QoS Prediction for Web Services Based on Similarity-Aware Slope One Collaborative Filtering

Chengying Mao and Jifu Chen
School of Software and Communication Engineering,
Jiangxi University of Finance and Economics, 330013 Nanchang, China
E-mail: maochy@yeah.net

Keywords: Web services, QoS prediction, Slope One, similarity, collaborative filtering

Received: December 15, 2012

Web services have become the primary source for constructing software system over Internet. The quality of whole system greatly dependents on the QoS of single Web service, so QoS information is an important indicator for service selection. In reality, QoSs of some Web services may be unavailable for users. How to predicate the missing QoS value of Web service through fully using the existing information is a difficult problem. This paper attempts to settle this difficulty through combining Pearson similarity and Slope One method together for QoS prediction. In the paper, we adopt the Pearson similarity between two services as the weight of their deviation. Meanwhile, some strategies like weight adjustment and SPC-based smoothing are also utilized for reducing prediction error. In order to evaluate the validity of our algorithm (i.e., similarity-aware Slope One algorithm, SASO), comparative experiments are performed on the real-world data set. The results show that SASO algorithm exhibits better prediction precision than both basic Slope One and the well-known WsRec algorithm in most cases. Meanwhile, our approach has the strong ability of reducing the impact of noise data.

Povzetek: Članek poskuša razrešiti problem ocenjevanja kakovosti storitve s kombiniranjem Parsonove podobnosti in metode Slope One.

1 Introduction

In recent years, the pattern of service-oriented computing (SOC) has been widely accepted to build large-scale system over Internet [1]. In this new style of software development paradigm, software is no longer built via the traditional process, but in the way of service unit reuse. Accordingly, some new problems such as service discovery, selection and composition are emerging, and play a great impact on the quality of service-based system.

In general, service unit is self-describing component to complete a specific task. Quality-of-Service (QoS) is an important way to describe non-functional characteristics of Web services. When several functionally-equivalent Web services exist in the network, QoS is viewed as a critical issue for picking out the appropriate service from equivalent service set. Web service QoS usually includes a number of properties, such as response time, throughput, failure probability, availability, price, popularity, and so on [2]. Due to different network environments, service users will have different QoS metrics for the same Web service. Therefore, each service user has to understand QoSs of all services to be invoked at his/her end.

In order to construct the software meeting the actual requirements, it needs to make the existing service units work together in accordance with the pre-defined business logic, that is the so-called Web service composition (WSC). During service selection, the quality of each service unit should be carefully considered so as to ensure the trustworthiness of WSC. However, service invoker may be lack of adequate historical information for some specific Web services. He/She has to estimate the QoS value of a given Web service before determining to introduce it into WSC, i.e., QoS prediction for Web services. Since the service user has not even invoked the service in past, the estimation for such service's QoS has to get help from other similar users or self's invocation records on other Web services.

The similar work firstly emerged in the field of E-commerce, vendors used consumer's historical purchase records and the similarity between costumers to recommend products [3]. In contrast, the prediction of Web service's QoS is much harder than product recommendation. Web service is merely an encapsulated and distributed Web API over network. Therefore, for service users, the information related with service execution are hardly collected. In order to improve the prediction precision, the limited available Web services invocation records should be fully utilized. As far as we known, study in [4] is the first work of predicting Web service's QoS through collaborative filtering (CF). Shao *et al.*'s work mainly considered the similarity among user's experiences on Web services, and proposed a service users' similarity-based prediction method, in which the similarity is measured by Pearson correlation coefficient. Subsequently, Zheng *et al.* [5] presented a

more comprehensive method for QoS prediction, in which they combined the traditional user-based and item-based collaborative filtering methods together through confidence weights. Recently, some improved methods based on personalized context [6, 7] or hierarchical and side information [8] are also proposed.

It is important to note that, most above mentioned methods are in accordance with Pearson-based similarity. Although this kind of similarity can provide good prediction effect, it not only cost much computation time but also lose performance for the very sparse data set. Besides the similarity-based collaborative filtering, Slope One [9] has been validated as an effective prediction method due to its simpleness and high performance. In the paper, we presented a hybrid QoS prediction method through introducing Pearson-based similarity into Slope One method. The experimental results revealed that our hybrid method (named *similarity-aware Slope One*, SASO) could outperform the basic Slope One and Pearson-based collaborative filtering methods in term of prediction precision.

The main contributions of this paper can be addressed as follows.

(1) A prediction algorithm of Slope One co-operated with Pearson similarity measurement has been proposed for providing QoS information for Web service user.

(2) Some strategies like weight adjustment and SPC-based smoothing are presented for improving the prediction precision.

(3) The detailed performance analysis on real-world data set is performed to verify the effectiveness of our method. Moreover, the two-stage filling strategy is also validated through experimental analysis.

The structure of the paper is as follows. In the next section, we state the QoS prediction problem for Web services, and introduce two typical collaborative filtering algorithms. In section 3, the overall QoS prediction framework is firstly addressed, and then the similarity-aware Slope One algorithm is described in details. The performance comparison and analysis are discussed in section 4. Section 5 gives some existing researches that are closely related with our prediction approach. Finally, section 6 concludes the paper.

2 Background

2.1 QoS prediction for Web services

When Web service users prepare to adopt some service units to construct an enterprise-level application, in general, they have to replace each abstract service in service orchestration plan with a concrete service. For each abstract service, perhaps quite a few service implementations will meet the requirement of its function. Therefore, the rational way is to pick out a service with high QoS from the candidate set. However, for a specific service user, the QoS values of some Web services may be not available. As a consequence, it is necessary to estimate the QoSs of such services according to the limited existing information, that is so-called QoS prediction problem.

Motivating Example. Here, we provide a simple illustration to address the QoS prediction for Web services. As shown in Table 1, there are response time (i.e. RT) records of three Web services w.r.t five users. The element $r_{i,j}$ means the RT value of user i for service j, and "NA" represents the corresponding value not available at present. Assume user $u3$ has some interests on the third service, since there is no ready record in the table, he has to predicate the issue $r_{3,3}$ according to his own and others' service invocation records.

User	Response time (second)		
	*service*1	*service*2	*service*3
*u*1	0.4	1.6	NA
*u*2	0.9	NA	1.9
*u*3	2.8	3.5	**??**
*u*4	NA	3.0	4.0
*u*5	0.8	NA	0.9

Table 1: An motivated example for illustrating QoS prediction problem.

How to estimate the missing value? Besides $u3$'s existing records on other two services (i.e. $r_{3,1}$ and $r_{3,2}$), the available service invocation records of other four users also should be taken into consideration. With regard to prediction techniques, experiences tell us that *collaborative filtering* (CF) techniques can be viewed as a good choice.

2.2 Review on collaborative filtering

In general, collaborative filtering is a technique of suggesting particularly interesting items or patterns based on past evaluations of a large group of users. The fundamental assumption of CF is that if users have similar tastes on some items, and hence they will rate or act on other items similarly. At present, CF techniques can be classified into three categories [10, 11]: (1) memory-based methods, (2) model-based methods, and (3) hybrid methods. Memory-based CF utilizes the user rating data to calculate the similarity or weight between users or items, and then make predictions according to those similarity values. This type of CF is the earlier mechanism and used in many commercial systems such as Amazon, Barnes and Noble. According to the background and feature of QoS prediction problem, memory-based CF is treated as the main research issue in the paper. Especially, two well-known methods, i.e., Pearson correlation CF and Slope One approach, are taken into consideration.

2.2.1 Pearson correlation-based method

In a typical CF scenario, there is a list of m users $\{u_1, u_2, \cdots, u_m\}$ and a list of n items $\{i_1, i_2, \cdots, i_n\}$, and

each user u_i has a list of items (i.e., Iu_i), which the user has rated, or about which their preferences have been inferred through their behaviors [10]. Generally speaking, the basic procedure of CF-based recommendation or prediction can be summarized as the following two steps:

(1) Look for users sharing the similar interests or rating patterns with a given user (called active user).

(2) Use the information from those like-minded users found in step (1) to calculate a prediction for the active user.

Here, we mainly address the case from the perspective of users, but the above process is also suitable for item-oriented analysis. It is not hard to find that, how to find the similar users (or items) for a specific user (or item) is a critical task in the whole process of CF. In practice, the common interests or patterns are expressed via the correlation between users (or items).

At present, *Pearson correlation coefficient* has been introduced for computing similarity between users or items according to the user-item data like in Table 1, which is usually called *user-item matrix*. For two given users a and u, their similarity can be computed as follows.

$$Sim(a,u) = \frac{\sum_{i\in I}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i\in I}(r_{a,i}-\bar{r}_a)^2}\sqrt{\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2}} \quad (1)$$

where $I = I_a \bigcap I_u$ is the subset of items which both user a and u have invoked previously, $r_{a,i}$ is a vector of item i observed (or rated) by user a, and $\bar{r}_a$ and $\bar{r}_u$ represent average values of different items observed (or rated) by user a and u, respectively.

The prediction method based on two users' similarity is referred as *user-based CF*. Similarly, CF can also be conducted through the similarity computation between two items, that is, *item-based CF*. According to the studies from other researchers, item-based CF can outperform user-based CF in most conditions, and has been treated as a preferred choice for prediction or recommendation problems.

As mentioned earlier, Shao *et al*. firstly adopted Pearson correlation-based CF for Web services' QoS prediction [4]. Recently, Zheng *et al*. improved prediction precision problem through combining item-based and user-based CF together [5]. Their WsRec algorithm exhibits better performance than other basic prediction methods, and has caused much attention in these two years.

2.2.2 Slope One method

Although previous studies have revealed that Pearson scheme CF can gain good prediction precision, its performance is not so satisfactory for the case of extremely sparse data. Meanwhile, Pearson-based method will cost a lot of computational overhead to measure the similarity between users or items. Fortunately, another well-known method called Slope One [9] can make up such deficiencies. On the one hand, Slope One can show good prediction effect for sparse data. On the other hand, this method can perform prediction activity with less computing cost.

As stated by Lemire *et al*., Slope One algorithm works on the intuitive principle of a "popularity differential"between items for users. In this algorithm, how much better one item is liked than another is determined in a pairwise fashion. Firstly, the difference between the averages of two items can be calculated via subtract operation. Then, once one item's value is available, the other's value can be predicted according to such difference. The process can be illustrated in Figure 1. For two users (a and b) and two items (i and j) in user-item matrix, the values of these two items for user a are known and the differential from i to j is 1.5-1=0.5. Thus, the item j's value for user b can be predicted via this mapping relationship, that is, 2+(1.5-1)=2.5. Of course, many such differentials exist in a training set for each unknown rating, the average of these differentials will be taken for predication.

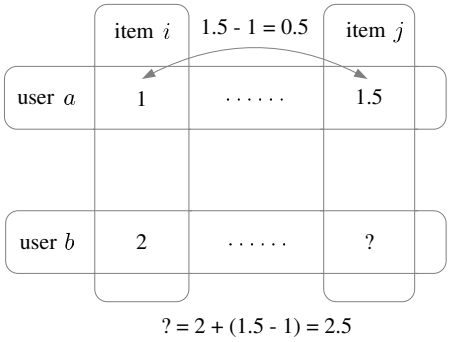


Figure 1: Illustration for Slope One prediction algorithm.

Formally speaking, for a given user-item matrix, the set of the users who contain rating records both on item i and item j can be computed and denoted as $U_{i,j}$ here. Obviously, $U_{i,j} = U_{j,i}$. Then, the average deviation of item i with respect to item j can be denoted as:

$$\text{dev}_{j,i} = \sum_{u\in U_{j,i}} \frac{r_{u,j}-r_{u,i}}{card(U_{j,i})} \quad (2)$$

where $card(U_{j,i})$ returns the element number of set $U_{j,i}$.

Based on the deviations of items, the rating of user u for item j, i.e. $r_{u,j}$, can be predicated via the following way.

$$P(r_{u,j}) = \frac{1}{card(R_j)} \sum_{i\in R_j} (\text{dev}_{j,i} + r_{u,i}) \quad (3)$$

where R_j=$\{i|r_{u,i} \neq \text{NA}, i \neq j \text{ and } card(U_{j,i}) > 0\}$ is the set of items which have co-occurrence relationship with item j.

The above discussion belongs to user-oriented prediction. Obviously, Slope One method can also be used in the other style, i.e., item-oriented prediction. In addition, several kinds of extensions are proposed. For instance, single or bivariate regression is used for finding the best mapping

relation [12, 13], bi-polar strategy is used for users' two different attitudes [9]. However, variant algorithms can't lead to obvious improvements over the basic form in all cases.

3 Similarity-Aware Slope One for QoS prediction

With regard to the usage scenario of Web services, services' QoS data from different users can form a sparse matrix of service invocation records. In order to help service user make a rational decision about service selection, the prediction for a specific service's QoS w.r.t. of the current user is very necessary. In this paper, we provide a hybrid prediction method through comprehensively adopt the merits both from Pearson correlation-based algorithm and Slope One algorithm.

3.1 The overall prediction framework

For an active service user u, the number of services which have been invoked by u is named *given number* (i.e. GN). For all n service items, GN is usually a little part. In order to provide precise QoS estimations for the remaining service items w.r.t user u, we should take full use of other users' invocation records for these services. Here, we assume the historical QoS data about m users for n service items is matrix $\mathcal{M}$. Similarly, each service user only has partial QoS information in that matrix. The proportion of existing QoS data in matrix is denoted as *density* (d for short).

In our investigations on collaborative filtering techniques, we have found a fact as follows: Slope One method is suitable for the very sparse data set (i.e. very low density data), whereas Pearson-based CF can achieve desired prediction results for the case of high density data. Therefore, in our method, we mainly adopt Slope One method for prediction and compute Pearson correlation between services to adjust the reference weight. The closer relation between a service and the subject service for user u, the higher weight should be assigned to the QoS deviation between these two services.

The whole procedure of Web service QoS prediction is shown in Figure 2. At the initial stage, the historical QoS records of n Web services for m users can be collected. Here, we call it training data $\mathcal{M}$. In general, a service user could not have QoS records for all n services, and usually has only very limited ones of them. As a result, training data is a sparse matrix in real-world scenarios. The matrix $\mathcal{M}$ should be filled as full as possible so that it can provide more useful information for QoS prediction. In the second step, we present a similarity-aware Slope One algorithm (SASO for short) as a way to fill the 'NA' (a.k.a. *null*) records in the training data set. For the perspective of Web service execution, there maybe exist some abnormal QoS records in the above training data, especially for the QoS attribute with wide scale values. In order to handle this problem, in the third step, we adopt *statistical process control* (SPC) strategy to adjust such exception data.

Based on the above treatments, the training data set has been enhanced and its data density has a great promotion. According to the renewed training matrix, SASO algorithm is also utilized for predicting Web service's QoS for active user. Finally, prediction quality is measured via error analysis.

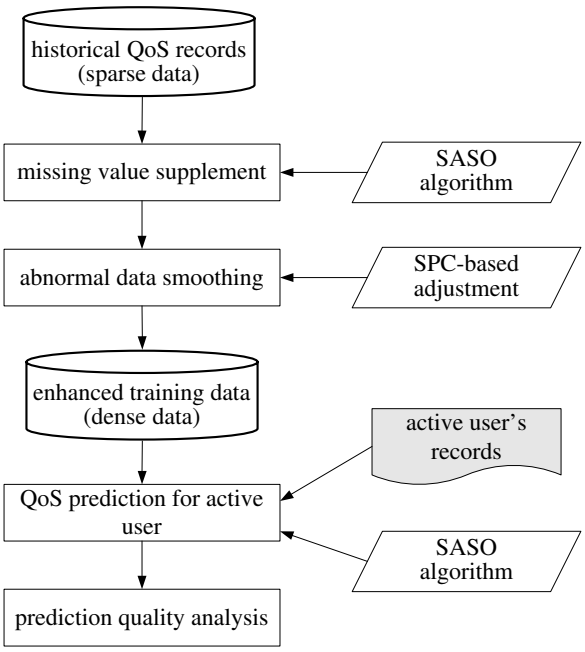


Figure 2: The overall framework of Web service's QoS prediction based on similarity-aware Slope One (SASO).

3.2 Prediction method

With regard to QoS prediction framework, it is not hard to find that SASO algorithm and SPC-based adjustment strategy play important roles for improving the precision. The details of these two key algorithms are addressed as follows.

3.2.1 SASO algorithm

As mentioned before, Slope One-based CF exhibits its advantage for sparse data. Since each active user has only GN (usually $GN << n$) QoS records for n Web services, we adopt item-oriented Slope One method to predict QoS value for active user. However, the similarity between items is not taken into consideration in the basic Slope One prediction method. In our work, we introduce the similarity between two items into Slope One method to form a new QoS prediction algorithm for Web services. The basic idea is that, the service with the higher similarity should give the higher priority when considering the deviation in Slope One method.

Here, we adopt item-based Pearson correlation to measure the similarity between two Web services. For service

i and j, their's similarity can be calculated as follows.

$$Sim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}} \quad (4)$$

where $U = U_i \bigcap U_j$ is the subset of users who have QoS records both on service i and service j previously (i.e., identical to $U_{i,j}$ in equation (2)), and $\bar{r}_i$ represents the average QoS value of service i observed by different users.

To predict a missing value $r_{u,j}$ in the user-item matrix, we have to measure the similarities between j and other services invoked by user u, that is, $I_u - \{j\}$. After removing the services with negative similarity to service j from them, the remaining items are called the related services of j w.r.t. user u, denoted as $R(j|u)$. It can be formally expressed as follows.

$$R(j|u) = \{i | i \in I_u,\ Sim(i,j) > 0,\ i \neq j\} \quad (5)$$

Then, we give the prediction formula based on similarity-aware Slope One algorithm as below.

$$P(r_{u,j}) = \frac{1}{card(R(j|u))} \sum_{i \in R(j|u)} (w_{i,j} \cdot \text{dev}_{j,i} + r_{u,i}) \quad (6)$$

where $w_{i,j}$ is an adjustment weight in accordance with the similarity between j and another service i ($i \in R(j|u)$).

As for a further comment, $w_{i,j}$ can be computed by the following formula:

$$w_{i,j} = \frac{Sim^{\lambda}(i,j)}{\sum_{k \in R(j|u)} Sim^{\lambda}(k,j)} \quad (7)$$

where λ (=1, 2 or 3) is a factor of adjustment strength, higher value means stronger adjustment. Meanwhile, $Sim^{\lambda}(k,j)$ is λ power of $Sim(k,j)$, i.e. $[Sim(k,j)]^{\lambda}$.

It should be noted that, both two steps of missing value supplement and the final QoS prediction for active user adopt SASO algorithm (cf. equation (6)) to provide prediction value.

3.2.2 SPC-based smoothing strategy

Here, we denote the original service's QoS record matrix as $\mathcal{M}$, and call the intermediate matrix after filling the missing values as $\mathcal{M}'$. On the one hand, some exceptional QoS records for Web services perhaps exist in $\mathcal{M}$. The so-called exceptional (or abnormal) record, means the QoS value of a specific user is far away from the records of neighbor users. On the other hand, it is also very sparse in the original state. As a result, the filled matrix $\mathcal{M}'$ maybe contain some QoS items which are far from the common situation. Obviously, these abnormal records will cause bad influence on the next stage of prediction. Thus, we should identify them out from matrix $\mathcal{M}'$ firstly, and then smooth them via a heuristic strategy.

In the paper, we borrow the idea from *statistical process control* (SPC) [14] to tackle the abnormal QoS data in $\mathcal{M}'$. SPC is a realtime monitoring technique for the process of industrial production in the way of statistical analysis. It can scientifically distinguish the exceptional fluctuation from the normal random fluctuation, so it is used for providing early warning for production process to manager. We mainly utilize this technique to pick out the abnormal QoS values so as to achieve better prediction performance.

At first, for matrix $\mathcal{M}'$, we judge whether item $r_{u,i}$ (i.e., the QoS of service i for user u) is an exception or not according to the following rule.

$$isAbn(r_{u,i}) = \begin{cases} true, & \mu_i - \theta \cdot \sigma_i < r_{u,i} \\ & < \mu_i + \theta \cdot \sigma_i \\ false, & \text{otherwise} \end{cases} \quad (8)$$

where μ_i is the average QoS value of service i ($1 \leq i \leq n$), and σ_i is the standard deviation of service i's QoS records from different users. θ is a positive integer used for regulating the normal range of QoS value. It is usually set to 3 in most applications of SPC.

When a suspected record of abnormal QoS is detected through the above approach, this isolated item should be smoothed before the prediction step. Here, we introduce a strategy called "small amplitude shift"for smoothing treatment. Suppose $r_{u,i}$ is an abnormal issue according to judgement of equation (8), the smoothing action can be performed via the following formula. The value after adjustment is denoted as $\tilde{r}_{u,i}$.

$$\tilde{r}_{u,i} = \begin{cases} \mu_i - \theta \cdot \sigma_i, & r_{u,i} < \mu_i - \theta \cdot \sigma_i \\ \mu_i + \theta \cdot \sigma_i, & r_{u,i} > \mu_i + \theta \cdot \sigma_i \\ r_{u,i}, & \text{otherwise} \end{cases} \quad (9)$$

That is to say, we use the upper (or lower) limit to replace the unusually high (or low) QoS record, respectively.

3.3 Computational complexity analysis

As shown in Figure 2, our algorithm mainly includes three linear steps as below.

(1) *Complexity of missing value supplement*. Obviously, the computational complexity for computing the similarity $Sim(i,j)$ between two services (i.e. i and j) is $O(m)$. Then, the complexity of computing similarities of all service pairs is $O(mn^2)$. At the same time, the computational complexity for calculating the deviation of each service pair is also $O(mn^2)$. Based on the above interim results, the complexity of providing the supplement value for each missing item is $O(dn)$ (here, d stands for *density*). Accordingly, $O(d(1-d)mn^2)$ is the complexity in respect to fill all missing items. Thus, the complexity of this step is $O(mn^2)$.

(2) *Complexity of abnormal data smoothing*. The complexity of computing the mean value of QoS is $O(m)$ for each Web service, so $O(mn)$ is for all n services. Meanwhile, the complexity of smoothing action for all items in matrix $\mathcal{M}$ is also $O(mn)$. Therefore, the complexity for smoothing the exceptional data is $O(mn)$.

(3) *Complexity of QoS prediction*. In the third step, each active user has $n - GN$ missing values to be predicated.

The complexity of computing the similarity and deviation between current Web service with all known GN services is $O(m \cdot GN)$. Therefore, the complexity of predicting all $n - GN$ missing values is $O(m \cdot GN(n - GN))$.

Altogether, the computational complexity of our approach for an active user is $O(mn^2)$. In literature [5], the complexities of five steps have been discussed in detail. Based on the comprehensive analysis on the above complexities, the overall computational complexity of WSRec algorithm is $O(m^2n + mn^2)$. As a result, our method and WSRec haven't obvious distinction from the perspective of computation time.

4 Implementation and Experiments

4.1 Experimental setup

In order to validate the effectiveness of our proposed algorithm for QoS prediction, some experiments are employed on a public published data set[1], which is collected by Zheng *et al.* [5] and has been widely adopted in the current researches [7, 15]. The original data set contains 5825 service invocation records from 339 users, and QoS attributes include *response time* (RT) and *throughput* (TP).

In our experiments, we select partial QoS records related with 100 Web services and 150 users from the original data. Then, this data is randomly divided into two parts: training data and test data. Here, 100 users are selected as training users, that is, the data about them is treated as training data. The remaining 50 users are viewed as test users (i.e., the *active user* in the above section). In the real-world situation, the known QoS records for a user only occupy a very small part of all 100 services. For satisfying the actual condition, some records are removed from training data matrix to construct three kinds of sparse data sets, whose data densities are set as 5%, 10% and 15%, respectively.

Similarly, for active users, we only retain GN (=5, 10 or 20) QoS records for each one of them. The records which are kicked out from test data set are treated as real data for prediction quality evaluation. The main parameters in our experiments are listed in Table 2.

4.2 Comparative analysis

In general, recommendation system uses mean absolute error (MAE) to evaluate prediction effect. It is the average of difference values between the predicted QoS and the real record.

$$\text{MAE} = \frac{\sum_{U,S} |P(r_{u,s}) - r_{u,s}|}{N} \tag{10}$$

where $P(r_{u,s})$ is the predicted QoS value of service s observed by user u, $r_{u,s}$ is the real QoS value of service s w.r.t. user u, and N is the total number of predictions.

Since the range of service's QoS value may be different from each other, MAE is not objective enough to reflect the accuracy of prediction algorithm. Here, we adopt the normalized MAE (NMAE) as a metric to compare the prediction quality of three algorithms.

$$\text{NMAE} = \frac{\text{MAE}}{\sum_{U,S} \frac{r_{u,s}}{N}} \tag{11}$$

It is not hard to find that, the smaller NMAE value means the more accurate prediction algorithm.

For the purpose of comparison, WSRec algorithm and basic Slope One algorithm are also implemented in our experiments. All three algorithms run on the same data set described in the above subsection. Other particular settings of WSRec algorithm are in accordance with reference [5]. For QoS attribute response time (RT) and throughput (TP), the comparisons on three algorithms are performed respectively. In the experiments, we repeated 100 times for each case of $density$ and GN value, and reported the average NMAE metrics.

The experimental results (i.e. NMAEs) on QoS attribute response time (RT) are shown in Table 3. It is clear that our SASO algorithm can outperform WsRec and basic Slope One algorithm for most cases. When $density$=5%, the basic Slope One can get the best result for the case of GN=5, but algorithm SASO (λ=1) overcomes other two algorithms for the remaining cases about GN. For the rest values (i.e. 10% and 15%) of $density$, algorithm SASO (λ=3) can achieve the lowest NMSE for nearly all cases except of $density$=15% and GN=20. On the whole, SASO's performance is better than those of WsRec and basic Slope One for almost all situations, especially λ=2 or 3.

The NMAE values of three algorithms on QoS attribute throughput (TP) are shown in Table 4. It is not hard to find our algorithm SASO (λ=3) has obvious improvement both for WsRec and basic Slope One, except of the case of $density$=15% and GN=20. With regard to SASO algorithm itself, the predication error of SASO reduces with the increase of λ value. When λ reaches to 2, algorithm SASO outperforms other two algorithms in most conditions.

According to the above experimental analysis, we can reasonably draw a conclusion that our SASO algorithm is a better choice than WsRec and basic Slope One for service's QoS prediction, especially when the data density of user-service record matrix is low.

4.3 Filling pattern analysis

As we stated earlier, the advantage of Slope One-based method is mainly for the sparse training data. However, Pearson correlation-based method will exhibit its merit with the increase of data density. In the above experiments, we only used one way (i.e. our SASO algorithm) to fill the missing data in training matrix. How about the effect of missing value supplement with two kinds of approaches? Suppose the density of training matrix during the procedure of missing value supplement is d' (obviously, $d' > d$), and the boundary point for switching filling approach is denoted as ρ. Here, we attempt to answer the above problem

[1] WS-DREAM data set, http://www.wsdream.net:8080 /wsdream/

No.	Parameter	Value	Description
1	m	150	The number of service users, 100 users for training and the rest for test.
2	n	100	Service number.
3	*density* (d)	5%, 10% or 15%	Data density of the training matrix.
4	Given Number (GN)	5, 10 or 20	The number of known QoS records for each active user.
5	λ	1, 2 or 3	The factor of adjustment strength.

Table 2: Parameter settings for service QoS prediction algorithm.

Algorithm		d=5%			d=10%			d=15%		
		GN=5	GN=10	GN=20	GN=5	GN=10	GN=20	GN=5	GN=10	GN=20
Slope One		**0.6306**	0.6142	0.6015	0.6050	0.5878	0.5718	0.5951	0.5819	0.5665
WsRec		0.6463	0.6240	0.6110	0.6001	0.5762	0.5578	0.5755	0.5596	**0.5221**
SASO	λ=1	0.6330	**0.6107**	**0.5957**	0.5923	0.5719	0.5570	0.5821	0.5645	0.5492
	λ=2	0.6350	0.6123	0.5960	0.5856	0.5654	0.5514	0.5721	0.5537	0.5401
	λ=3	0.6375	0.6155	0.5987	**0.5825**	**0.5627**	**0.5491**	**0.5652**	**0.5472**	0.5348

Table 3: Experimental results (NMAEs) for the algorithm Slope One, WsRec and SASO for the QoS attribute RT.

by using a two-stage filling strategy: SASO algorithm can be used for filling data when the matrix is relative sparse (i.e. $d' \leq \rho$). Once training matrix reaches to a certain degree of density (i.e. $d' > \rho$), we used Pearson correlation-based method to supply the missing values.

In order to validate the effect of the above two-stage filling pattern, the training matrixes with different ρ are prepared for SASO prediction algorithm. The experimental results are shown in Figure 3-6. On the whole, the two-stage filling strategy has certain improvement for some situations, but it is not so obvious. Specifically speaking, for the case of *density*=5%, the optimal boundary point is $\rho = 0.4$ for QoS attribute response time (TR). The NMAE value gradually declines when ρ is lower than 0.4. Instead, when ρ exceeds the best point 0.4, prediction error will slowly climb with the increase of ρ's value. On the second attribute throughput (TP), the trend of NMAE's change is relatively simple, that is, the error descends with the increase of value of boundary point (ρ).

For the second case *density*=10%, the change trends of prediction error for two QoS attributes are highly consistent. From the overall point of view, NMAE basically decreases along with the growth of ρ's value. However, NMAE has a little drop at the point of ρ=0.7. As a consequence, the best boundary point for this case is 0.7.

While considering the last case (i.e. *density*=15%), the change trend of prediction error for attribute RT is very similar to the second case of this attribute, just the current fluctuation is too small. There is an exceptional case for attribute TP when ρ=0.2, the corresponding NMAE value is suddenly low. Meanwhile, the prediction error value has a relatively high value at point ρ=0.3. Subsequently, it has a small reduction at first and then gradually takes off. The optimal boundary point in this case can be considered as 0.5 for most situations.

On the whole, we can argue that the two-stage filling strategy has a small improvement w.r.t prediction error. Considering the selection of boundary point, the value in the domain from 0.5 to 0.7 is worth considering in practice.

5 Related work

From the perspective of service users, how to select a suitable service is a critical step to build a reliable software system. In general, service selection is mainly in accordance with the property of QoS. Accordingly, QoS prediction for Web services has caused widespread attention in the field of service computing.

As we mentioned earlier, Pearson correlation-based algorithms are the main-stream strategies to treat such problem at current stage. Shao *et al.* [4] firstly attempted to use Pearson similarity-based collaborative filtering to provide the QoS value of a specific Web service. But their experiments are performed on a data set in small scale, and the error analysis is not so sufficient. Subsequently, Zheng *et al.* [5] firstly collected plenty of QoS records from different service users via a monitoring platform Planet-lab[2]. Then, they combined user-based and item-based CF together to form a comprehensive algorithm (i.e. WsRec) for service's QoS prediction. Their WsRec exhibits better performance than the single user-based or item-based prediction algorithm.

Recently, some improvements on Pearson correlation-based algorithm have been proposed. Liu's research group presented a personalized hybrid collaborative filtering (PHCF) algorithm by considering the personal information about service user [7]. However, it is not so easy to obtain such personal information, so the application of their method is limited. Reference [15] adopted an improved similarity measure for Web service similarity computation, and the corresponding normal recovery collaborative filtering (NRCF) was proposed for personalized Web service recommendation. In essence, it is only a minor modify

[2]http://www.planet-lab.org

Algorithm		d=5%			d=10%			d=15%		
		GN=5	GN=10	GN=20	GN=5	GN=10	GN=20	GN=5	GN=10	GN=20
Slope One		0.5115	0.5083	0.5054	0.4901	0.4873	0.4815	0.4793	0.4795	0.4707
WsRec		0.5326	0.5245	0.5195	0.4798	0.4724	0.4670	0.4579	0.4520	**0.4404**
SASO	λ=1	0.5050	0.5007	0.4968	0.4793	0.4736	0.4657	0.4687	0.4657	0.4571
	λ=2	0.5027	0.4967	0.4918	0.4735	0.4663	0.4582	0.4616	0.4566	0.4488
	λ=3	**0.5015**	**0.4946**	**0.4888**	**0.4701**	**0.4620**	**0.4533**	**0.4567**	**0.4502**	0.4426

Table 4: Experimental results (NMAEs) for the algorithm Slope One, WsRec and SASO for the QoS attribute TP.

on the similarity measure for the WsRec prediction framework. In addition, Shi *et al*. [16] presented a linear regression prediction algorithm for Web service's QoS based on clustering user in respect to location and network condition. It is not hard to find that the distance between users plays a significant role for prediction precision, however, which is not easily measured in practice.

Of course, there are also some Slope One-based methods for service's QoS prediction. Reference [6] presented a personalized context-aware QoS prediction method based on the Slope One approach. In this work, the basic Slope One algorithm is used for prediction, but it has been validated to be not very precise in our experiments. Then, Li *et al*. [17] utilized an enhanced Slope One method called Bi-Polar Slope One to predict the ratings of Web services. On the one hand, their approach mainly aims at the rating prediction problem. On the other hand, Bi-Polar phenomenon maybe exists in the data set in rating style, but not obvious in QoS data (i.e. the continuous data type).

With regard to the combination of Slope One and Pearson similarity, the preliminary researches in [18] and [19] have contributed an incipient idea for blending them together. However, the above works merely provide a primitive form of similarity-aware Slope One prediction algorithm, that is, the case of λ=1 in our work. As shown in our experimental results, this basic form without weight adjustment is not very effective for QoS prediction problem. At the same time, the experimental analysis and discussion are very limited in their work. Besides the weight adjustment strategy illustrated in formula (7), here, a more important strategy named SPC-based smoothing is also proposed to reduce prediction error.

6 Conclusion

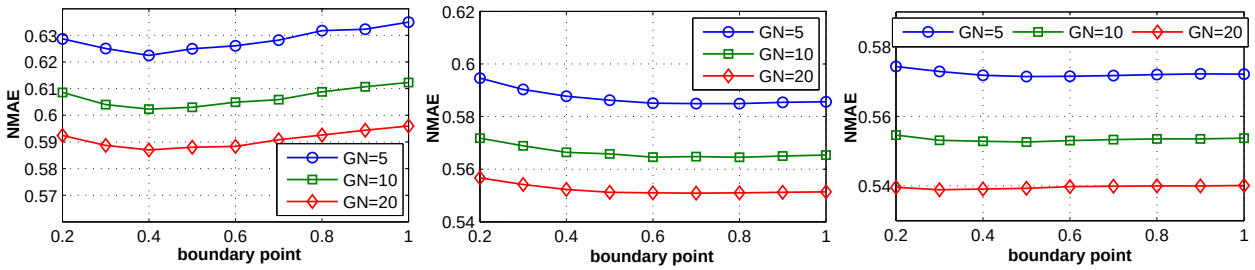
With the widespread application of service computing, Web services have been viewed as a prevalent form of components for building software on the Web. In order to ensure the reliability and trustworthy of the composite software system, users generally are very concerned about the quality of service. Unfortunately, the QoS metrics of some services can not be provided due to the actual situation. Therefore, how to predicate QoS of Web service becomes a valuable task in the field of service engineering.

In the paper, we introduce the Pearson similarity between Web services into Slope One collaborative filtering for solving QoS prediction problem. Instead of assigning the identical weight to each service, we adjust Pearson similarity as a weight for differentiating the deviation between services. In order to improve the prediction accuracy, a SPC-based smoothing is presented for correcting the exceptional data. In the empirical aspects, besides our approach, the basic Slope One and the well-known WsRec algorithm are also implemented. Meanwhile, the comparative analysis is also performed on the public published data set. The experimental results indicate that our hybrid algorithm (SASO) outperforms other two methods in the term of prediction precision. The SPC-based smoothing strategy can effectively handle the noise data so as to reduce prediction error. Furthermore, an additional strategy called two-stage filling is studied, and the appropriate boundary point for transforming filling methods is also suggested here.

The practice of SASO algorithm is obvious, it can guide users to pick out desired services from cloud platform. At the same time, this algorithm can also be used in the field of E-commerce to help consumers choose goods. Of course, although our approach achieves some promising results at present, there are still quite a few complicated issues should be further investigated. For instance, the QoS prediction for Web services from the dynamic perspective [20], as well as the service quality prediction in the environment of mobile computing. In addition, to find more effective data filling algorithm for training data is an interesting research direction.

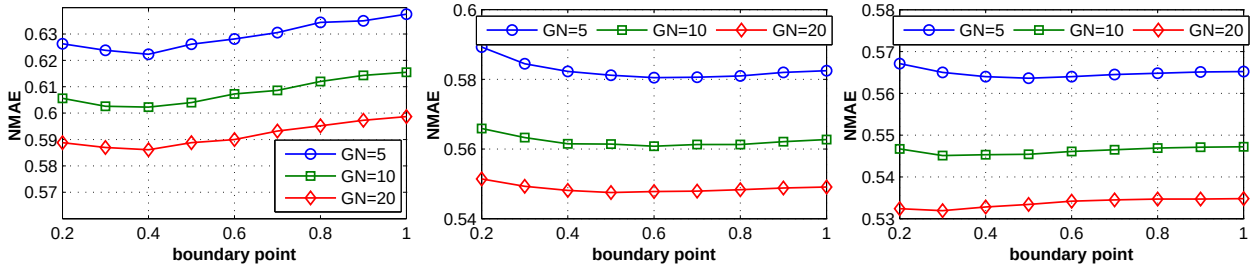
Acknowledgement

The authors would like to express their appreciation to the reviewers for their thoughtful and constructive comments. The authors would also like to thank Xiaomei Xie for her helpful feedback on the earlier version of this paper. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 60803046 and 61063013, the Natural Science Foundation of Jiangxi Province under Grant No. 2010GZS0044, the Science Foundation of Jiangxi Educational Committee under Grant No. GJJ10433, the Open Foundation of State Key Laboratory of Software Engineering under Grant No. SKLSE2010-08-23, and the Program for Outstanding Young Academic Talent in Jiangxi University of Finance and Economics.



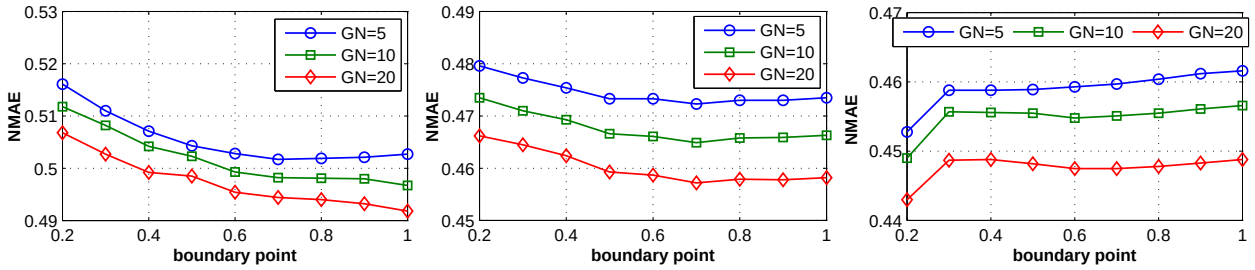
(a) *density*=5% (b) *density*=10% (c) *density*=15%

Figure 3: The NMAEs on attribute RT for different boundary points (λ=2).



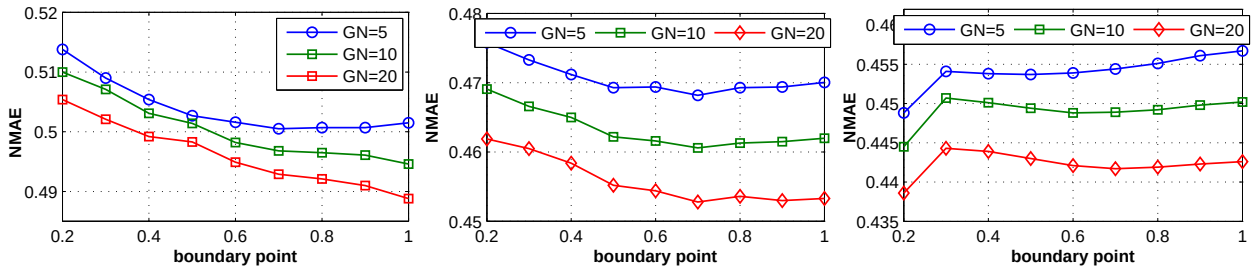
(a) *density*=5% (b) *density*=10% (c) *density*=15%

Figure 4: The NMAEs on attribute RT for different boundary points (λ=3).



(a) *density*=5% (b) *density*=10% (c) *density*=15%

Figure 5: The NMAEs on attribute TP for different boundary points (λ=2).



(a) *density*=5% (b) *density*=10% (c) *density*=15%

Figure 6: The NMAEs on attribute TP for different boundary points (λ=3).

References

[1] M. P. Papazoglou and D. Georgakopoulos, (2003). Serive-Oriented Computing, *Communcications of the ACM*, Vol. 46, No. 10, ACM Press, pp. 25–65.

[2] D. A. Menasce, (2002). QoS issues in Web services, *IEEE Internet Computing*, Vol. 6, No. 6, IEEE CS Press, pp. 72–75.

[3] J. B. Schafer, J. Konstan, and J. Riedi, (1999). Recommender Systems in E-Commerce, *Proc. of the 1st ACM Conference on Electronic Commerce (EC'09)*, ACM Press, Denver, CO, USA, pp. 158–166.

[4] L. Shao, J. Zhang, Yong Wei, and *et al*., (2007). Personalized QoS Prediction forWeb Services via Collaborative Filtering, *Proc. of the IEEE International Conference on Web Services (ICWS'07)*, IEEE CS Press, Salt Lake City, Utah, USA, pp. 439–446.

[5] Z. Zheng, H. Ma, M. R. Lyu, and I. King, (2011). QoS-Aware Web Service Recommendation by Collaborative Filtering, *IEEE Trans. on Services Computing*, Vol.4, No. 2, IEEE CS Press, pp. 140–152.

[6] Q. Xie, K. Wu, J. Xu, and *et al*., (2010). Personalized Context-Aware QoS Prediction for Web Services Based on Collaborative Filtering, *Proc. of the 6th International Conference on Advanced Data Mining and Applications (ADMA'10)*, Part II, Springer-Verlag Berlin, Chongqing, China, pp. 368–375.

[7] Y. Jiang, J. Liu, M. Tang, and X. Liu, (2011). An Effective Web Service Recommendation Method based on Personalized Collaborative Filtering, *Proc. of IEEE International Conference on Web Services (ICWS'11)*, IEEE CS Press, Washington, DC, USA, pp. 211–218.

[8] A. K. Menon, K. P. Chitrapura, S. Garg, and *et al*., (2011). Response Prediction using Collaborative Filtering with Hierarchies and Side-information, *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, ACM Press, San Diego, CA, USA, pp. 141–149.

[9] D. Lemire and A. Maclachlan, (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering, *Proc. of the 2005 SIAM International Data Mining Conference (SDM'05)*, Newport Beach, California, USA, pp. 1–5.

[10] X. Su and T. M. Khoshgoftaar, (2009). A Survey of Collaborative Filtering Techniques, *Advances in Artificial Intelligence*, Hindawi Publishing Corporation, pp. 1–19.

[11] Linyuan Lü, Matéš Medo, Chi Ho Yeung, and *et al*., (2012). Recommender Systems, *Physics Reports*, Vol. 519, No. 1, Elsevier B. V., pp. 1–49.

[12] S. Vucetic and Z. Obradovic, (2000). A Regression-based Approach for Scaling-up Personalized Recommender Systems, *Proc. of the ACM WebKDD Workshop (WebKDD'00)*, Boston, MA, USA, pp. 1–9.

[13] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, (2001). Item-based Collaborative Filtering Recommender Algorithms, *Proc. of the 10th International Conference on World Wide Web (WWW'01)*, ACM Press, Hong Kong, China, pp. 285–295.

[14] J. Oakland, (2003). *Statistical Process Control*, the 5th Revised Edition, Butterworth-Heinemann Ltd.

[15] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, (2011). NRCF: A Novel Collaborative Filtering Method for Service Recommendation, *Proc. of IEEE International Conference on Web Services (ICWS'11)*, IEEE CS Press, Washington, DC, USA, pp. 702–703.

[16] Y. Shi, K. Zhang, B. Liu, and L. Cui, (2011). A New QoS Prediction Approach Based on User Clustering and Regression Algorithms, *Proc. of IEEE International Conference on Web Services (ICWS'11)*, IEEE CS Press, Washington, DC, USA, pp. 726–727.

[17] J. Li, L. Sun, and J. Wang, (2012). A Slope One Collaborative Filtering Recommendation Algorithm Using Uncertain Neighbors Optimizing, *Proc. of WAIM 2011 International Workshops*, Springer-Verlag Berlin, Wuhan, China, pp. 160–166.

[18] P. Wang and H. W. Ye, (2009). A Personalized Recommendation Algorithm Combining Slope One Scheme and User Based Collaborative Filtering, *Proc. of International Conference on Industrial and Information Systems (IIS'09)*, IEEE CS Press, Haikou, China, pp. 152–154.

[19] D. J. Zhang, (2009). An Item-based Collaborative Filtering Recommendation Algorithm Using Slope One Scheme Smoothing, *Proc. of the Second International Symposium on Electronic Commerce and Security (ISECS'09)*, Vol. 2, IEEE CS Press, Nanchang, China, pp. 215–217.

[20] X. Li, F. Zhou, and X. Yang, (2011). Research on Trust Prediction Model for Selecting Web Services based on Multiple Decision Factors, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 21, No. 8, World Scientific Publishing Company, pp. 1075–1096.