

Predicting Quality of Cloud Services for Selection

Mingdong Tang, Wei Liang, Buqing Cao and Xiangyun Lin

School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
mdtang@hnust.edu.cn

Abstract

Predicting quality of services (QoS) is an important need when ranking cloud services for selection. QoS values of cloud services usually depend heavily on the user's and service's environments. Therefore, personalized QoS value prediction for cloud services is more desirable to users. Collaborative Filtering (CF) has recently been applied to personalized QoS prediction for services on the Web. However, seldom did they take the context information of service users and services into consideration. The following paper presents a CF-based method for predicting QoS values of cloud services. The method exploits not only the QoS information of users and services, but also one type of the most important context information of users and services, i.e., their geographic locations. Experiments conducted on a real dataset show that geographic location information is indeed helpful for improving the QoS prediction performance. The experimental results also demonstrate that the proposed method is significantly better than previous methods in prediction accuracy.

Keywords: *cloud services, service selection, QoS prediction, collaborative filtering*

1. Introduction

Cloud computing is Internet-based computing, whereby shared configurable resources (e.g., infrastructure, platform, and software) are provided to computers and other devices as services [1]. With the rising popularity of cloud computing, the past years have witnessed an exponential growth of cloud applications. Cloud applications, which are applications deployed in the cloud environment, are typically large-scale and complex. Similar to traditional component-based systems, cloud applications typically involve multiple cloud components communicating with each other over application programming interfaces. These cloud components, which were usually implemented as Web services, can be provided and deployed in the cloud by other companies [2]. Since there are a number of functionally equivalent services in the cloud, optimal service selection becomes important to service users. In this paper, service users refer to cloud applications or developers that use the cloud services.

In cloud service selection, quality of services (QoS) usually plays a very important role. QoS of cloud service is usually determined by a set of attributes such as throughput, delay, reliability and security, which are referred to as QoS attributes. There are at least two great challenges facing cloud service selection. Firstly, since there are multiple QoS attributes concerned, how to leverage different QoS attribute values and the user's requirements to measure quality of a service for the user is a challenging issue. Substantial works has been carried out for addressing this issue, and a variety of approaches have been proposed [3-5]. Therefore, we focus on the second challenging issue: how to acquire the QoS values of cloud services. In reality, QoS values of cloud services are usually incomplete or even missing. Not all service providers will publish accurate QoS information for their services. Some dishonest service providers may exaggerate their services' QoS to attract more users. Moreover, because service QoS is

highly dependent on users' and services' circumstances, the perceived QoS of the same service may be different from user to user [6]. To get the accurate QoS values of cloud services, users can conduct evaluations on the services by invoking them. However, this process is both time-consuming and resource-consuming, and thus is impractical when there are many candidate services for selection. More effective approaches to acquire cloud service QoS information are therefore required.

Inspired by the idea of user-collaboration at the Web 2.0 era, Collaborative Filtering (CF) techniques have been widely used to make prediction on missing values in recommender systems [7]. Recently, the CF techniques were also employed by some studies to predict QoS of service for the purpose of service recommendation [8]. The main idea is to identify groups of similar users and collect their useful QoS information on a service to predict its QoS value for an active user. However, Previous CF-based service recommendation methods have rarely taken into account the peculiar characteristics of cloud service QoS when making QoS predictions. QoS attributes of cloud services such as response time and throughput are highly dependent on the locations of service users and services as well as their underlying networks. However, this observation is usually ignored by the previous work.

This paper proposes an enhanced CF method for predicting cloud service QoS. The method incorporates a data smoothing procedure into CF via exploiting the geographic location information of service users. The method is composed of a user-based CF and a service-based CF. the user-based CF identifies a group of similar users based on their QoS data and uses their QoS data to make a QoS prediction for an active user and a target service. The service-base CF identifies a group of similar services based on their QoS data and uses their QoS data to make a QoS prediction for a active user and a target service. Finally, the method combines the above two QoS predictions to generate the final QoS prediction for the active user.

The remainder of this paper is organized as follows. Section 2 surveys the related work of this study. Section 3 gives an overview of the proposed method. Section 4 elaborates our QoS prediction method, including the user-based QoS prediction, the service-based QoS prediction and their integration. Section 5 presents the evaluation of our method and Section 6 concludes the paper.

2. Related Work

As many services with similar or equivalent functionalities rise, QoS has become a major concern in service selection. Previous related works are majorly focused on how to rank service candidates by leveraging the services' QoS and the users' QoS requirements. For example, Masri and Mahmoud [9] normalize values of different QoS attributes into a unique range, and then calculate the overall quality of services by leveraging the normalized QoS values and the user's preference on QoS attributes. Comuzziand and Pernici [10] use a price model to combine multiple QoS factors. The price model converts each QoS attribute of a service to a price, and combines all prices together to measure the service's QoS. Yau and Yin [11] proposes a service selection method that selects the service that best satisfies users QoS requirements, instead of the service with the best QoS. Liu, Fletcher, and Tang [12] use fuzzy operators to aggregate individual QoS values, instead of simple summation or average operators. However, almost all service selection methods are based on the assumption that QoS values of service candidates are already acquired and are same to all users. The assumption is not realistic in practice. Service QoS information is usually incomplete or even missing, as stated in the introduction, and there should be no service whose QoS is the same to all users.

Inspired by the successful application of CF in various recommender systems, recently, several works have applied CF to predict QoS of Web services for a user in a personalized way [8, 13]. Based on the QoS information recorded from a set of users invoking a set of

services, CF-based methods firstly identify a subset of users similar to an active user, and then aggregate the QoS information of these users on a target service to make a QoS prediction for the active user and the target service. To further improve the accuracy of QoS prediction, a few enhanced CF methods that explicitly incorporate context information of users and services are proposed. For example, work [14] exploits the IP addresses of users for improving the process of identifying similar users in CF, based on the assumption that users with similar IP addresses are likely to be similar on their QoS information. Work [15] assumes that users within the same Autonomous System (AS) or country are likely to be similar, and thus exploits such location information to improve similar user or service search. Work [16] uses a virtual Internet coordinate system to locate users and improve the similar user search results.

Recently, some emerging works [17-19] have employed the Matrix Factorization (MF) techniques to improve the accuracy of QoS prediction. However, their methods are essentially complicated and time-consuming. Different from previous work, this work incorporates a data smoothing procedure into CF via exploiting the geographic location information of service users.

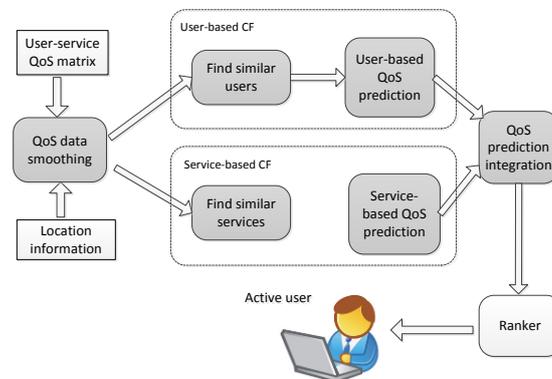


Figure 1. Framework of Our Proposed QoS Prediction Method for Cloud Service Selection

3. Overview of the Method

Fig. 1 presents an overall architecture for the proposed cloud service QoS prediction method. Suppose that the active user's interest is already known, and a list of cloud services that matching his/her functional interest has been identified as candidates for his/her selection. The components in the architecture are described as follows.

The user-service QoS matrix is used to save the QoS information of cloud services after they are used by different service users. The location information contains locations of both service users and services. The QoS data smoothing component is used to address the data sparsity of the QoS matrix, by exploiting the location information.

The user-based CF employs the QoS matrix to find similar users for an active user, and then uses the similar users' QoS information to make QoS predictions for the active user and the target service. In a similar manner, the service-based CF employs the QoS matrix to find similar services for a target service, and then uses the similar services' QoS information to make QoS predictions for the active user and the target service. The two kinds of QoS predictions are finally aggregated to make more accurate QoS predictions. After predicting all missing QoS values for the active user and every service candidate, a ranker is used to ranking cloud services based on their QoS values for the active user to make service selection decision.

4. The QoS Prediction Algorithm

Given a cloud service selection system consisting of M service users and N cloud services, the relation between users and services is denoted by an $M \times N$ matrix, namely, the user-service QoS matrix. Every entry in this matrix $r_{u,i}$ represents the a vector of QoS values (e.g., response time, throughput, etc.), that is observed by the service user u on the cloud service i . If user u did not invoke the service i before, then $r_{u,i} = \text{null}$. In the following, we introduce how to predict the value of $r_{u,i}$ using CF techniques.

4.1. QoS Data Smoothing

Data sparsity is a fundamental problem for CF. Due to the data sparsity of the user-item matrix, there are insufficient data for the CF to measure similarity and identify similar neighbors accurately, and thus the accuracy of predicted results will be significantly lowered. To improve the density of the user-item matrix and address the data sparsity problem, several works [20, 21] have employed data smoothing techniques to fill the missing values in the user-item matrix. Those methods usually cluster all users into different groups based on the calculated user similarity via employing K-means algorithms. Then, for every missing value in the user-item matrix, they replace it with the average value of the item to the group where the user belongs. The above data smoothing methods, however, have two weaknesses. Firstly, when the user-item matrix is very sparse, user similarity computation could be inaccurate and thus leading to ineffective user clustering. Secondly, the clustering algorithm has to be re-performed whenever the data in the user-item matrix change, thus it is likely to be resource-consuming. In the following, we propose a novel QoS data smoothing algorithm by employing the geographic location information of users and services.

In cloud environments, we observe that users near to each other in location usually share similar interests and perceive similar QoS on the same service. Suppose the geographic locations of two users a and u are known, the distance between them is calculated with

$$d(a, u) = 111.199 * \sqrt{(A_a - A_u)^2 + (L_a - L_u)^2 \times \cos^2\left(\frac{A_a + A_u}{2}\right)} \quad (1)$$

where $A_a \in (-180, 180]$ represents the altitude in location of a , and $L_a \in (-180, 180]$ indicates the latitude in location of a . We define the users with a small distance to a as its local neighborhood, i.e.,

$$N(a) = \{u \mid d(a, u) < D\} \quad (2)$$

where D is preset distance threshold. Based on the above definitions, suppose $r_{u,i}$ is an item in the user-service matrix and the user u has not invoked service i yet, $r_{u,i}$ is calculated with

$$r_{u,i} = \bar{r}_u + \Delta r_{N(u),i} \quad (3)$$

where \bar{r}_u is the vector of average QoS values of services invoked by u , and $\Delta r_{N(u),i}$ is average QoS deviations for all users in $N(u)$ to the service i , which is defined as:

$$\Delta r_{N(u),i} = \sum_{u \in N(u)} (r_{u,i} - \bar{r}_u) / |N(u)| \quad (4)$$

where $|N(u)|$ is the number of users in $N(u)$.

4.2. Similarity Computation

Pearson Correlation Coefficient (PCC) has been widely used in many recommendation systems for similarity computation, since it can be easily implemented and can achieve high accuracy. In user-based collaborative filtering, the formula for computing the similarity between two users a and u can be written as:

$$sim(a, u) = \frac{\sum_{s \in I} (r_{a,s} - \bar{r}_a)(r_{u,s} - \bar{r}_u)}{\sqrt{\sum_{s \in I} (r_{a,s} - \bar{r}_a)^2} \sqrt{\sum_{s \in I} (r_{u,s} - \bar{r}_u)^2}} \quad (5)$$

where $I = I_a \cap I_u$ is the subset of services that are commonly invoked by user a and user u ; $r_{a,s}$ represents the vector of QoS values of service s observed by user a ; and \bar{r}_a represents the vector of average QoS values of user a . From the above definition, the user similarity, $sim(a, u)$, is in the interval of $[-1, 1]$, with a larger value indicating that user a and user u are more similar.

PCC may overestimate the similarities of service users who are actually not similar but happen to have similar QoS experiences on just a few co-invoked services [22]. To address this problem, we modify the user similar computational formula by introducing a weighted factor to reduce the influence of a small number of co-invoked services. The adjusted user similar computational formula is

$$sim'(a, u) = \frac{\min(|I|, \gamma_1)}{\gamma_1} \times sim(a, u) \quad (6)$$

where $|I|$ indicates the number of services invoked by user a and user u in common, γ_1 is a preset threshold for adjusting the user similarity, mainly determined by the sparsity of the user-service QoS matrix. Generally, if the matrix is very sparse, the number of services commonly used by two users is relatively small; the threshold should be set smaller. If $|I|$ is bigger than the threshold γ_1 , the user similarity will not be adjusted.

In service-based collaborative filtering, PCC can also be employed to compute similarity between two services i and j . The computational formula is as Formula (3):

$$sim(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{s \in I} (r_{u,j} - \bar{r}_j)^2}} \quad (7)$$

where $U = U_i \cap U_j$ is the subset of users who invoke service i and j ; $r_{u,i}$ represents the vector of QoS values of service i observed by user u ; and \bar{r}_i represents the vector of average QoS values of service i . The service similarity, $sim(i, j)$, is also in the interval of $[-1, 1]$.

Just like the user-based collaborative filtering method, an adjusted PCC for the similarity computation of different services is defined as:

$$sim'(i, j) = \frac{\min(|U|, \gamma_2)}{\gamma_2} \times sim(i, j) \quad (8)$$

where γ_2 is a preset threshold for adjusting the service similarity. If $|U|$ is bigger than the threshold γ_2 , the service similarity will not be adjusted.

4.3. Similar Neighbors Selection

Similar neighbor selection is a very important step of CF. In user-based CF, a subset of users similar to the active user is identified, while in item-based CF, a subset of services similar to the target service is identified. In this section, we present our similar neighbor selection strategies for both the active user and the target service.

From Section 4.2, it can be inferred that a similar user to the active user should have a user similarity whose value is at least greater than 0. To remove the effect of dissimilar users, we define the similar neighbors of an active user a as follows:

$$N'(a) = \{u \mid sim'(a, u) > \eta\} \quad (9)$$

where $\eta \geq 0$ is a preset similarity threshold for users. In a similar manner, the similar neighbors of a target service i are defined as follows:

$$N'(i) = \{j \mid sim'(i, j) > \theta\} \quad (10)$$

where $\theta \geq 0$ is a preset similarity threshold for services.

4.4. QoS Prediction

In the user-based CF, Formula (11) is used to predict missing QoS values of service i to the active user a .

$$uR_{a,i} = \bar{r}_a + \frac{\sum_{u \in N'(a)} sim'(a, u)(r_{u,i} - \bar{r}_u)}{\sum_{u \in N'(a)} sim'(a, u)} \quad (11)$$

where \bar{r}_a is the vector of average QoS values of services invoked by a and $N'(a)$ is the set of similar users to user a . Similarly, the service-based method uses Formula (12) to predict the missing QoS values of service i to the active user.

$$iR_{a,i} = \bar{r}_i + \frac{\sum_{j \in N'(i)} sim'(i, j)(r_{a,i} - \bar{r}_j)}{\sum_{j \in N'(i)} sim'(i, j)} \quad (12)$$

where \bar{r}_i is the vector of average QoS values of service i , and $N'(i)$ the set of similar services to service i .

Since these two predicted values may have different prediction performance, we use a tunable parameter λ to balance these two predicted values. We combine the two methods by using Formula (13):

$$R_{a,i} = \lambda \times uR_{a,i} + (1 - \lambda) \times iR_{a,i} \quad (13)$$

By the above mechanism, the parameter λ determines how much the predicted results rely on the user-based method or the service-based method. The parameter λ makes our method adaptable to different environments

5. Evaluation

During our experiments, we adopted a real-world service dataset, WSDream dataset 2 [6], published in *www.wsdream.com*. This dataset contained the QoS

records of service invocations on 5825 services from 339 users. The dataset can be transformed into a user-service matrix. Each item of the user-service matrix is a pair of values: response time (also called Round Trip Time, RTT) and throughput (TP). Therefore, the original user-service matrix can be decomposed into two simpler matrices: RTT matrix and TP matrix. We used either the RTT matrix or the TP matrix to compute both the user and the service similarities. This dataset also contained the geographic location information of both users and services. Our experiments evaluate the proposed QoS prediction method from the following three aspects:

1) What is the relation between users' (or services') locality and similarity, i.e., are users close to each other in location are also similar to each other in QoS? This is evaluated in Experiment 1.

2) How does our method compare with other CF-based QoS prediction methods? This is evaluated in Experiment 2.

3) How does parameter λ influence the prediction accuracy? This is evaluated in Experiment 3.

Our experiments are developed by using Matlab 7.0 and performed on a HP desktop computer with configuration as: Intel Core i3 3.20GHz CPU, 2GB RAM, and Windows 7 operating system.

5.1. Experiment 1

To evaluate the relation between a user's local neighborhood and similar neighborhood, we conduct the following experiment: For each targeted user, we calculate the distance between it and all the other users using Eq. (1), we also calculate the similarities between it and all the other users by employing the RTT matrix (or the TP matrix); then we ranking all the other users in descending order of distance to the targeted user; finally, for all rankings we calculate the average similarity of the users at position p ($p=1,2,\dots,338$), and verify whether the average similarity values of users with higher positions are greater than those with lower positions or not. If the average similarity values of users with higher positions are significantly greater than those with lower positions, it indicates that there is a relation between locality and similarity of users, and the more difference, the stronger is the relation.

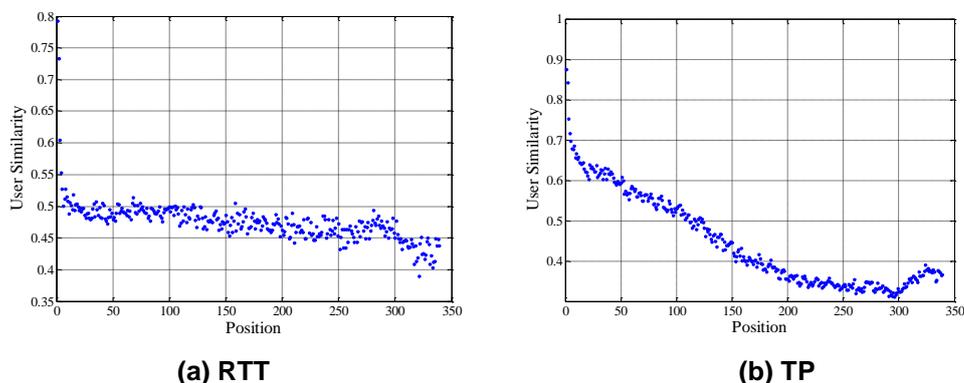


Figure 2. The Average User Similarities Calculated For Each Position in the Distance-Based Ranking of Users

Fig. 2 reports the results of the above experiment. In specific, Fig. 2(a) shows the results produced by employing the RTT matrix, and Fig. 2(b) shows the results produced by employing the TP matrix. From the experimental results we can see, the average user similarities calculated decrease significantly when the distances of users increase. In particular, the top-ranked users have much larger similarity values than the other users. For example, the top 10 ranked users are observed to have an average similarity about 0.58 when employing the RTT matrix to compute user similarity, and the top 10 ranked users has an average similarity about 0.73 when employing the TP matrix to compute user similarity. The above experimental results validate that there is a strong relation between user locality and user similarity, and users closer to a targeted user in location are highly likely to be more similar to the targeted user in QoS.

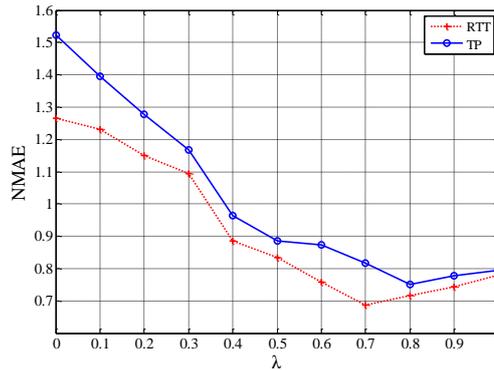


Figure 3. Impacts of Λ on the Prediction Accuracy

5.2. Experiment 2

This experiment evaluated the prediction accuracy of our QoS prediction method, and compared it with other existing QoS prediction methods. In this experiment, the 339 service users are divided into two groups: 300 randomly selected training users and the rest as test (active) users. The RTT matrix is divided into the RTT-training-matrix and the RTT-test-matrix, and so is the TP matrix. In order to simulate the real-world situation, we randomly remove entries of the training matrices and the test matrices to reduce the data density to 20%. The removed entries of the test matrices are used for evaluating the prediction quality. We set $\lambda = 0.7$ and $\lambda = 0.8$ for RTT and TP prediction respectively, to make our method achieve the best performance. The values of the other parameters are set as $D = 100$, $\gamma_1 = \gamma_2 = 5$, $\eta = \theta = 0.2$.

Mean Absolute Error (MAE) is often used in collaborative filtering methods to measure the prediction accuracy. MAE is defined as:

$$MAE = \frac{\sum_{i,j} |r_{a,i} - R_{a,i}|}{N} \quad (14)$$

where $r_{a,i}$ is the actual QoS values of Web service i observed by service user a , and $R_{a,i}$ is the predicted QoS values, and N denotes the number of predicted values. Because different QoS attributes of Web services have distinct value ranges, we also use the Normalized Mean Absolute Error (NMAE) metric to measure the prediction accuracy. NMAE is defined as:

$$NMAE = \frac{MAE}{\sum_{a,i} r_{a,i} / N} \quad (15)$$

And smaller NMAE values represent higher prediction accuracy.

We compare our method with some other well-known prediction methods: user-based prediction method using PCC (UPCC) [23], item-based prediction method using PCC (IPCC) [24], hybrid user and item-based prediction method using PCC (UIPCC) [8,25], and a location-aware CF algorithm, LACF [15], to evaluate its prediction performance. Formula (5) and Formula (7) are used for computing the UPCC and IPCC respectively. The results are reported in Table 1, where CloudPred represents our proposed QoS prediction method. From this table, we can see that our method had significantly smaller MAE and NMAE values than the other methods, indicating that our proposed method has a better prediction performance.

Table 1. Performance Comparison between Our Method and other QoS Prediction Methods

Metrics	UPC C	IPC C	UIPCC	LAC F	CloudPre d
RTT_MAE	0.72 8	0.95 7	0.638	0.54 8	0.533
RTT_NMAE	1.02 5	1.38 5	0.968	0.71 9	0.708
TP_MAE	41.5 7	90.7 5	40.21	34.0 5	33.71
TP_NMAE	1.13 2	2.16 6	0.983	0.79 3	0.764

5.3. Experiment 3

Different datasets may have different data correlation characteristics. Parameter λ determines how much the proposed QoS prediction method relies on either the user-based prediction or the item-based prediction. Parameter λ makes the prediction feasible for various environments. Given $D = 100$, $\gamma_1 = \gamma_2 = 5$, $\eta = \theta = 0.2$, we vary the value of λ , from 0 to 1, with each step of 0.1, and study its influence on our proposed QoS prediction method. Fig. 3 reports the influence of λ on the prediction accuracy of both RTT and TP. Our method achieved the best accuracy for RTT prediction when $\lambda = 0.7$ and for TP prediction when $\lambda = 0.8$.

6. Conclusion

This paper presented a CF-based QoS prediction method for cloud service selection. Based on the observation that the QoS of cloud services highly depends on the locations of service users, and users that are close in location tend to share similar QoS experiences, the method exploits the geographic location information of users and incorporate it into a data smoothing procedure for the user-service matrix, which can significantly improve its data density and thus is helpful for improving prediction quality. The method then employs both the user-based CF and the item-based CF to make QoS predictions, and integrate them for more accurate prediction. Experimental results not only verify the strong relation between locality and similarity of both users and services, but also show the better prediction performance of our proposed method compared with the other four QoS prediction methods.

Acknowledgements

The work described in this paper was supported by the National Natural Science Foundation of China under grant No. 61100054, 61202462 and 61402168, and Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2013-1-21).

References

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the clouds, A Berkeley view of cloud computing", Technical Report, EECS-2009-28, University of California, Berkeley, (2009).
- [2] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu and J. Wang, "QoS Ranking Prediction for Cloud Services", IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, (2013), pp. 1213-1222.
- [3] P. Adamopoulou, E. Sakkopoulos, S. Sioutas, A. Tsakalidis and Y. Manolopoulos, "Personalized Selection of Web Services for Mobile Environments, The m-scroutz solution", ACM-SIGAPP MEDES, (2009), pp. 254-290.
- [4] S. Wang, Q. Sun and F. Yang, "Towards Web Service selection based on QoS estimation", International Journal of Web and Grid Services, vol. 6, no. 4, (2010), pp. 424-443.
- [5] S. Wang, Z. Zheng, Q. Sun, H. Zou and F. Yang, "Reliable web service selection via QoS uncertainty computing", International Journal of Web and Grid Services, vol. 7, no. 4, (2011), pp. 410-426.
- [6] Z. Zheng, Y. Zhang and M. R. Lyu, "Investigating QoS of Real-World Web Services", IEEE Transactions on Services Computing, vol. 7, no. 1, (2014), pp. 32-39.
- [7] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques", Advances in Artificial Intelligence, (2009).
- [8] Z. Zheng, H. Ma, M. R. Lyu and I. King, "WSRec, a collaborative filtering based web service recommendation system", In ICWS 2009, (2009), pp. 437-444.
- [9] E. A. Masri and Q. H. Mahmoud, "QoS-based Discovery and Ranking of Web Services", in Proceedings of the International conference on Computer Communications and Networks (ICCCN), (2007), pp. 529-534.
- [10] M. Comuzzi and B. Pernici, "A Framework for QoS-Based Web Service Contracting", ACM Transactions on The Web, vol. 3, no. 3, Article 10, (2009).
- [11] S. S. Yau and Y. Yin, "QoS-based Service Ranking and Selection for Service-based Systems", in Proceedings of the International conference on Services Computing (SCC), (2011), pp. 156-863.
- [12] X. F. Liu, K. K. Fletcher and M. Tang, "Service Selection based on Personalized Preference and Trade-Offs among QoS Factors and Price", The 1st IEEE International Conference on Services Economics, (2012) June, pp. 32-39, research track, Hawaii, USA.
- [13] Y. Jiang, J. Liu, M. Tang, and X. F. Liu, "An effective Web service recommendation method based on personalized collaborative filtering", In ICWS, (2011), pp. 211-218.
- [14] X. Chen, X. Liu, Z. Huang and H. Sun, "RegionKNN, A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation", in Proceedings of International Conference on Web Services, Miami, FL, USA, (2010), pp. 9-16.
- [15] M. Tang, Y. Jiang, J. Liu, X. F. Liu, "Location-Aware Collaborative Filtering for QoS-Based Service Recommendation", in Proc. 10th International Conference on Web Services, Hawaii, USA, (2012), pp. 202-209.
- [16] J. Zhu, Z. Zheng and M. R. Lyu, "WSP, A Network Coordinate based Web Service Positioning Framework for Response Time Prediction", in Proc. 10th IEEE International Conference on Web Services, Research Track, Hawaii, US, (2012), pp. 90-97.
- [17] W. Lo, J. Yin, S. Deng, Y. Li and Z. Wu, "An Extended Matrix Factorization Approach for QoS Prediction in Service Selection", in Proceedings of International Conference on Services Computing, Hawaii, USA, (2012), pp. 162-169.
- [18] Z. Zheng, H. Ma, M. R. Lyu and I. King, "Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization", IEEE T. Services Computing, vol. 6, no. 3, (2013), pp. 289-299.
- [19] Q. Yu, Z. Zheng and H. Wang, "Trace Norm Regularized Matrix Factorization for Service Recommendation", ICWS 2013, (2013), pp. 34-41.
- [20] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing", In SIGIR (2005).
- [21] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. Zhou and Z. Wu, "Predicting Quality of Service for Selection by Neighborhood-based Collaborative Filtering", IEEE Transactions on System, Man, and Cybernetics, Part A [TSMC], vol. 43, no. 2, (2013).
- [22] M. R. McLaughlin and J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience", In SIGIR (2004).
- [23] J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", in Proc. of the UAI, (1998), pp. 43-52.
- [24] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, "GroupLens, An open architecture for collaborative filtering of netnews", in Proc. of ACM Conference on Computer Supported Cooperative Work, (1994).
- [25] Z. Zheng, H. Ma, M. R. Lyu and I. King, "QoS-Aware Web Service Recommendation by Collaborative Filtering", IEEE Transactions on Services Computing, vol. 4, no. 2, (2011), pp. 140-152.

Authors



Mingdong Tang, is an associate professor in the School of Computer Science and Engineering at Hunan University of Science and Technology, Xiangtan, China. He received his Ph.D. degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2010. His current research interests include services computing, cloud computing, and social networks. He is a member of IEEE and ACM. Email: mdtang@hnust.edu.cn.



Wei Liang, is an associate professor in the School of Computer Science and Engineering at Hunan University of Science and Technology, Xiangtan, China. He received his Ph.D. degree in Computer Science from Hunan University, China, in 2013. His current research interests include distributed computing and wireless sensor networks. Email: wliang@hnust.edu.cn.



Buqing Cao, is an associate professor in the School of Computer Science and Engineering at Hunan University of Science and Technology, Xiangtan, China. He received his Ph.D. degree in Computer Science from Hunan University, China, in 2011. His current research interests include services computing and cloud computing. Email: bucao@hnust.edu.cn



Xiangyun Lin, is a master student in the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. He received his B.S. degree in Computer Engineering from Hunan University of Humanities, Science and Technology, China, in 2011. His research interests include service recommendation and selection. Email: linxiangyun0911@gmail.com

