Reputation-aware QoS Value Prediction of Web Services

Weiwei Qiu*, Zibin Zheng†, Xinyu Wang*, Xiaohu Yang* and Michael R. Lyu†

*College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China, 310027
Email: {qiuweiwei, wangxinyu, yangxh}@zju.edu.cn
†Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
Email: {zbzheng, lyu}@cse.cuhk.edu.hk

Abstract—QoS value prediction of Web services is an important research issue for service recommendation, selection and composition. Collaborative Filtering (CF) is one of the most widely used methods which employs QoS values contributed by similar users to make predictions. Therefore, historical QoS values contributed by different users can have great impacts on prediction results. However, existing Web service QoS value prediction approaches did not take data credibility into consideration, which may impact the prediction accuracy. To address this problem, we propose a reputation-aware QoS value prediction approach, which first calculates the reputation of each user based on their contributed values, and then takes advantage of reputation-based ranking to exclude the values contributed by untrustworthy users. CF QoS prediction approach is finally used to predict the missing QoS values based on the purified dataset. Experimental results show that our approach has higher prediction accuracy than other approaches.

Keywords-QoS value prediction; reputation-based ranking; collaborative filtering; Web service;

I. INTRODUCTION

With the increasing popularity of service-oriented computing and cloud computing, more and more functionally equivalent Web services are available on the Internet. Quality-of-Service (QoS) becomes a differentiating aspect for functionally equivalent Web services. QoS is widely employed for service recommendation [1], service selection [2], [3], service automatic composition [4] and so on. Influenced by the dynamic Internet environment, it is impractical for service users to acquire accurate QoS evaluation values all the time. Thus, personalized Web service QoS value prediction becomes a promising approach for obtaining user-observed QoS values.

In recent years, extensive research work [1], [5], [6] has been conducted on Web service QoS value prediction. Amongst them, Collaborative Filtering (CF) is widely used. The typical process of CF QoS value prediction is first to identify a set of similar users or services based on Pearson Correlation Coefficient (PCC), and then use the QoS values of similar users and/or services to make predictions. The predictions are made based on the historical QoS values contributed by different users. Therefore, the prediction accuracy of CF approaches is highly influenced by the trustworthiness of the user-contributed QoS values. However, the

existing CF prediction approaches [1], [5] are based on the hypothesis that all user-contributed values on services are trustworthy. In reality, however, some user-contributed QoS values can be untrustworthy for the following reasons: (1) malicious users may submit random values as their service QoS evaluation data; (2) some users may always give the maximal/ minimal value for their unevaluated services,(3) for service users who are service providers at the same time, they may give high QoS values of their own services and bad mouthing their competitors' service QoS values. Therefore, it is important to take data credibility into consideration to enable more robust Web service QoS value prediction.

Inspired by user reputation ranking of traditional recommendation systems, we calculate a reputation value for each user based on the difference of the user-contributed QoS values and the weighted average of other users' QoS values. Users with low reputation values will be taken as untrustworthy users and theit past data will be not be used for the QoS prediction. Based on this intuition, we propose a Reputation-Aware Prediction (RAP) approach in this paper. RAP first designs a reputation ranking algorithm to calculate and rank the reputation values for the service users. After that, untrustworthy users are excluded. Finally, predictions are conducted for missing QoS values by employing information of both similar users and similar services. Extensive experimental results show that our reputation-aware approach can solve the data credibility problem and significantly improve the prediction accuracy.

The main contribution of this paper includes:

- This paper presents a user reputation ranking algorithm to identify untrustworthy users and solve the data credibility problem.
- A reputation-aware QoS value prediction approach RAP is proposed by combining user reputation and existing neighborhood-based collaborative filtering methods. Extensive experiments based on real-world Web service QoS values are conducted to show the effectiveness of our RAP approach.

The remainder of this paper is organized as follows: Section II reviews the related work on collaborative filtering and reputation-based systems. Section III presents the prediction framework. Section IV describes our RAP reputation-aware

prediction algorithm. Section V shows experimental results and Section VI concludes the paper.

II. RELATED WORK

A. Collaborative Filtering

Collaborative filtering is the process of filtering for information or patterns by involving collaboration among multiple agents [7]. Collaborative filtering methods are widely used for recommendation systems [8], [9]. There are mainly two types of collaborative filtering methods: memory-based methods and model-based methods. Memory-based approaches calculate similarities between users or items based on user rating data. On the other hand, model based approaches develop models by employing data mining and machine learning algorithms based on training data. Compared to the model based methods, memory-based approaches are widely used in practice since they are easier to be understood and implemented. Typical memory-based approaches include user-based approach [10], item-based approach [11] and the hybrid approach [12]. In recent years, a number of research investigations have applied the collaborative filtering techniques for Web service QoS prediction. Shao et al. [13] employed user-based collaborative filtering technique to predict QoS values. Zheng, et al. combined user-based and item-based approaches to achieve a better prediction accuracy [1]. Work [14] releases real world Web service QoS datasets for experimental studies. However, these approaches did not recognize and exploit the characteristics of different Web services and users, so the prediction accuracy is unsatisfied.

In order to improve the prediction accuracy, various enhancement methods are proposed [15], [16], [17]. Work [15] incorporated the influence of the personalization of Web service items when computing degree of similarity between users. Work [16] proposed the RegionKNN approach, which takes advantage of user location information. Service location information is used in addition to user location information work [17] to make predictions. Although these approaches improve the prediction accuracy compared to the basic approaches, none of them take data credibility into consideration.

In [18], the authors take data credibility into account and build their own trust and reputation management model. Their approach depends on third party evaluation data to detect false ratings by dishonest users and providers. However, it is difficult to find such trustworthy third parties in practice. Furthermore, influenced by the dynamic network environment, some QoS properties (e.g., response time, failure rate, etc.) may vary over time among different users, thus it is also difficult to acquire accurate evaluation data to distinguish false ratings from user specific observations.

B. Reputation Systems

Reputation systems compute and publish reputation scores for a set of entities based on a collection of ratings and feedbacks from other entities. The reputation score can be taken as a measurement of trustworthiness for certain entity. Many online services such as Amazon, Epinions, and eBay implement reputation systems to reduce the risk of fraud and deception. Work [19] summarizes some reputation computation methods such as simple summation or average of ratings, Bayesian systems, Discrete Trust Models, and so on. Another approach is co-determination algorithm introduced in [20] which uses the difference between current user's ratings and the corresponding objects' aggregated ratings of other users to measure the user reputation. That is, the user whose ratings are often different from those of other users will get a low reputation value. However the co-determination algorithm has the convergence problem. The recently presented reputation-based ranking algorithms in [21] solved the convergence problem. Based on these previous research investigation, we propose a reputation-aware approach for Web service QoS value prediction in this paper.

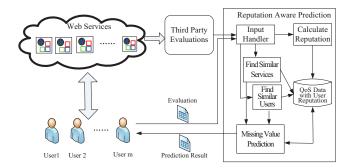## III. THE PREDICTION FRAMEWORK

Figure 1.   Reputation-aware QoS Value Prediction Framework

To make accurate Web service QoS value predictions, sufficient QoS values from different users are required. However, it is difficult to obtain the QoS values of Web services invoked by different users. To address this problem, collaborative frameworks such as WSRec [1], WSP [22] are proposed to share Web service QoS values between users. However, these frameworks typically make predictions based on their own collected dataset which ignored the information from active users, or make prediction based on the collected QoS values from users without doubt. Both practise have drawbacks. The former neglects useful information, while the latter is vulnerable to malicious users.

To address this problem, we propose a reputation-aware prediction framework as shown in Figure 1, which includes the following procedures: (1) Active users submit QoS prediction requests on target services with their individually
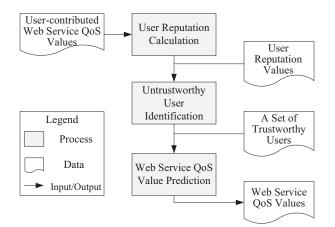
Figure 2. Process of Reputation-aware QoS Value Prediction

obtained Web service QoS values; (2) the *Input Handler* module processes and formats the input data; (3) the *Calculate Reputation* module calculates the reputation of active users based on the QoS values contributed by other users, and stores the Web service QoS values submitted by active users with user reputation values; (4) the *Find Similar Users* module finds similar users from the database; (5) the *Find Similar Services* module finds similar services from the database; and (6) the *Missing Value Prediction* module predicts the QoS value of target services for active users employing a hybrid collaborative filtering algorithm and returns the prediction results to active users.

IV. REPUTATION-AWARE QOS VALUE PREDICTION

As illustrated in Figure 2, our reputation-aware QoS value prediction approach is designed as a three-phase process. First, the reputation values of different service users are calculated. Second, a set of suspected users are identified based on the reputation values. Third, QoS predictions are made by employing neighborhood-based collaborative filtering on QoS values contributed by trustworthy users. The details of each process are presented in Section IV-A to Section IV-C, respectively.

A. User Reputation Calculation

Reputation-aware QoS value prediction aims to improve the prediction accuracy by excluding QoS values contributed by untrustworthy users. The trustworthiness of users are determined by user reputation. Assume there are m service users and n Web services, the user-service relationship is represented by a $m \times n$ matrix M. Each entry $q_{i,j}$ in the matrix denotes the QoS value of service j obtained by user i. The reputation r_i of user i is calculated by:

$$r_i^{k+1} = 1 - \frac{d\sum_{j \in I_i} |q_{i,j} - avg_j^{k+1}|}{|I_i|}, \qquad (1)$$

where I_i is the set of services invoked by user i, avg_j denotes the aggregated evaluation value for service j, and d is the damping factor whose value is in (0,1). In Eq.(1), avg_j^{k+1} can be calculated by:

$$avg_j^{k+1} = \frac{1}{|U_j|} \sum_{i \in U_j} q_{i,j} \cdot r_i^k, \qquad (2)$$

where U_j denotes the set of users which invokes service j, and r_i^k (the weight of different users) is the reputation value of user i calculated in last iteration (i.e., the kth iteration).

From the above equations, we can see that the calculation of user reputation is iterative (k denotes the kth iteration). The reputation values of users are determined by the difference of their evaluation data $q_{i,j}$ and the weighted average of other users' evaluation on the same service.

We use the following equation to initialize the calculation process:

$$avg_j^1 = \frac{1}{|U_j|} \sum_{i \in U_j} q_{i,j}, \qquad (3)$$

which indicates that each user is taken as trustworthy and has a reputation value of 1 at the very beginning (i.e., $r_i^0 = 1$). This initialization is also compliant to Eq. (1) and Eq. (2).

B. Untrustworthy User Identification

After calculating reputation values of users, the users can be ranked according to their reputation values. In our reputation-aware approach, a parameter Top-R is employed to identify untrustworthy users. Top-R indicates that R users who have lower reputation values than others, will be identified as untrustworthy users:

$$\overline{U} = \{u | r_u \leq r_R\} \qquad (4)$$

where $\overline{U}$ denotes the set of untrustworthy users, and r_R is the Rth lowest reputation value among all users.

In reality, however, it is difficult to determine the value of Top-R. A large Top-R can weed out trustworthy users and harm the prediction accuracy. A small Top-R value seems safe, but may limit the effectiveness of the approach. Optimal value of Top-R is highly related to the application domain and need to be identified based on experiences. Comprehensive experimental studies of the parameter Top-R will be conducted in Section V-D.

C. Reputation-aware Collaborative Filtering Algorithm

Pearson Correlation Coefficient (PCC) is employed for the similarity computation. When calculating the similarity between users, only trustworthy users are taken into consideration. That is, for service user u and service user $a \notin \overline{U}$:

$$Sim(u,a) = \frac{\sum\limits_{i \in I_u \cap I_a} (q_{u,i} - \overline{q_u})(q_{a,i} - \overline{q_a})}{\sqrt{\sum\limits_{i \in I_u \cap I_a} (q_{u,i} - \overline{q_u})^2} \sqrt{\sum\limits_{i \in I_u \cap I_a} (q_{a,i} - \overline{q_a})^2}} \qquad (5)$$

43

where $I_u \cap I_a$ is the set of Web services invoked by both user u and user a, $q_{u,i}$ is the QoS value observed by user u on service j, and $\overline{q_u}$ denotes the average QoS value of all Web services invoked by user u.

Similarly, only the data of trustworthy users are used to calculate service similarity. For Web service i and Web service j:

$$
Sim(i,j) = \frac{\sum\limits_{u \in U_i \cap U_j, u \notin \overline{U}} (q_{u,i} - \overline{q_i})(q_{u,j} - \overline{q_j})}{\sqrt{\sum\limits_{u \in U_i \cap U_j, u \notin \overline{U}} (q_{u,i} - \overline{q_i})^2} \sqrt{\sum\limits_{u \in U_i \cap U_j, u \notin \overline{U}} (q_{u,j} - \overline{q_j})^2}}
\tag{6}
$$

where $U_i \cap U_j$ denotes the set of service users who invoked both service i and j, $u \notin \overline{U}$ represents u is a trustworthy user, and $\overline{q_i}$ is the average QoS value of service i observed by all its users.

The set of similar users $S(u)$ can be identified by:

$$
S(u) = \{a | Sim(u,a) \geq Sim_k, Sim(u,a) > 0, a \neq u, a \notin \overline{U}\}
\tag{7}
$$

where $a \notin \overline{U}$ ensures only trustworthy users can be selected as similar users.

Similarly, the set of similar services is identified by:

$$
S(i) = \{j | Sim(i,j) \geq Sim_k, Sim(i,j) > 0, j \neq i\}
\tag{8}
$$

By employing both the similar user and user reputation information, the reputation-aware user based predict the missing value $q_{u,i}^u$ in the user-item matrix by the following equation.

$$
q_{u,i}^u = \widetilde{q_u} + \frac{\sum\limits_{a \in S(u)} Sim(u,a)(q_{a,i} - \widetilde{q_a})}{\sum\limits_{a \in S(u)} Sim(u,a)}
\tag{9}
$$

where

$$
\widetilde{q_u} = \frac{r_u \cdot \overline{q_u}}{\sum\limits_{j \in S(u)} r_j}
\tag{10}
$$

is the reputation weighted average QoS value of all Web services invoked by user u.

Similar to the user-based approach, we employ user reputation weighted service average QoS value and the information of similar Web services to predict missing value $q_{u,i}^i$:

$$
q_{u,i}^i = \widetilde{q_i} + \frac{\sum\limits_{k \in S(i)} Sim(i,k)(q_{u,k} - \widetilde{q_k})}{\sum\limits_{a \in S(u)} Sim(i,a)}
\tag{11}
$$

where

$$
\widetilde{q_i} = \frac{r_u \cdot \overline{q_i}}{\sum\limits_{j \in U_i} r_j}
\tag{12}
$$

$\overline{q_i}$ is the average QoS value of all trustworthy users who invoked service i. That is:

$$
\overline{q_i} = \frac{\sum\limits_{k \in U_i, k \notin \overline{U}} q_{k,i}}{|U_i| - |U_i \cap \overline{U}|}
\tag{13}
$$

$|U_i|$ denotes the number of users who have invoked service i, and $|U_i| - |U_i \cap \overline{U}|$ is the number of trustworthy users who invoked service i.

To fully utilize the information of trustworthy similar users as well as similar services, we combine Eq. (9) and Eq. (11) and propose a hybrid approach RAP. When $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$, RAP predicts the missing QoS value $q_{u,i}$ by the following equation:

$$
q_{u,i} = \lambda \times q_{u,i}^u + (1 - \lambda) \times q_{u,i}^i
\tag{14}
$$

The parameter λ determines the proportion that the prediction relies on the similar users compared to the similar Web services. When $S(u) \neq \emptyset \wedge S(i) = \emptyset$, the RAP approach degrades to the user-based approach, and the missing QoS values can be calculated by employing Eq. (9). When $S(u) = \emptyset \wedge S(i) \neq \emptyset$, predictions are made by employing Eq. (11), which only relies on the similar service information. When $S(u) = \emptyset \wedge S(i) = \emptyset$, there are no similar user or service information, the following equation is used for prediction:

$$
q_{u,i} = \begin{cases} \lambda \widetilde{q_u} + (1 - \lambda) \widetilde{q_i} & \widetilde{q_u} \neq null \wedge \widetilde{q_i} \neq null \\ \widetilde{q_u} & \widetilde{q_u} \neq null \wedge \widetilde{q_i} = null \\ \widetilde{q_i} & \widetilde{q_u} = null \wedge \widetilde{q_i} \neq null \\ NoPrediction & \widetilde{q_u} = null \wedge \widetilde{q_i} = null \end{cases}
\tag{15}
$$

There are four cases as shown in Eq. (15): (1) The user u has invoked other services previously and the service i has been invoked by other users. The reputation weighted user-average $\widetilde{q_u}$ and service average value $\widetilde{q_i}$ are combined to make predictions.(2) Service i has not been invoked by any user, but user u has invoked other services. Then the weighted average QoS value $\widetilde{q_u}$ of user u is used for prediction. (3) User u did not invoke any service previously, but service i has been invoked by other users. Then the weighted average service value $\widetilde{q_i}$ is employed for prediction. (4) If user u did not invoke any service previously and the service i is never invoked by any user, there is no information available. Therefore no prediction will be provided.

By the three-phrase process presented in Section (IV), personalized QoS value prediction can be made in the presence of untrustworthy users. Extensive experiments are conducted in Section (V) to evaluated the performance RAP and the impact of different parameters on the prediction accuracy.

V. EXPERIMENTS

In this section, we conduct experiments on real world Web service QoS data. The experiments aim to (1) compare RAP

44

to the popularly used UPCC, IPCC and UIPCC methods under different percentage of untrustworthy users; (2) investigate the impact of various parameters to the prediction accuracy.

A. Experimental Setup

We conducted the experiments on the dataset collected by Zheng et. al [14]. The dataset contains real-world QoS values from 339 users on 5825 Web services. In order to make the experiments more realistic, we first randomly remove values from the 339*5825 user-service matrix to a certain *Matrix density*, and then randomly select *p percent of users* and generate random values to substitute the QoS values of their evaluated services. As a result, these *p percent of users* are acting as untrustworthy users. Extensive experiments are conducted on the modified dataset to compare the performance of different approaches and evaluate the impact of different parameters.

B. Metrics

In our experiments, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are employed as measurement criterias of the prediction accuracy for different approaches. MAE is defined as:

$$MAE = \frac{\sum_{u,i} \left| R_{ui} - \widehat{R}_{ui} \right|}{N} \qquad (16)$$

where R_{ui} denotes the observed response time of user u on service i, $\widehat{R}_{ui}$ is the predicted value, and N is the number of all predicted values.

RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{u,i}(R_{ui} - \widehat{R}_{ui})^2}{N}} \qquad (17)$$

MAE is the average of differences between the predicted and observed values. It gives the same weight for each individual difference. In RMSE, lager errors are weighted higher, since the errors are squared before they are averaged. A smaller MAE or RMSE value indicates better prediction accuracy.

C. Comparison

In this section, we compare our RAP approach with the following methods:

- UPCC (User-based collaborative filtering method using Pearson Correlation Coefficient): This method makes predictions based on the observed values of similar users, which is introduced by [13].
- IPCC (Item-based collaborative filtering method using Pearson Correlation Coefficient): This method employs the information of similar items (Web services) to make predictions, which is first introduced by [11].
- UIPCC: This method is an combination of UPCC and IPCC. Predictions are made by employing the information of both the similar users and similar items [23].

In this experiment, the *Matrix density* is set as 10%, which means that 10% entries in the matrix are used for predicting QoS values of the other 90% entries. The percentage of untrustworthy users is varied from 0.5% to 5% by a step of 0.05%. The top 30 similar users and services are used to make predictions. The dumping factor d in equation (1) is set as 0.1. The *Top-R* is varied from 2 to 20 by a step of 2, and the optimal prediction result is adopted as the final value. The performance comparison under the two metrics of different approaches are shown in Table I.

The experimental results show that:

- RAP approach obtains smaller MAE and RMSE under all untrustworthy user percentage settings, which indicates that RAP outperforms UPCC, IPCC and UIPCC. The last rows of each metrics in Table I show the percentages of the accuracy improvement compared with the best of other existing approaches.
- The prediction accuracy of UPCC, IPCC and UIPCC all decrease as the percentage of untrustworthy users increases. While the performance of RAP is more stable than the other three approaches. Since the users with randomly generated QoS values can be identified by RAP and their information will not be used for the QoS prediction.
- The UPCC method is the most vulnerable among all approaches. As the percentage of untrustworthy users increases, its prediction accuracy declines substantially (MAE declines by 62%, and RMSE declined by 57%). Although IPCC and UIPCC performs better than UPCC in the presence of untrustworthy users, their prediction accuracy is still lower than that of RAP. For instance, the RAP approach has about 25% improvement on MAE and 8% on RMSE compared with the UIPCC method when the percentage of untrustworthy users is 4%.
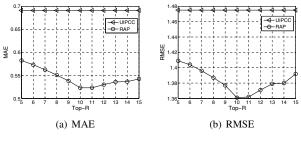
D. Impact of Top-R

(a) MAE   (b) RMSE

Figure 3.   **Impact of Top-R**

The parameter *Top-R* determines the number of untrustworthy users as mentioned in Section IV-B. That is, the R users with lower reputation than others are taken as untrustworthy users. The QoS value contributed by untrustworthy users will not be used for prediction. To study the impact of

45

Table I
PERFORMANCE COMPARISON

Metrics	Methods	Percentage of Untrustworthy Users									
		0.5%	1%	1.5%	2%	2.5%	3%	3.5%	4%	4.5%	5%
MAE	UPCC	0.569	0.622	0.650	0.696	0.747	0.794	0.801	0.858	0.923	0.926
	IPCC	0.672	0.684	0.692	0.700	0.711	0.721	0.724	0.734	0.741	0.747
	UIPCC	0.557	0.579	0.614	0.644	0.669	0.691	0.693	0.712	0.725	0.732
	RAP	0.517	0.519	0.526	0.522	0.528	0.524	0.537	0.530	0.526	0.537
Improvements(%)		6%	10%	14%	19%	21%	24%	23%	25%	27%	27%
RMSE	UPCC	1.503	1.659	1.731	1.857	1.986	2.904	2.108	2.234	2.338	2.363
	IPCC	1.511	1.519	1.524	1.525	1.534	1.537	1.542	1.551	1.552	1.555
	UIPCC	1.380	1.417	1.434	1.447	1.469	1.475	1.478	1.494	1.498	1.506
	RAP	1.352	1.353	1.368	1.356	1.365	1.358	1.365	1.373	1.366	1.381
Improvements(%)		2%	5%	5%	6%	7%	8%	8%	8%	8%	8%

Top-R, we vary the value of *Top-R* from 5 to 15 by a step of 1. The matrix density is set as 10%, *Top-K* is set as 30, and the value of 3% randomly selected users are replaced by random values. The experimental results are illustrated in Figure 3.

The experimental results show that:

- The performance of the RAP approach gets better when the value of *Top-R* approaches the actual number of untrustworthy users (i.e., 10 in this experiment). When the value of *Top-R* is less than 10, there are some untrustworthy users left in the matrix for making prediction, which impact the prediction accuracy.
- On the other hand, when the value of *Top-R* is larger than 10, some normal users who provided real observed data are taken as untrustworthy. Their QoS data are excluded which also harms the prediction accuracy. As shown in Figure 3, both MAE and RMSE increase when the number of *Top-R* is larger than 10. However, RAP can still perform better than other approaches with slightly overrated *Top-R*. For instance, when *Top-R* is set as 15, it still has smaller MAE and RMSE than UIPCC.

In practice, it is difficult to determine the number of untrustworthy users. A straightforward approach is to initiate prediction with a small *Top-R* and adjust the setting for each prediction iteration to achieve better prediction accuracy. In the future, we will study automatic selection methods for *Top-R*.

E. Impact of Top-K

The parameter *Top-K* determines the number of similar user of services used to make prediction. To study the impact of parameter *Top-K* on prediction accuracy, we increase the value of *Top-K* from 5 to 50 by a factor of 5. The Matrix density is set as 10% and the percentage of untrustworthy users is set as 3% in this experiment.

According to experimental results shown in Figure 4, the prediction accuracy increases as the value of *Top-K* increases. Both curves begin to flatten with a larger value of *Top-K* since there are limited number of similar users and services in the dataset.
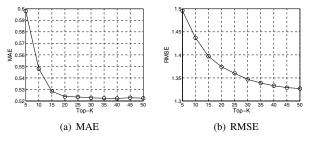

(a) MAE   (b) RMSE

Figure 4. **Impact of Top-K**

F. Impact of Matrix Density
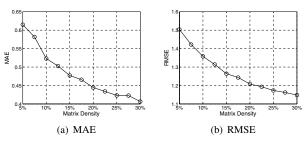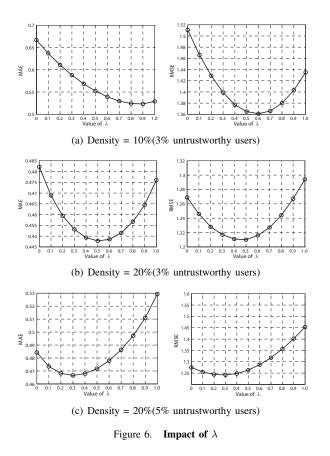

(a) MAE   (b) RMSE

Figure 5. **Impact of Matrix Density**

Matrix density refers to the percentage of unremoved entries in the user-service matrix. To study the impact of *Matrix density* on the prediction accuracy, we vary the density of matrix from 5% to 30% with a step value of 2.5%. In this experiment, the percentages of untrustworthy users are set as 3%, *Top-K* is set as 30 and *Top-R* is set as 10, respectively.

Figure 5 illustrates the experimental results. With the increase of *Matrix density*, both the MAE and RMSE of RAP decline, which indicates that denser available QoS values in the matrix can benefit the prediction accuracy, since more information can be mined.

G. Impact of λ

Parameter λ indicates how much RAP is relied on the information of similar users compared to the similar Web

46

(a) Density = 10%(3% untrustworthy users)

(b) Density = 20%(3% untrustworthy users)

(c) Density = 20%(5% untrustworthy users)

Figure 6.   **Impact of λ**

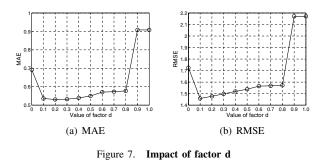services. To study the impact of parameter λ, we vary the value of λ from 0 to 1 with a step value of 0.1 under different matrix density settings (10% and 20%, respectively) and different percentage of random users (3% and 5%, respectively).

The experimental results show that:

- The value of parameter λ has significant impact on the prediction accuracy. An appropriate value of λ introduces proper combination of predicted value based on similar users and that based on similar services, which benefits the prediction accuracy.
- The optimal value of λ differs with different matrix density settings. For example, when the matrix density is 10%, the optimal value of λ is around 0.7, where both the MAE and RMSE is low, as shown in Figure 6(a). While the optimal value of λ is 0.5, when the matrix density is 20%, as shown in Figure 6(b). This observation indicates that when the matrix become denser, RAP relies more on similar Web services' information to make better prediction.
- The optimal value of λ also varies according to the percentage of untrustworthy users. As shown in Figure 6(b), the optimal value of λ is 0.5, when the percentage of random users is 3%. While in Figure 6(c) the optimal value of λ is 0.3 when the percentage of

untrustworthy users is 5%. This observation indicates that when there tends to be more unreliable users, RAP employs less information of similar users to prediction.

H. Impact of factor d

(a) MAE

(b) RMSE

Figure 7.   **Impact of factor d**

The factor d is used as a dumping factor in Eq. (1) to iteratively calculate the user reputation values. To study the impact of factor d on the prediction accuracy, we vary the value of d from 0 to 1 by a step of 0.1, where $d = 0$ means all users are taken as trustworthy. For other parameters, we set matrix density as 10% and the percentage of untrustworthy users as 3%, respectively.

The experimental results are illustrated in Figure 7. As shown in the figure, smaller MAE and RMSE are obtained when $d = 0.1$ compared with other settings. The prediction accuracy decreases as the value of d increases. When the value of d is larger than 0.9, the reputation calculation has convergent problems in this experiment, thus the MAE and RMSE value increase sharply. The results show that when $d = 0.1$, the approach has the best prediction performance, which coincides with the results presented in [21]. Therefore, we set $d = 0.1$ in this paper.

VI. CONCLUSION

User-contributed QoS values have great impacts on the prediction results of CF methods. To address the data credibility problem, we propose an reputation aware QoS value prediction approach (RAP) of Web Services in this paper. The main process of RAP is first to calculate user reputation based on the QoS values contributed by service users and then select trustworthy values according to user reputation. At last predictions are made by employing information of trustworthy similar users and services. Extensive experimental results show that RAP makes an significant prediction accuracy improvement.

Currently, the setting of parameter *Top-R* for RAP is highly related to the application domain and needs to be identified based on experiences. In the future, we will develop automatic identification methods for *Top-R* and more flexible reputation management models to improve the prediction performance. In addition, RAP calculates user reputation based on the QoS values contributed by all users.

However, certain QoS properties (such as response time, failure rate, etc.) can vary for users in different locations. Thus, the user reputation values may be affected by their locations which will further more affect the accuracy of QoS value prediction. In the future work, we will take the location information into consideration and cluster users before calculating reputation to improve the prediction accuracy.

REFERENCES

[1] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Proc. 7th Int'l Conf. Web Services (ICWS'09)*, 2009, pp. 437–444.

[2] J. E. Hadad, M. Manouvrier, and M. Rukoz, "Tqos: Transactional and qos-aware selection algorithm for automatic web service composition," *IEEE Trans. Services Computing*, vol. 3, no. 1, pp. 73–85, 2010.

[3] P. Wang, "Qos-aware web services selection with intuitionistic fuzzy set under consumer's vague perception," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4460–4466, 2009.

[4] J. M. Ko, C. O. Kim, and I.-H. Kwon, "Quality-of-service oriented web service composition algorithm and planning architecture," *Journal of Systems and Software*, vol. 81, no. 11, pp. 2079–2090, 2008.

[5] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative web service qos prediction with location-based regularization," in *Proc. 10th Int'l Conf. Web Services (ICWS'12)*, 2012, pp. 464–471.

[6] S.-Y. Hwang, H. Wang, J. Tang, and J. Srivastava, "A probabilistic approach to modeling and estimating the qos of web-services-based workflows," *Information Sciences*, vol. 177, no. 23, pp. 5484–5503, 2007.

[7] L. Terveen and W. Hill, "Beyond recommender systems: Helping people help each other," in *HCI in the New Millennium*. Addison-Wesley, 2001, pp. 487–509.

[8] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1–24, 2010.

[9] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Annual Conf. Uncertainty in Artificial Intelligence (UAI'98)*, 1998, pp. 43–52.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int'l Conf. World Wide Web (WWW'01)*, 2001, pp. 285–295.

[12] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. 29th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'06)*, 2006, pp. 501–508.

[13] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction forweb services via collaborative filtering," in *Proc. 5th Int'l Conf. Web Services (ICWS'07)*, 2007, pp. 439–446.

[14] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *Proc. 8th Int'l Conf. Web Services (ICWS'10)*, 2010, pp. 83–90.

[15] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Proc. 9th Int'l Conf. Web Services (ICWS'11)*, 2011, pp. 211–218.

[16] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proc. 8th Int'l Conf. Web Services (ICWS'10)*, 2010, pp. 9–16.

[17] M. Tang, Y. Jiang, J. Liu, and X. F. Liu, "Location-aware collaborative filtering for qos-based service recommendation," in *Proc. 10th Int'l Conf. Web Services (ICWS'12)*, 2012, pp. 202–209.

[18] L.-H. Vu, M. Hauswirth, and K. Aberer, "Qos-based service selection and ranking with trust and reputation management," in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, 2005, pp. 466–483.

[19] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[20] M. Medo and J. R. Wakeling, "The effect of discrete vs. continuous-valued ratings on reputation and ranking systems," *EPL (Europhysics Letters)*, vol. 91, no. 4, p. 48004, 2010.

[21] R.-H. Li, J. X. Yu, X. Huang, and H. Cheng, "Robust reputation-based ranking on bipartite rating networks," in *Proc. SDM*, 2012, pp. 612–623.

[22] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "Wsp: A network coordinate based web service positioning framework for response time prediction," in *Proc. 10th Int'l Conf. Web Services (ICWS'12)*, 2012, pp. 90–97.

[23] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," *IEEE Trans. Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.