

# QoS Prediction in Dynamic Web Services with Asymmetric Correlation

Qi Xie<sup>1</sup>(✉), Bing Tang<sup>2</sup>, Zibin Zheng<sup>3</sup>, and Mengtian Cui<sup>1</sup>

<sup>1</sup> College of Computer Science and Technology,  
Southwest University for Nationalities, Chengdu 610041, China  
qi.xie.swun@gmail.com

<sup>2</sup> School of Computer Science and Engineering,  
Hunan University of Science and Technology, Xiangtan 411201, China

<sup>3</sup> School of Data and Computer Science, Sun Yat-sen University,  
Guangzhou 510000, China

**Abstract.** Web services are the mainstream implementation of service-oriented architectures in which both functional and non-functional Quality of Service (QoS) is significantly considered by service users and providers. Although multiple QoS-based models to QoS prediction for Web services via collaborative filtering have been proposed, the accuracy of existing ones is not adequate, since these works rarely have considered the influence of asymmetric correlation among service users on prediction accuracy. This paper combines asymmetric correlation among service users and asymmetric correlation propagation into the deviation computation of different service items in QoS prediction. In this paper, a novel method for QoS prediction in dynamic Web services is proposed, which includes a framework consisting of asymmetric correlation model, asymmetric correlation propagation model and deviation computation algorithm with correlation. To study the QoS prediction performance of our method, a well-known dataset consisting of about 1.97 million real-world QoS records is used in the experiments. The experimental results demonstrate that our method achieves better prediction accuracy than other well-known methods.

**Keywords:** Services computing · QoS prediction · Collaborative filtering · Slope one · Asymmetric correlation

## 1 Introduction

Web services technologies are emerging as a powerful vehicle for organizations that need to integrate their applications within and across organizational boundaries [1–4]. The growing presence and adoption of Web services as the delivery mode in business have caused a wide range of attentions not only from industry but also academia. Due to the large number of Web services that meet the specific functional requirements of service users, its time and resource consuming for service users to choose appropriate services to satisfy their various needs.

To help service users find their desired services, there is urgently need to build effective methods of service selection and recommendation.

To facilitate effective Web service recommendation, the quality of candidate services needs to be assessed from non-functional properties [5]. Quality-of-Service (QoS), which is usually employed for representing the non-functional characteristic of Web services such as response-time, failure-rate and availability, has become an important differentiating point of different Web services [6]. And different Web service QoS properties can be divided into two parts: one is the user-independent QoS properties (e.g., price, popularity, etc.) and the other is the user-dependent QoS properties (e.g., response-time, failure-rate, throughput, etc.) [7]. Specially, the values of user-independent QoS properties are usually provided by service providers and equal for different users, while the values of user-dependent QoS properties can fluctuate widely for different service users since the values are highly related to the Internet conditions and the positions of the service users. Consequently, the same Web service may have very different QoS values for different service users, and methods where QoS values obtained by one user are straight used by other users are inappropriate. Instead, a personalized Web service prediction is needed, in which accurate values of the user-dependent QoS properties should be obtained immediately.

Some previous works [8–11] have employed collaborative filtering (CF) [12–15] to Web service recommendation. These methods can predict the QoS performance of a Web service for a user by employing historical QoS information from other similar service users, who have similar historical QoS experience on the same set of commonly-invoked Web services [5–7].

To improve the prediction accuracy, some hybrid prediction methods are proposed by combining of both memory-based CF method and model-based CF method [16] or user-based CF [17, 18] method and item-based CF method [10, 11, 19, 20]. Work [16] employs clustering model and item-based CF method to enhance the prediction accuracy. Zheng et al. [10, 11] and Tang et al. [19, 20] combine user-based CF method and item-based CF method to balance two different CF methods, then increase the prediction accuracy of the traditional CF methods.

However, there is still one problem of the previous works to be solved, which influences the performance of QoS prediction. That is to say, the current methods fail to recognize the importance of the asymmetric correlation among different service users. Through the experiments implemented on a real-world Web service dataset<sup>1</sup>, which includes 1.97 million Web service invocation results of 339 Web services observed by 5825 service users, we conclude that the asymmetric correlation among service users is the same significant as the similarity of services for QoS prediction.

To address the absence of asymmetric correlation among service users, different from the above methods, we propose a novel method for QoS prediction in dynamic Web services. The proposed method systematically integrates asymmetric correlation among service users and asymmetric correlation propagation

---

<sup>1</sup> <http://www.wsdream.net>.

into the deviation computation of different service items in QoS prediction. In addition, this paper mainly focuses on the asymmetric correlation model for service users, asymmetric correlation propagation model and deviation computation method with asymmetric correlation.

The main contributions of this work are threefold:

- Firstly, the asymmetric correlation for service users is introduced into the QoS prediction in dynamic Web services, which has a significant effect on the prediction accuracy.
- Secondly, asymmetric correlation propagation model is also proposed to take into account of the propagation and attenuation of asymmetric correlation for service users.
- Finally, experiments are conducted to evaluate our method by employing a real-world QoS dataset which includes about 1.97 million invocations for Web services.

The remainder of this paper is organized as follows: Sect. 2 provides an overview on the proposed framework of QoS value prediction. Section 3 introduces the QoS value prediction method with asymmetric correlation in detail. Section 4 presents the experiments of our method. Section 5 concludes the paper with a summary and a description of future work.

## 2 Framework of QoS Value Prediction

To provide the optimal services for service users among functionally equivalent ones, the framework shown in Fig. 1 acts as a platform for service users to obtain optimal Web services with satisfying predicted QoS values by employing asymmetric correlation and asymmetric correlation propagation. Figure 1 shows the framework overview of QoS value prediction, which includes the following procedures.

- (a) Asymmetric correlation (AC) modeling is the first crucial problem to be solved in our framework. Firstly, existing QoS values of Web services are collected and analyzed to build the correlation matrix (CM) for QoS values. And the correlation matrix records the numbers of Web services that have been invoked by each pair of service users. After that, asymmetric correlation matrix (ACM) for service users can be constructed by using correlation matrix, where each entry in the ACM matrix is related to not only the numbers of Web services invoked by each pair of service users but also the numbers of Web services invoked by the others. Finally, according to the asymmetric correlation matrix, asymmetric correlation graph for service users can be built.
- (b) Asymmetric correlation propagation modeling is the second crucial problem to be attacked. In this step, the propagation of asymmetric correlation for service users in asymmetric correlation graph is taken into account adequately. Furthermore, it's very important to obtain the relation flow, which transfers the values of asymmetric correlation for a service user to others strongly connected to it.

- (c) Deviation computation of Web services in QoS prediction framework will be built based on the values of asymmetric correlation propagation and the QoS dataset. Then the results of deviation computation can be used to find the different services for corresponding service users. In this step, traditional deviation computation employing SLOPEONE method [14] is modified and extended with the values of propagation.
- (e) After finding the appropriate different services, the missing QoS values for service users can be predicted by employing the deviation character.

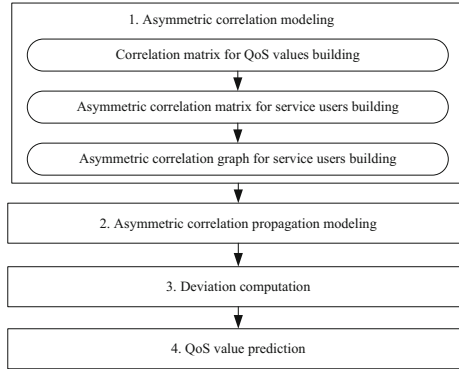


Fig. 1. Framework overview of QoS value prediction

### 3 QoS Value Prediction with Asymmetric Correlation

#### 3.1 Preliminaries

In order to make an accurate prediction, some researchers employ collaborative filtering methods to provide personalized prediction of QoS values for service users [8–11]. Due to the great successes in modeling characteristics of service users and Web services, collaborative filtering techniques have been widely employed in famous commercial systems, such as Amazon, eBay, etc. [10]. In the QoS prediction scenario, CF methods can automatically predict QoS values of unused services for service users by collecting Web service QoS information from other users.

In a typical QoS prediction scenario [10, 11], there is a user-item matrix which includes  $M$  service users and  $N$  Web service items. Every entry in this user-item matrix  $r_{m,n}$  denotes a vector of QoS values (e.g., response-time, failure-rate, etc.), which is obtained by the service user  $m$  on the Web service item  $n$ . And if Web service item  $n$  is not invoked by service user  $m$  previously, then  $r_{m,n} = 0$ .

Item-based CF method can provide dramatically better performance than user-based method [10, 15]. And Slope One method (named as SLOPEONE) is a

typical item-based CF method which works on comparing the intuitive principle of a popular differential between service items rather than similarity between service items [15]. The typical deviation computation of service item  $i$  and  $j$  can be calculated by:

$$Dev(i, j) = \sum_{u \in U(i) \cap U(j)} \frac{r_{u,i} - r_{u,j}}{|U(i) \cap U(j)|}, \quad (1)$$

where every user  $u$  has invoked both service item  $i$  and service item  $j$ , and  $|U(i) \cap U(j)|$  is the number of service users consisting of  $u$ .  $r_{u,i}$  and  $r_{u,j}$  represent the QoS values of service item  $i$  and service item  $j$  observed by user  $u$ , respectively.

In addition, the SLOPEONE method uses different service items to predict the QoS values by employing the following equation:

$$P(r_{u,i}) = \bar{r}_u + \frac{\sum_{j \in R_u} Dev(i, j)}{|R_u|}, \quad (2)$$

where  $P(r_{u,i})$  is a vector of predicted QoS values of entry  $r_{u,i}$  in the user-item matrix using SLOPEONE method.  $\bar{r}_u$  is a vector of average QoS values of different service items invoked by service user  $u$ . Furthermore,  $R_u = \{j | j \in r_u, j \neq i, num(Dev(i, j)) > 0\}$  is the set of all relevant items, in which  $num(Dev(i, j)) > 0$  represents the number of service user  $u$ , who invoked both service item  $i$  and service item  $j$ , should be larger than 0. And  $|R_u|$  is the number of  $R_u$ .  $Dev(i, j)$  can be calculated by Eq. (1).

### 3.2 Asymmetric Correlation Modeling

In this paper, asymmetric correlation is introduced to address the absence of correlations among service users, which have a significant effect on QoS value prediction. Furthermore, modeling of asymmetric correlation includes three steps which are described as follows.

**Correlation Matrix (CM) for QoS Values Building.** A necessary first step before correlation matrix (CM) building is to give the definition of CM, which is based on the assumption that the more items that have been rated by both user  $u_p$  and user  $u_q$ , the closer the users are [21]. Based on work [21], in which correlation rating matrix records the number of items that have been rated by each pair of users in MovieLens scenario, the CM in QoS value prediction for Web services can be defined by:

$$S(u_p, u_q) = \begin{cases} \{i_n : (r_{u_p, i_n} \neq \emptyset) \wedge (r_{u_q, i_n} \neq \emptyset)\} & (u_p \neq u_q) \\ \emptyset & (u_p = u_q) \end{cases} \quad n = 1, \dots, N, \quad (3)$$

$$CM_{u_p, u_q} = |S(u_p, u_q)|, \quad p = 1, \dots, M, q = 1, \dots, M, \quad (4)$$

where  $S(u_p, u_q)$  is a set of Web services observed by both service user  $u_p$  and service user  $u_q$ , and  $|S(u_p, u_q)|$  is the number of  $S(u_p, u_q)$ .  $r_{u_p, i_n}$  and  $r_{u_q, i_n}$  are

the sets of QoS values of Web service in observed by service user  $u_p$  and service user  $u_q$ , respectively. Furthermore,  $M$  is the number of service users and  $N$  is the number of Web service items in user-item matrix.

Equations (3) and (4) show that CM is an  $M$ -by- $M$  symmetric matrix. Taking into account the degrees of service user correlations, the values of correlations for service users, which invoked different number of Web services, should be different.

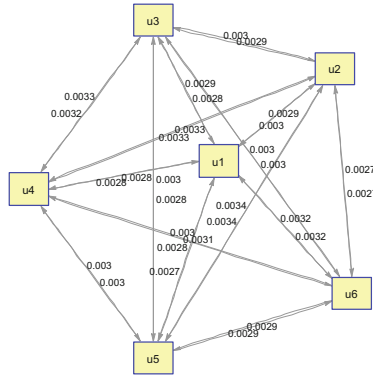


Fig. 2. Asymmetric correlation graph for service users

**Asymmetric Correlation Matrix (ACM) for Service Users Building.**

According to [21], the correlation of service users is related to not only the numbers of Web services invoked by each pair of users but also the total numbers of Web services invoked by other users. Therefore, CM matrix can be improved and transformed to asymmetric correlation matrix (ACM). So, the ACM matrix for service users in our method can be calculated by:

$$ACM_{u_p, u_q} = \begin{cases} \frac{|S(u_p, u_q)|}{\sum_{l \in M, l \neq p} |S(u_p, u_l)|}, & p \neq q, \\ 0, & p = q, \end{cases} \quad p = 1, \dots, M, q = 1, \dots, M, \quad (5)$$

where  $u_p$  and  $u_q$  are the service users in user-item matrix, and  $u_l$  are the service users who are not equal to  $u_p$ .

From Eqs. (4) and (5), it can be observed that ACM matrix is different from CM matrix and ACM matrix is an asymmetric matrix.

**Asymmetric Correlation Graph for Service Users Building.**

According to ACM matrix, the correlations of service users can be built as a directed graph as shown in Fig. 2, in which a node  $u_p$  expresses a service user and a directed edge  $e_{pq}$  from node  $u_p$  to node  $u_q$  expresses an asymmetric correlation from  $u_p$  to  $u_q$ . Therefore, the asymmetric correlation graph can be employed in asymmetric correlation propagation model to obtain the relation flow.

### 3.3 Asymmetric Correlation Propagation Modeling

In the step, we take into account not only asymmetric correlation among service users but also propagation and attenuation of asymmetric correlation. Observing from Fig. 2, we can intuitively see that the correlation among service users can spread throughout the graph and some users may obtain high correlation than others.

PageRank is an excellent way to prioritize the results of Web keyword searches, which counts the approximation of a pages importance or quality by not counting links from all pages equally, and by normalizing by the number of links on a page. And the PageRank for node  $n$  is defined as follows [22]:

$$PR(n) = a \cdot \sum_{q:(q,n) \in \varepsilon} \frac{PR(q)}{w_q} + (1 - a) \cdot \frac{1}{|v|}, \tag{6}$$

where  $PR(n)$  is the importance score for every node  $n \in v$  according to the graph connectivity.  $|v|$  is the number of nodes in graph  $G = (v, \varepsilon)$ , in which  $\varepsilon$  is the set of directed connected edges among nodes.  $a$  is a decay factor which is usually set to 0.85. And  $w_q$  is the out-degree of node  $q$ .

Similar to PageRank [23], an asymmetric correlation propagation model is proposed which is based on the intuition that the correlation of service users can be effectively computed by employing a similar method as PageRank.

So, the following equation can be defined to compute the propagation of correlation among service users by modifying Eq. (6). And then, different correlation ranks for service users can be obtained.

$$PR(u_q) = a \cdot \sum_{p \in N(u_q)} PR(u_p) ACM(u_p, u_q) + (1 - a) \cdot \frac{1}{M}, \tag{7}$$

where  $PR(u_q)$  is a vector of correlation ranks for service user  $u_q$ .  $N(u_q)$  is a set of service users that connect to service user  $u_q$ .  $ACM$  is the asymmetric correlation matrix calculated by Eq. (5), and  $M$  is the number of service users in user-item matrix.

In vector form, Eq. (7) can be equivalently defined as:

$$\begin{bmatrix} PR(u_1) \\ \vdots \\ PR(u_M) \end{bmatrix} = a \cdot ACM^T \cdot \begin{bmatrix} PR(u_1) \\ \vdots \\ PR(u_M) \end{bmatrix} + \begin{bmatrix} (1 - a)/M \\ \vdots \\ (1 - a)/M \end{bmatrix}, \tag{8}$$

where  $ACM^T$  is the transposed matrix of the  $ACM$ .

The correlation ranks for service users can be determined by calculate the eigenvector with eigenvalue 1 or by repeating the process until all correlation ranks become stable [23].

### 3.4 Deviation Computation and QoS Value Prediction

In traditional CF, Gao et al. [21] use different weights for users to represent the relationships. Different from this work which focuses on movie recommendation by using MovieLens dataset, our method employs about 1.7 million real-world QoS records (eg., response-time, throughput.) for Web services observed by service users to predict the missing QoS values for service users, and modifies the propagation model to suit the dynamic QoS prediction Scenario.

In this step, the asymmetric correlation and propagation among service users are combined into deviation computation of Web services for QoS value prediction in service-oriented architecture.

And then, the following equations are employed to compute the deviation and predict the QoS values by using enhanced SLOPEONE method respectively.

$$Dev'(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - r_{u,j}) * PR(u)}{\sum_{u \in U(i) \cap U(j)} PR(u)}, \quad (9)$$

$$P'(r_{u,i}) = \bar{r}_u + \frac{\sum_{j \in R_u} Dev'(i, j)}{|R_u|}, \quad (10)$$

where  $PR(u)$  is a set of correlation ranks for service user  $u$  which can be calculated by Eq. (8).  $Dev'(i, j)$  is the new deviation value that is employed to predict the QoS values in Eq. (10).  $P'(r_{u,i})$  is a new vector of predicted QoS of  $r_{u,i}$  in user-item matrix, in which the asymmetric correlation and propagation have been took into account adequately.

## 4 Experiments

### 4.1 Dataset Description

To study the QoS value prediction performance of our method, we use a well-known dataset provided by Zheng et al. [7], which includes about 1.97 million invocations for Web services which are obtained from 339 service users on 5825 publicly available Web services. In addition, a  $339 \times 5825$  user-service item matrix can be obtained by analyzing and processing the dataset, and each entry in the user-service item matrix represents a QoS value (i.e., response-time.).

In order to evaluate the performance of our method, we randomly choose different number of services from 5825 available Web services named as  $NS$  ranging from 1000 to 4000 with a step value of 1000. Then, a  $339 \times NS$  user-service item matrix is constructed. After that, all entries in the  $339 \times NS$  user-service item matrix are divided into two parts randomly, one as the training set and the other as the test set. A parameter *percent* ( $0 < percent \leq 1$ ) is employed to present the percentage of training set in whole user-service item matrix, and  $(1 - percent)$  shows the percentage of test data. Furthermore, to simulate the real situation, entries in training set are removed randomly with *density* ( $0 < density \leq 1$ ).



## 4.2 Metrics

Mean Absolute Error (MAE) is employed to measure the prediction accuracy of our method in comparison with other well-known CF methods, such as the User-based method using Pearson Correlation Coefficient (named as UPCC), the Item-based method using Pearson Correlation Coefficient (named as IPCC), and WSRec method [10]. The MAE is defined as:

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \hat{r}_{u,i}|}{N}, \quad (11)$$

where  $r_{u,i}$  is the real QoS value of service item  $i$  invoked by user  $u$ ,  $\hat{r}_{u,i}$  denotes the predicted QoS value for service item  $i$ , and  $N$  is the number of predicted values. In addition, the Normalized Mean Absolute Error (NMAE) is also used to measure the prediction quality as follows.

$$NMAE = \frac{MAE}{\sum_{u,i} r_{u,i}/N}. \quad (12)$$

The lower the MAE, NMAE are, and the more effective the prediction methods are.

In order to study the prediction performance more completely, other metrics are introduced in this paper, which include Precision, Recall, and F-measure [24, 25].

Precision is defined as the ratio of number of truly “high” ratings among these that were predicted to be “high” by a prediction method in this way [25]:

$$Precision = \frac{|R_{u,i} \cap T_{u,i}|}{|R_{u,i}|}, \quad (13)$$

where  $R_{u,i}$  is a vector of predicted QoS values of  $r_{u,i}$  with “high” QoS values and  $T_{u,i}$  is a vector of true QoS values observed by users with “high” QoS values.

Recall is defined as the ratio of number of correctly predicted “high” ratings among all the ratings obtained to be “high” which is calculated by [25]:

$$Recall = \frac{|R_{u,i} \cap T_{u,i}|}{|T_{u,i}|}. \quad (14)$$

Larger Precision and Recall correspond to the better performances respectively. However, the values of Precision and Recall are usually conflictual. That is to say, the Precision can be improved by increasing the number of “high” QoS values with high efficiency, while the Recall is increasing correspondingly with high cost [26]. Due to the inconsistent performances between precision and recall, F-measure metric can be used to balance of above two competitive metrics [27].

$$F_\beta = \frac{(1 + \beta^2) \cdot (Precision \cdot Recall)}{\beta^2 \cdot Precision + Recall}, \quad (15)$$

where the parameter  $\beta$  is a regular certain value which is set to 0.5, 1, and 2 respectively.

### 4.3 Comparison

Since the user-service item matrix is usually very sparse in reality, the QoS value prediction methods always encounter the sparsity problem. Then, the parameter *density* is set to 0.1 to keep the consistency with the real-world data distribution. Table 1 shows the MAE and NMAE values of five different prediction methods on response-time with *percent* = 0.8, *density* = 0.1 and *NS* from 1000 to 4000 with a step value of 1000. And the best performance result is shown with parameter  $\lambda = 0.2$  in WSRec method [10]. The UPCC method and IPCC method are the famous CF methods which are widely studied. In addition, SLOPEONE method is defined in Eq. (2) and our method is named SLOPEONE-R which takes into account the asymmetric correlation and propagation adequately. Figure 3 includes three versions of Table 1. One version compares the MAE and NMAE values with different *NS* range from 1000 to 4000, the second version uses the bar graph to show the MAE and NMAE values with different methods while the last version employs curve graph to illustrate the five methods with different MAE and NMAE values under different experimental sets.

**Table 1.** MAE and NMAE comparison with basic methods

NS	Metric	Methods				
		UPCC	IPCC	WSRec	SLOPEONE	SLOPEONE-R
1000	MAE	0.8217	0.7047	0.6986	0.6560	0.6506
	NMAE	0.9172	0.7889	0.7820	0.7323	0.7217
2000	MAE	0.8268	0.6970	0.6934	0.6479	0.6478
	NMAE	0.9242	0.7833	0.7792	0.7242	0.7241
3000	MAE	0.8260	0.6952	0.6926	0.6476	0.6474
	NMAE	0.9232	0.7789	0.7760	0.7237	0.7236
4000	MAE	0.8422	0.7094	0.7049	0.6518	0.6517
	NMAE	0.9205	0.7797	0.7747	0.7123	0.7122

The experimental results for the five methods using the same error measure and over the same data set shown in Table 1 and Fig. 3 indicate that:

- Our SLOPEONE-R method improves the prediction performance, and outperforms other four methods for two different metrics-MAE and NMAE under all experimental settings.
- The SLOPEONE-based methods outperform the Pearson Correlation Coefficient (PCC)-based methods, which indicates that deviation computation is more important than similarity computation in QoS prediction.
- The SLOPEONE-R method obtains smaller MAE and NMAE values than SLOPEONE method. This observation shows that asymmetric correlation and propagation among service users have an important effect on QoS prediction accuracy.

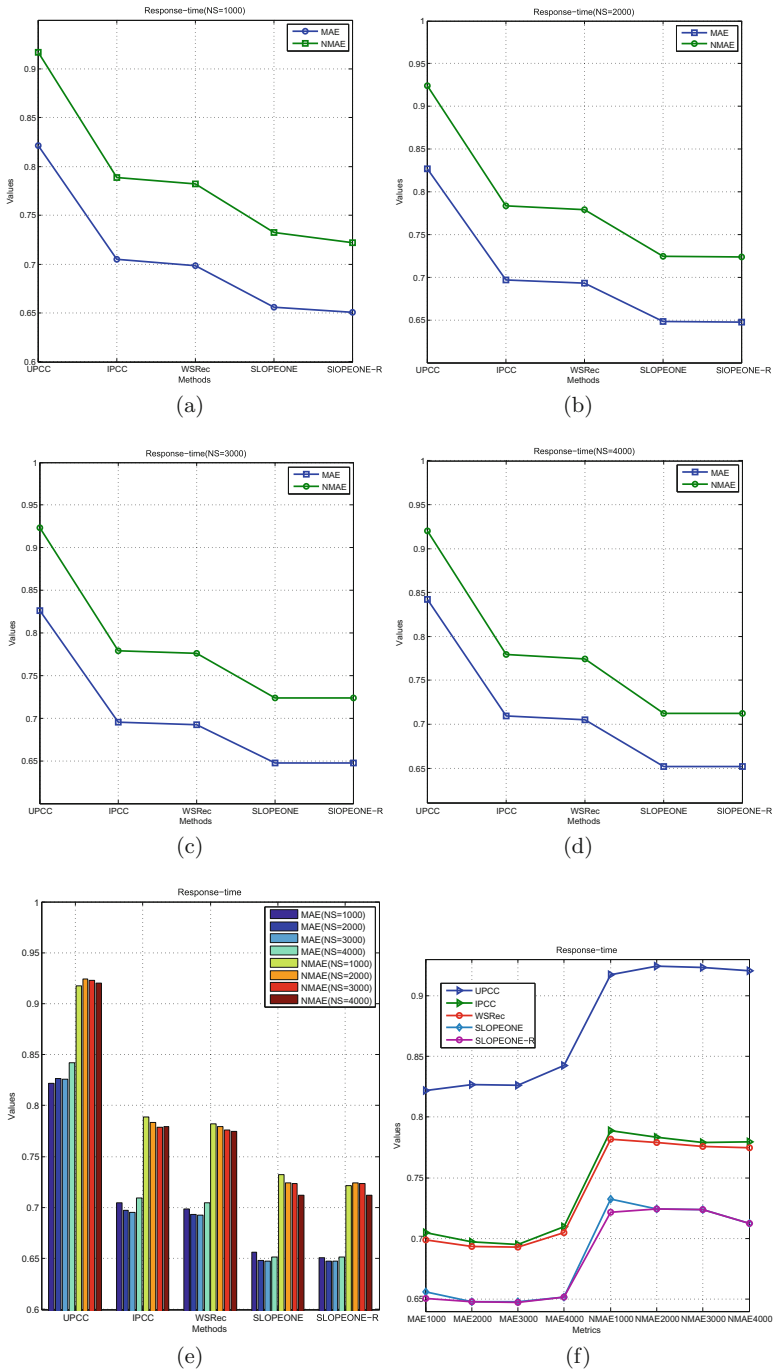


Fig. 3. MAE and NMAE values comparison (Color figure online)

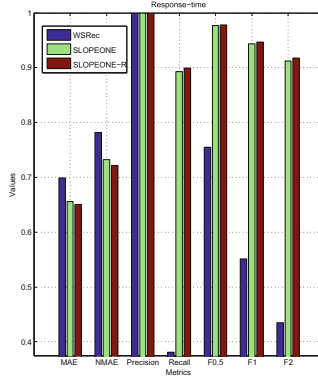
- Fig. 3(a)–(d) show that the MAE and NMAE values have the similar downward trend of five different methods with different  $NS$ , which the MAE is smaller than NMAE consistently and the UPCC method obtains the largest MAE and NMAE values while SLOPEONE-R method always gains the smallest MAE and NMAE values. This illustrates the better prediction accuracy of our method.
- Both Fig. 3(e) and (f) illustrate that the MAE and NMAE values of five methods generally experience a downward trend with the rise of the Web service number from 1000 to 3000, indicating that more QoS values can improve the prediction accuracy. By contrast, the MAE values of five methods experience an upward trend when the Web service number is set to 4000, since the prediction accuracy is influenced not only by the Web service number, but also by the nature of datasets. Furthermore, too many QoS values may suffer from data over fitting.
- It can be seen from the Table 1 and Fig. 3 that the absolute values of MAE and NMAE of our method are improved with slight enhancement. The reason is that the number of service users in user-service item matrix is 339 and the mean rank for every user is about 0.002950 with a small value. Though with slight enhancement of MAE and NMAE values, our method can obtain better performance with other metrics which will be discussed in following paragraph.

As shown in Table 2, we compare our SLOPEONE-R method with WSRec method and SLOPEONE method by employing three metrics, such as Precision, Recall and F-measure with the parameter  $\beta = 0.5$ ,  $\beta = 1$  and  $\beta = 2$  respectively. In the experiment, we set  $NS = 1000$ ,  $percent = 0.8$ , and  $density = 0.1$ . In services computing, the “high” QoS values in Eqs. (13) and (14) are more than 0 in the experiment. In addition, the ratios of performance improved by using SLOPEONE-R method versus WSRec method and SLOPEONE method are also presented in Table 2. Furthermore, Fig. 4 illustrates the different levels of performance with seven metrics which are composed of Tables 1 and 2.

**Table 2.** Performance comparison of three methods

Metric	Methods					
	WSRec	SLOPEONE	SLOPEONE-R	Impro. vs. WSRec	Impro. vs. SLOPEONE	
Precision	1	1	1	-	-	
Recall	0.3805	0.8930	0.8996	136.43 %	0.74 %	
F0.5	0.7543	0.9766	0.9782	29.68 %	0.16 %	
F1	0.5512	0.9435	0.9471	71.83 %	0.38 %	
F2	0.4343	0.9125	0.9180	111.37 %	0.60 %	

Observing from Table 2 and Fig. 4, we draw the conclusion that:



**Fig. 4.** Performance comparison with seven metrics (Color figure online)

- The SLOPEONE-R method outperforms other competitive methods steadily in all metrics except Precision, indicating that by using asymmetric correlation and propagation among service users, the prediction accuracy can be enhanced noticeably.
- It can be seen from the Table 2 that the prediction accuracy of SLOPEONE-R method obtains significant enhancement on average by 87.33 % versus WSRec, and by 0.47 % versus SLOPEONE for four different metrics (e.g., Recall, F0.5, F1, F2.), indicating better prediction accuracy of our method.
- In Table 2, the Precision values of three different methods are equal, since  $R_{u,i}$  is the subset of  $T_{u,i}$  in QoS prediction and the true and effective QoS values in  $T_{u,i}$  are all larger 0, then the values of Precision for different methods are equal to 1.
- In Fig. 4, the same as the experimental results given in Table 1 and Fig. 4, the experimental results of three different methods are following the same trends, which SLOPEONE-based methods outperforms PCC-based methods and SLOPEONE-R method exceeds SLOPEONE method with high values of metrics-Recall, F0.5, F1 and F2.

## 5 Conclusion and Future Directions

In this paper, we leverage the correlations of service users, more specifically asymmetric correlation and asymmetric correlation propagation among service users, to discover relations among all service users involved in QoS prediction. Furthermore, our method combines asymmetric correlation and asymmetric correlation propagation into the deviation computation of different service items in QoS prediction, by using improved PageRank method and extended SLOPEONE method to improve the prediction accuracy. In addition, experiments are conducted in real-world dataset indicating the effectiveness and feasibility of our method. Future direction includes integrating more QoS properties in our

method to predict QoS values. Moreover, experiments of proposed method will be enlarged under more different situations.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China under Grant No. 61502401 and No. 61379019, Guangdong Natural Science Foundation (Project No. 2014A030313151).

## References

1. Zheng, Z., Michael, R.L.: Selecting an optimal fault tolerance strategy for reliable service-oriented systems with local and global constraints. *J. IEEE Trans. Comput.* **64**, 219–232 (2015)
2. Ikbek, G., Nawal, G., Tarak, C., Said, T., Mohamed, J.: Pruning based service selection approach under QoS and temporal constraints. In: 2014 IEEE International Conference on Web Services (ICWS), pp. 9–16. IEEE Computer Society, Washington, DC (2014)
3. Mustapha, A., Mohamed, Q., Zahi, J.: Leveraging formal concept analysis with topic correlation for service clustering and discovery. In: 2014 IEEE International Conference on Web Services (ICWS), pp. 153–160. IEEE Computer Society, Washington, DC (2014)
4. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW, pp. 411–421, (2003)
5. He, P., Zhu, J., Zheng, Z., Xu, J., Lyu, M.R.: Location-based hierarchical matrix factorization for web service recommendation. In: 2014 IEEE International Conference on Web Services, pp. 297–304. IEEE (2014)
6. Menasce, D.: QoS issues in web services. *J. IEEE Internet Comput.* **6**, 72–75 (2002)
7. Zheng, Z., Zhang, Y., Michael R, L.: Distributed QoS evaluation for real-world web services. In: 8th International Conference on Web Services (ICWS 2010), pp. 83–90. IEEE Computer Society, Washington, DC (2010)
8. Rong, W., Liu, K., Liang, L.: Personalized web service ranking via user group combining association rule. In: International Conference on Web Services, pp. 445–452. IEEE Computer Society, Washington, DC (2009)
9. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS prediction for web services via collaborative filtering. In: International Conference on Web Services, pp. 439–446. IEEE Computer Society, Washington, DC (2007)
10. Zheng, Z., Ma, H., Michael, R.L., King, I.J.: QoS-aware web service recommendation by collaborative filtering. *IEEE Trans. Serv. Comput.* **4**, 140–152 (2011)
11. Zheng, Z., Ma, H., Michael R.L., King, I.: WSRec: a collaborative filtering based web service recommender system. In: 7th IEEE International Conference on Web Services, pp. 437–444. IEEE Computer Society, Washington, DC (2009)
12. Burke, R.: Hybrid recommender systems: survey and experiments. *J. User Model. User-Adap. Interact.* **12**, 331–370 (2002)
13. Hofmann, T.: Latent semantic models for collaborative filtering. *J. Acn Trans. Inf. Syst.* **22**, 89–115 (2004)
14. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: SIAM Data Mining (SDM 2005), pp. 471–480 (2005)
15. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: 10th International Conference on World Wide Web, pp. 285–295 (2001)

16. Chen, X., Zheng, Z., Liu, X., Huang, Z., Sun, H.: Personalized QoS-aware web service recommendation and visualization. *J. IEEE Trans. Serv. Comput.* **6**, 35–47 (2013)
17. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: *Computer Supported Cooperative Work*, pp. 175–186 (1994)
18. Breese, J. S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: *Uncertainty in Artificial Intelligence*, pp. 43–52 (1998)
19. Jiang, Y., Liu, J., Tang, M., Liu, X.: An effective web service recommendation method based on personalized collaborative filtering. In: *2011 IEEE International Conference on Web Services*, pp. 211–218. IEEE Computer Society (2011)
20. Tang, M., Jiang, Y., Liu, J., Liu, X.: Location-aware collaborative filtering for qos-based service recommendation. In: *19th International Conference on Web Services (ICWS)*, pp. 202–209. IEEE (2012)
21. Gao, M., Wu, Z., Jiang, F.: UserRank for item-based collaborative filtering recommendation. *J. Inf. Process. Lett.* **111**, 444–446 (2011)
22. Gori, M., Pucci, A., Roma, V., Siena, I.: ItemRank: a random-walk based scoring algorithm for recommender engines. In: *20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 778–781 (2007)
23. Zheng, Z., Michael R.L.: Component recommendation for cloud applications. In: *2nd International Workshop on Recommendation Systems for Software Engineering*, pp. 323–345 (2010)
24. Esmaili, K.S., Neshati, M., Jamali, M., Abolhassani, H., Habibi, J.: Comparing performance of recommendation techniques in the blogosphere. In: *ECAI 2006 Workshop on Recommender Systems*, pp. 40–44 (2006)
25. Gao, M., Fu, Y., Chen, Y., Jiang, F.: User-weight model for item-based recommendation systems. *J. Softw.* **7**, 2133–2140 (2012)
26. Liu, R., Jia, C., Zhou, T.: Personal recommendation via modified collaborative filtering. *J. Phys. A Stat. Mech. Appl.* **388**, 462–468 (2009)
27. Gong, M., Xu, Z., Xu, L.: Recommending web service based on user relationships and preferences. In: *20th International Conference on Web Services (ICWS)*, pp. 380–386. IEEE Computer Society, Washington, DC (2013)