idea of this step is to learn a map function with this information, then to predict the missing personal QoS by this map function, which represents the relation between the user’s feature and multi-QoS values obtained from that Web service.

For example, there are 7 users and a Web service shown in Fig. 3. If we have got the extracted feature of all users and collected the QoS experience observed by the users  $u_1, u_2, u_3, u_4, u_1, u_2, u_3, u_4$ , then we can learn a map function  $f(c)$  based on these data, and predicted the miss QoS experience for the users  $u_5, u_6, u_7$ .

In order to learn the map function, we employ the Multi-output Support Vector Regression (MSVR for short) algorithm (Yu et al. 2006) to predict the missing multi-QoS values at same time. The MSVR is a widely applied regression prediction method, which can control the accuracy of approximate arbitrary nonlinear function. It not only has the global optimum, good generalization ability, but also considers the correlation of outputs. It is suitable for predicting the multi-



**Fig. 7** RMSE of throughput from different Web services

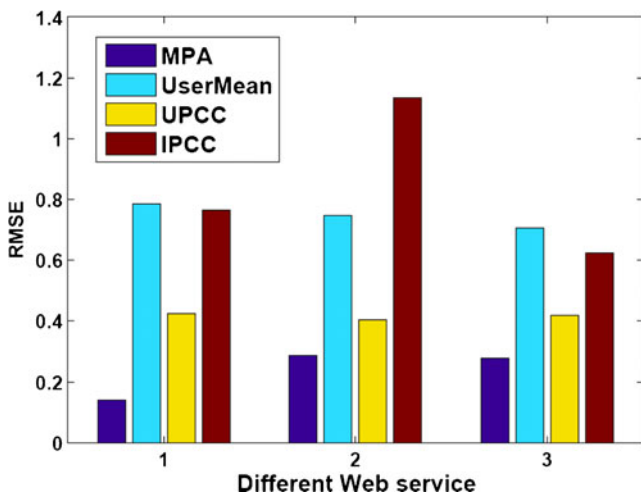


Fig. 8 RMSE of response time from different Web services

QoS together. The whole predicting process can be described as follow:

First, we select the training data according to the invoked experience for a targeted Web service.

Suppose there are  $d$  users  $U = \{u_1, u_2, \dots, u_d\}$  have invoked a target Web services, then the training data can be represented by  $T = \{(c_1, y_2), \dots, (c_d, y_d)\}$ , where  $c_d \in W$  is the obtained feature of user  $u_d$ ,  $y_d$  represents personal QoS experience about this Web service, which observed by  $u_d$ .

Second, we employ MSVR to build a function  $f$ , which represents the relation between the feature of user and Multi-QoS values obtained from a target Web service. Consider  $R^r$  is the feature space of users,  $R^m$  is the QoS space. The function  $f$  can be described by the following:

$$\begin{cases} f : R^r \rightarrow R^m \\ x_k \rightarrow W^T \varphi(x_k) + b \end{cases} \quad (9)$$

Where  $W^T \varphi(x_k) + b$  reflects a map from the feature space of user to the QoS space, which is learning from the training data by MSVR.

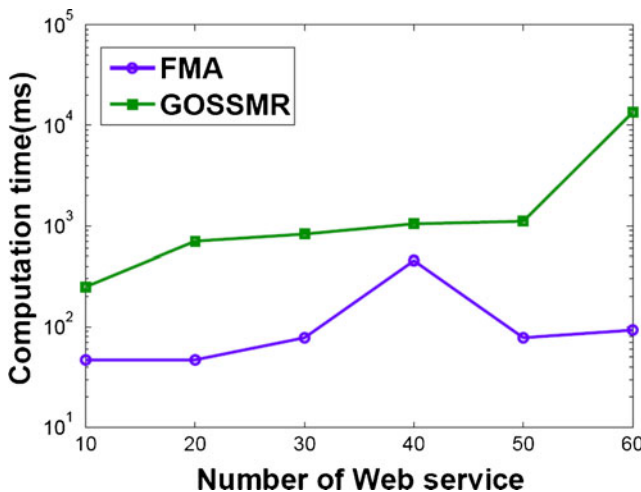


Fig. 9 Computation time with different number of Web services

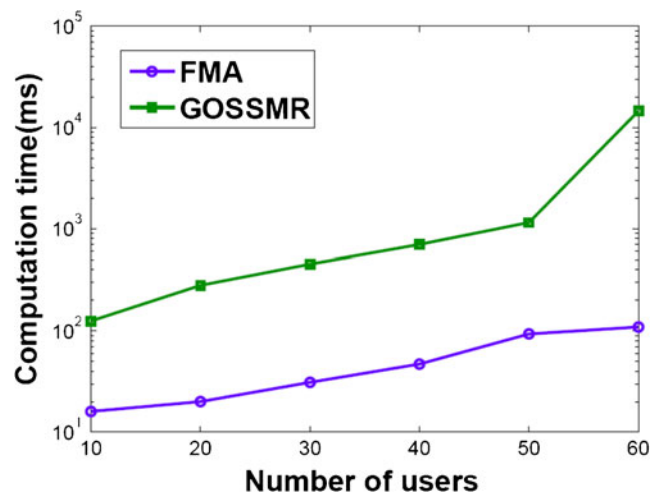


Fig. 10 Computation Time with different number of users

Last, we input the feature of user to predict his personalized missing multi-QoS values for a targeted Web service. let  $c_i$  is the feature of user  $u_i$ , the predicted multi-QoS  $Q_{ij}$  for a targeted Web service  $s_j$  can be calculated by the following:

$$Q_{ij} = W^T \varphi(c_i) + b \quad (10)$$

### 5 Web service selection approach

In this section, we propose a fast match approach to select Web services for multi-user based on the predicted multi-QoS values. To better describe it, we first introduce theorem, and then give the detail of our fast match approach.

**Theorem 1.** For given users' preference weight and their personalized multi-QoS values for candidate Web service, the multi-user Web service selection problem can be transformed to a maximum weight matching problem in bipartite graph.

**Proof.** Consider the maximum weight matching problem, defined by a weighted complete bipartite graph  $G = (U \cup S, U \times S)$ , where edge  $\langle i, j \rangle$  has weight  $w_{i,j}$ . We wish to find a matching  $M$  from  $U$  to  $S$  with maximum weight. We show that the multi-user Web service selection problem can be transformed to a maximum weight matching problem by adding

Table 2 The effect of the parameter  $r$  in term of MAE and D RMSE

Web service	QoS attributes	Performance	$r=2$	$r=4$	$r=6$	$r=8$
Web service 1	Response Time	MAE	0.1657	0.1983	0.1836	0.1699
		RMSE	0.2463	0.2773	0.2612	0.2445
	Throughput	MAE	4.8696	4.8280	5.2849	4.5330
		RMSE	6.3578	6.3002	6.8054	6.0105
Web service 2	Response Time	MAE	0.0935	0.0952	0.0882	0.0906
		RMSE	0.2878	0.2879	0.2869	0.2845
	Throughput	MAE	2.8130	2.9439	3.0751	3.6862
		RMSE	3.4430	3.6000	3.8931	4.4909

virtual Web services according to their processing capacity. Given a bipartite graph  $G=(U \cup S, U \times S)$  to represent the users, services and their utility, if processing capacity of Web service  $s_j \in S$  is  $k$ , we can map it to  $k$  virtual Web service which can only service one user. So our problem can be viewed as a maximum weight matching problem in the new bipartite graph.

According to Theorem 1, the multi-user Web service selection problem can be transformed to a maximum weight matching problem in new bipartite graph. For example, if we select candidate services for three requestors in the left of Fig. 4, the problem can be transform as a maximum weight matching problem in the right of Fig. 4. Where  $vs_{11}$ ,  $vs_{12}$  represent the virtual Web services, and their number equal to the processing capacity of Web service-1. From a matrix point of view, the utility matrix  $UM$  become as a new utility matrix  $NUM$  by the following:

$$NUM = PC \times UM \quad (11)$$

Where  $PC = \{pc_1, \dots, pc_i, \dots, pc_j\}$  is a processing capacity matrix of all the candidate Web service, vector  $pc_i$  represents the processing capacity of Web service  $s_i$ .

---

**Algorithm 1: Fast Match Approach (FMA)**


---

```

Input : (1)  $QoS$  //Predicted Multi-QoS
         (2)  $W$  //Preference Weight of Users
         (3)  $PC$  //Processing Capacity of Web Service
Output: Global Optimal Selection  $X$ 
// step 1. Calculate Utility Matrix  $UM$ 
for each user  $u_i \in U$  do
  | for each Web service  $s_j \in S$  do
  | |  $UM_{ij} = Utility(u_i, s_j)$ 
  | end
end
// step 2. Searching for Local Optimal Selection  $LX$ 
for each user  $u_i \in U$  do
  |  $LX_i = \arg \max Utility(u_i, S)$ 
end
// step 3. Searching for Global Optimal Selection  $X$ 
calculate each Web service's load  $l_j$  by  $LX$ 
if any Web service  $s_j$  satisfy  $l_j > p_j$  then
  | construct a new Utility Matrix  $NUM$  by Eq.11
  |  $X = KM(NUM)$ 
end
 $X = LX$ 
Output  $X$ 

```

---

Algorithm 1 shows the whole procedure of our fast matching approach. The main idea is to respectively calculate the optimal matching Web service for each requester, and check load conditions for each Web service, if number of requests for any Web service is greater than the Web service's processing capacity, we transform our problem to a maximum weight matching problem in a new bipartite graph and employ the Kuhn-Munkres algorithm (KM for short) to select the global optimal solution and avoid requests overloading. For time complexity of FMA, good behavior is  $O(n \times m)$ , and bad behavior is  $O(\max\{n, |P|\}^4)$ , where  $n$  is the number of requestors,  $m$  is the number of Web services,  $|P|$  is sum of all Web service's processing capacity.

## 6 Experiment

To evaluate our methods for the multi-user Web service selection problem, we conduct several experiments on publicly available dataset and the synthetic data. In this section, we are focusing on the prediction quality of our proposed approach and the efficiency of fast match approach.

### 6.1 Comparison results on QoS prediction

To study the prediction performance on multi-QoS, we conduct several experiments to show the prediction quality of our proposed approach on real-world Web service dataset (Zheng et al. 2010). This dataset includes QoS performance of 5,825 openly-accessible real-world Web services from 73 countries. The QoS values are observed by 339 distributed computers located in 30 countries from PlanetLab4. The detail can be describe as Table 1.

In order to evaluate the performance of our approach, We use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as our measurement criteria of prediction accuracy. The metric MAE and RMSE can be calculate by the following:

$$MAE = \frac{\sum_j |q_{ij}^k - \tilde{q}_{ij}^k|}{N} \quad (12)$$

$$RMSE = \sqrt{\frac{\sum_j (q_{ij}^k - \tilde{q}_{ij}^k)^2}{N}} \quad (13)$$

where  $q_{ij}^k$  is the  $k$ -th QoS value of Web service observed by user  $u_i$ ,  $\tilde{q}_{ij}^k$  denotes the  $k$ -th QoS value of Web service would be observed by this user as predicted by a method, and  $N$  is the number of predicted QoS values. Both the MAE and RMSE are negatively-oriented scores, Lower values are better.

In this section, we compare the prediction accuracy of our proposed approach MEQP with single QoS value predict approaches, including the following methods.

- 1) **UserMean**. This method uses the mean QoS value of each user to predict the missing values.
- 2) **UPCC** (User-based collaborative filtering method using Pearson Correlation Coefficient). This method employs PCC to calculate similarities between users and predicts QoS value based on similar users (Shao et al. 2007; Breese et al. 1998).
- 3) **IPCC** (Item-based collaborative filtering method using Pearson Correlation Coefficient). This method employs PCC to calculate similarities between Web services and predicts QoS value based on similar items (item refers to Web service in this paper) (Resnick et al. 1994).



To make our experiment more realistic, we randomly remove 85 % multi-QoS experience of users and only use the remaining 15 % entries to predict the removed multi-QoS experience of different user for a targeted Web service (which is chosen by random). In this part, the above methods are compared with our proposed approaches given the same training and test cases. The parameter settings of our proposed approaches are  $r=4$ ,  $\varepsilon=0.02$ ,  $C=678$ ,  $\sigma=0.5$ .

Figures 5, 6, 7 and 8 show the experimental results of different approaches. From those figures, we can observe that our approach can obtain smaller MAE and RMSE on both two QoS attributes. It shows that our approach has better performance on prediction accuracy on both response-time and throughput for different users. This indicates that our approach suits better for predicting the personalized multi-QoS for different users.

## 6.2 Comparison results on service selection

To study the efficiency of Web service selection, we compare our Web service selection approach, i.e., fast match approach (FMA) with GOSSMR (Kang et al. 2011) in term of the computation time. GOSSMR is a very famous Web service selection approach and it employs a 0–1 integral programming model to solve problem of optimal service selection for multiple requesters). In the first experiment, we will investigate how the execution time of following approach with the number of candidate Web service increases. We varied the number of candidate Web services from 10 to 60 with an increment of 10, but fixed the number of requests is 40. The results of this experiment are presented in Fig. 9. It can be seen that the computation time of FMA is less than the GOSSMR with different number of Web service.

In the second experiment, we will investigate how the execution time of following approach with the number of candidate Web service increases. We vary the number of users from 10 to 60 with an increment of 10, but fixed the number of candidate Web service is 100. The results in Fig. 10 display that the computation time of FMA is less than the GOSSMR with different number of users.

Based on the above two experiments, the conclusions can be drawn: 1) our fast match approach can scalable with change number of users and Web services; 2) compare with GOSSMR algorithms, our approach can spend less time to search a global solution; 3) our approach can meet the Web service selection for multi-users.

## 6.3 Effect of the parameter $r$

Table 2 shows the effect of the  $r$  for our QoS prediction. To show its impact clearly, we vary the value of  $r$  from 2 to 8 with a step value of 2.

We random select two Web services with two QoS attributes (*Response time* and *Throughput*) in the experiment. Table 2 shows the following: 1) for the Response time attribute, the MAE and RMSE of our approach is very low with different  $r$  values. This observation indicates that our QoS prediction approach is very good for predicting the Response time attribute of Web services; 2) Our approach is also suitable to predict the Throughput of Web services.

## 7 Conclusion

In this paper, we propose a new framework for multi-user Web services selection problem. This framework first predicts the missing multi-QoS values according to the historical QoS experience from different users, and then selects the global optimal solution for multi-user by our fast match approach. Comprehensive empirical studies demonstrate the utility of the proposed approach.

**Acknowledgments** The work presented is supported by the NSFC (61202435); NSFC (61272521); Natural Science Foundation of Beijing under Grant No.4132048; Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110005130001; Program for New Century Excellent Talents in University of China under Grant No.NCET-10-0263; Innovative Research Groups of the National Natural Science Foundation under Grant No.61121061; and 863 (2012AA111601).

## References

- Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q. (2003). Quality-driven Web services composition. Proc. the 12th International Conference on the World Wide Web, pp.411–421.
- Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H. (2004). QoS-aware middleware for Web services composition, vol. 30, no.5. IEEE Transaction on Software Engineering, IEEE Computer Society, pp. 311–327.
- Canfora, G., Penta, M.D., Esposito, R., Villani, M.L. (2005). An approach for QoS-aware service composition based on genetic algorithms. Proc. the 2005 conference on Genetic and Evolutionary Computation, pp.1069–1075.
- Alrifai, M., Risse, T. (2009). Combining global optimization with local selection for efficient QoS-aware service composition. Proc. the 18th International Conference on the World Wide Web, pp.881–890.
- Alrifai, M., Skoutas, D., Risse, T. (2010). Selecting skyline services for QoS-based Web service composition. Proc. the 19th International Conference on the World Wide Web, pp.11–20.
- Shahand, S., Turner, S. J., Cai, W., & Khademi, H. (2010). DynaSched: A dynamic Web service scheduling and deployment framework for data-intensive Grid workflows. *Procedia Computer Science*, 1(1), 593–602.
- Dyachuk, D., Deters, R. (2006). Scheduling of composite web services. On the move to meaningful internet systems 2006: OTM 2006 Workshops, pp. 19–20.
- Kang, G., Liu, J., Tang, M., Liu, X., Fletcher, K.K. (2011) Web service selection for resolving conflicting service requests. Proc. IEEE International Conference on Web Service, pp. 387–394.

- Lo, W., Yin, J., Deng, S., Li, Y., Wu, Z. (2012). An extended matrix factorization approach for QoS prediction in service selection. Proc. IEEE Ninth International Conference on Services Computing (SCC), pp. 162–169.
- Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H. (2007). Personalized QoS prediction for Web services via collaborative filtering. Proc. IEEE International Conference on Web Services, pp.439–446.
- Zheng, Z., Ma, H., Lyu, M.R., King, I. (2009). WSRec: A collaborative filtering based web service recommendation system. Proc. IEEE International Conference on Web Services, pp.437–444.
- Jiang, Y., Liu, J., Tang, M., Liu, X.F. (2011). An effective Web service recommendation method based on personalized collaborative filtering. Proc. IEEE International Conference on Web Services, pp.211–218.
- Zhang, L., Zhang, B., Liu, Y., Gao, Y., Zhu, Z. (2010). A Web service QoS prediction approach based on collaborative filtering. Proc. IEEE Asia-Pacific Services Computing Conference, pp.725–731.
- Chen, X., Liu, X., Huang, Z., Sun, H. (2010). RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized Web service recommendation. Proc. IEEE International Conference on WebServices, pp.9–16.
- Tang, M., Jiang, Y., Liu, J., Liu, X. (2012). Location-aware collaborative filtering for QoS-based service recommendation. Proc. IEEE International Conference on Web Services, pp.202–209.
- Ortega, M., Rui, Y., Chakrabarti, K., Mehrotra, S., Huang, T.S. (1997). Supporting similarity queries in MARS. Proc. the ACM Multimedia, pp.403–413.
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791.
- Yu, S. P., Yu, K., & Volker, T. (2006). Multi-output regularized feature projection. *IEEE Transactions on Knowledge and Data Engineering*, 18(12), 1600–1613.
- Zheng, Z., Zhang, Y., Lyu, M. (2010). Distributed QoS evaluation for real-world Web services. Proc. IEEE International Conference on Web Services, pp.83–90.
- Breese, J., Heckerman, D., Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. Proc. the Fourteenth conference on Uncertainty in artificial intelligence, pp.43–52.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., (1994) GroupLens: An open architecture for collaborative filtering of netnews. Proc. the 1994 ACM conference on Computer supported cooperative work, pp.175–186.

**Shanguang Wang** is an assistant professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). He received his Ph.D. degree at BUPT in 2011. His PhD thesis was awarded as outstanding doctoral dissertation by BUPT in 2012. His research interests include Service Computing, Cloud Services, QoS Management. He has served as reviewers for numerous journals, including IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Services Computing, Information

Sciences, The Computer Journal, IET Software, IJSS, JNCA, IEEE IC, International Journal of Web and Grid Services, etc.. URL: [www.sguangwang.com](http://www.sguangwang.com).

**Ching-Hsien Hsu** is presently a visiting scholar in department of computer science at Virginia Tech. He is a professor in department of computer science and information engineering at Chung Hua University, Taiwan; and distinguished chair professor in school of computer and communication engineering at Tianjin University of Technology, China. His research includes high performance computing, cloud computing, parallel and distributed systems, ubiquitous/pervasive computing and intelligence. He has published 200 papers in refereed journals, conference proceedings and book chapters in these areas. He has been involved in more than 100 conferences and workshops as various chairs and more than 200 conferences/workshops as a program committee member. He is the editor-in-chief of international journal of Grid and High Performance Computing, and serving as editorial board for around 20 international journals. He has been acting as an author/co-author or an editor/co-editor of 10 books from Springer, IGI Global, World Scientific and McGraw-Hill. He has also edited a number of international journal special issues as a guest editor, such as IEEE Transactions on Services Computing, Future Generation Computer Systems, Journal of Supercomputing, Concurrency and Computation: Practice and Experience, The Knowledge Engineering Review, Internet Research, Information System Frontiers, etc. He was awarded 5 times annual outstanding research award through 2005 to 2012 and a distinguished award in 2008 for excellence in research from Chung Hua University. He has been serving as executive committee of Taiwan Association of Cloud Computing (TACC) from 2008-2011; executive committee of the IEEE Technical Committee of Scalable Computing (2008-2011). He is member of Phi Tau Phi Scholastic honor society; IEEE senior member; regional director of the Future Technology Research Association (FTRA); and standing director of Taiwan Association of Cloud Computing (TACC).

**Zhongjun Liang** is a Ph.D. Students at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). His research interests include Service computing and Web services.

**Qibo Sun** received his Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 2002. He is currently an associate professor at the Beijing University of Posts and Telecommunication in China. He is a member of the China computer federation. His research interests include services computing, Internet of things, and network security.

**Fangchun Yang** received his Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 1990. He is currently a professor at the Beijing University of Posts and Telecommunication, China. He has published 6 books and more than 80 papers. His research interests include network intelligence, services computing, communications software, soft switching technology, and network security. He is a fellow of the IET.