2017 IEEE 14th International Conference on Services Computing

Temporal Influences-Aware Collaborative Filtering for QoS-Based Service Recommendation

Jinglin Li, Jie Wang, Qibo Sun, Ao Zhou
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications, Beijing, China
{ jlli, jayw, qbsun, aozhou }@bupt.edu.cn

***Abstract*—As service computing becomes increasingly prevalent, the number of web services grows rapidly. It becomes very important to recommend suitable, personalized web services to users. Collaborative Filtering based on Quality of Service (QoS) has been widely used for service recommendation, and variety of factors such as location, environment are taken into account to improve the accuracy of recommendation. However, temporal influences, which is one of key factors affecting the QoS, are not fully considered by the investigators. In this paper, we propose a novel temporal influences-aware collaborative filtering method which designs an enhanced temporal influences-aware similarity measurement to predict QoS values. Finally, we conduct a series of experiments to evaluate the effectiveness of our method, and results show that our method outperforms other state-of-the-art methods.**

Keywords-service recommendation; temporal influences-aware; QoS; collaborative filtering

I. INTRODUCTION

Service computing as a promising computing paradigm for software engineering and distributed computing has gotten lots of attention in recent years. An application always combines multiple services to form an integrated service to satisfy various needs of users. The quality of the invoked web services greatly influences the quality of integrated service. Therefore, to build a high quality composited service application, it is very important to identify and select an appropriate web service.

With the exponential growth of web services, there are many web services with similar or identical functionalities, but different QoS. Since QoS describes the non-functional characteristics of web services including availability, through put, response time etc. [1], it is employed as an important factor at the time of analogous service recommendation. QoS based service recommendation can come to the aid of service users to select optimal QoS performance services that meet needs. However, it is a difficult task due to:

- Service providers rarely deliver the declared QoS values, since web services are loosely-coupled, located in different location and probably subject to different development, verification, as well as testing process [2].
- QoS values are highly related to invocation time, for the reason that service status (e.g., number of clients, workload etc.) and the network environment (e.g. network latency, bandwidth etc.) change dynamically over time, and have a great influence on QoS performance.

Recently, Neighborhood-based Collaborative Filtering (CF) has become the most prevalent method for personalized service QoS prediction. CF includes user-based, item-based and hybrid approaches, the user-based method employs historical QoS ratings from similar users to predict the missing QoS values, while the item-based method utilizes historical QoS experience from similar services to make QoS prediction. The hybrid method combines both user-based and item-based method to achieve a better accuracy. Furthermore, not only historical QoS information but also context information is considered for better accuracy. The most common discussed context factor is location, but there are few researchers considering the influence of time information accurately enough, which is one of key factors affecting the QoS of web services.

To further enhance the performance of web service recommendation, this paper proposes a novel temporal influences-aware Collaborative Filtering method for service QoS prediction. The key contributions of our work are as follows:

1) In order to get high quality recommendation, temporal dynamic characteristics of QoS values is adequately considered in our similarity measurement algorithm.
2) Experiments are conducted to evaluate the prediction accuracy of our temporal influences-aware CF method.

The rest of this paper is organized as follows. Section II reviews related work. Section III describes our temporal influences-aware Collaborative Filtering algorithm in detail. Section IV presents experiments for evaluating our method and comparisons with other CF algorithms. Section V concludes this paper and outlines our future work.

II. RELATED WORK

Web service recommendation aims to recommend a qualified web service with optimal QoS values to a service user based on invocation records of web services. Neighborhood-based CF methods which includes three main categories, user-based, item-based and hybrid are widely employed to the service QoS prediction. Zheng et al proposes a hybrid CF method combining user-based CF and

2474-2473/17 $31.00 © 2017 IEEE
DOI 10.1109/SCC.2017.67

item-based CF to enhance the prediction accuracy and conduct a series of large-scale experiments to evaluate their method based on a real web dataset [3]. In addition, investigators incorporate variety of contextual factors such as location, environment into basic CF approaches. For example, location factor is one of most widely discussed context information [4]. The geographically close users or services are deemed to be more similar, and location information of users or services is integrated with service invocation history to predict QoS values. Chen et al. group service users into a hierarchy of regions based on users' QoS experiences and locations and only considers users that belongs to the region of the target user to make QoS prediction [5].

Another widely used context information is time information. Yan hu et al. think that more temporally close or recent QoS experience on a same service from two different users contributes more to the user similarity measurement. And a well-designed exponential time decay function is used to weight the similarity. And Xiaoliang Fan et al. enhance the time decay function [6]. In addition, [2] divides service invocation time into several fixed short time intervals and only consider historical QoS records in time intervals adjacent to target time point. However, dividing invocation time into fixed time interval is not accurate enough in QoS prediction. We will address the problem in this paper.

III. PROPOSED METHOD

A. Notations and Definitions

To formalize the research problem involved in the paper, the following are important notations:

- $U = \{u_1, u_2, ..., u_n\}$ is a set of service users, where n is the total number of users.
- $S = \{s_1, s_2, ..., s_m\}$ is a set of services, where m is the total number of services in the recommendation system.
- $M = \{q_{u_i, s_j, ts} \mid 1 \le i \le n, 1 \le j \le m\}$ is the user-service-time matrix recording services invoked records, where $q_{u_i, s_j, ts}$ denotes a vector of QoS attribute values $\{c_1, c_2, ..., c_l\}$ obtained from user u_i invoking services s_j at time interval ts.
- $TS = \{ts_1, ts_2, ..., ts_r\}$ is a set of time interval, where r is total number of time interval. For example, one day can be partitioned to $r = 24$ time intervals, and every time interval lasts 1 hour.
- d_{min}, d_{max} and d_{avg} represent the minimum, the maximum and the average distance between QoS classes respectively.

B. Algorithm overviews

The overview description of our temporal influences-aware QoS based recommendation method will be presented in this section. Suppose u_i is the target user and s_j is the candidate service, our problem is to predict the missing QoS values of s_j for u_i at target time interval ts_{target} based on the historical QoS information in M, and recommend the service which not only meet the required functionalities but also has optimal QoS values to u_i at time interval ts_{target}. The following are major sub-procedures:

1) Normalize QoS data and cluster QoS values using k-means algorithm, and then calculate distance d_{min}, d_{max} and d_{avg} between QoS classes.
2) Find a time interval set ts_D iteratively satisfying that the average QoS change amount of s_j does not exceed the threshold value d where d can be d_{min}, d_{max} and d_{avg}. And calculate the similarity between candidate service s_j and s_{j1} at the time interval set ts_D where $s_{j1} \in S - \{s_j\}$. Select top k services most similar to s_j, denoted by S_{topK}.
3) Calculate the user similarity based on S_{topK} obtained by step 2, and get top k neighbors of the target user, denoted by U_{topK}.
4) Make QoS prediction for target user u_i based on U_{topK}.

C. Data normalization and clustering

The value of different QoS properties can vary in different scales, so we need to normalize the historical QoS data. Normalization can adjust properties measured on different scales to a common scale. Here Gaussian normalization is employed to normalize our QoS data in consideration of most properties are normally distributed.

$$c_i = \frac{c_i - \mu_i}{\sigma_i} \tag{1}$$

where μ_i is mean of c_i and σ_i is standard deviation of c_i. After normalization, we apply the k-means algorithm to classify the QoS records into k clusters $\{L_1, L_2, ..., L_k\}$ $(k > 0)$. The distance between two QoS values is defined by the following:

$$dis(q_1, q_2) = \sqrt{\sum_i^l (q_1(c_i) - q_2(c_i))^2} \tag{2}$$

After clustering, the distances between each centriole are calculated. d_{min}, d_{max} and d_{avg} are defined as follows:

$$\begin{aligned} d_{min} &= min(d_{i,j}) \\ d_{max} &= max(d_{i,j}) \\ d_{avg} &= average(d_{i,j}) \end{aligned} \tag{3}$$

where $d_{i,j}$ represents the distance between centriole i and centriole j $(i \ne j,\ 1 \le i, j \le k)$.

D. Service similarity calculation

Service similarity measurement is one of key steps of neighborhood-based CF. In this step, we fully consider the influence of service invocation time, which has a significant impact on prediction accuracy because of the time-varying network environment and service status. There is an intuitive principle that more close QoS experiences on two different services from the same user at the same timespan contribute more to service similarity measurement.

Algorithm 1: Optimal time timespan searching algorithm

Input: the historical QoS records M of s_j , the target recommendation time ts_{target} , the threshold d

Output: the time interval set ts_D

initial $l = 0$, $h = 0$, $t =$ ts_{target} , $delta = 0$
calculating the average QoS values of s_j in t
while $delta <= d$ **do**
    $l = l + 1$
    $delta$ = calculating the change amount of QoS values at $ts_{target} - l * \Delta t$ against average QoS values at ts_{target}
end
$delta = 0$
while $delta <= d$ **do**
    $h = h + 1$
    $delta$ = calculating the change amount of QoS values at $ts_{target} + h * \Delta t$ against average QoS values at ts_{target}
end
return $t_D = [ts_{target} - l * \Delta t,\ ts_{target} + h * \Delta t]$

However, a too long timespan may neglect the influence of time factor and a short timespan may result in fluctuations in the prediction accuracy. A novel algorithm is proposed to find an appropriate timespan in this paper, which iteratively extends time interval from the target recommendation time and makes the change amount of the average QoS in this timespan does not exceed a threshold value d . The algorithm is described in Algorithm 1.

The service similarity is calculated based on the found time interval set ts_D as follows:

$$sim(s_j, s_{j1})_{ts_{target}} = \frac{\sum_{ts \in ts_D} sim(s_j, s_{j1})_{ts}}{|ts_D|} \tag{4}$$

where $ts_D = [ts_{target} - l * \Delta t,\ ts_{target} + h * \Delta t]$, Δt is the minimum time interval unit. The similarity measurement is based on PCC, and the base measurement is defined as:

$$sim^*(s_j, s_{j1})_{ts} = \frac{\sum_{u \in U_{s_j, s_{j1}}} (q_{u,s_j,t} - \overline{q_{s_j,ts}})(q_{u,s_{j1},t_1} - \overline{q_{s_{j1},ts}})}{\sqrt{\sum_{u \in U_{s_j, s_{j1}}} (q_{u,s_j,t} - \overline{q_{s_j,ts}})^2}\sqrt{\sum_{u \in U_{s_j, s_{j1}}} (q_{u,s_{j1},t_1} - \overline{q_{s_{j1},ts}})^2}} \tag{5}$$

where $U_{s_j, s_{j1}}$ denotes users that invoked both s_j and s_{j1} at the time interval ts . Furthermore, we apply a weighted factor to eliminate the influence of services which are really not similar but just happen to have similar QoS experiences on a few common users. Finally, the service similarity measurement $sim(s_j, s_{j1})_{ts}$ is defined as:

$$sim(s_j, s_{j1})_{ts} = \frac{2|U_{s_j, s_{j1}}|}{|U_{s_j}| + |U_{s_{j1}}|} sim^*(s_j, s_{j1})_{ts} \tag{6}$$

where $|U_{s_j}|$ and $|U_{s_{j1}}|$ are the numbers of users invoking s_j and s_{j1} respectively. On the basis of service similarity measurement method, we select top k services most similar to s_j , S_{topK} .

E. User similarity calculation

The similarity of users is measured based on S_{topK} , and only services which belong to S_{topK} are considered. Formula below is used to calculate the similarity between users:

$$sim(u_i, u_{i1})_{ts_{target}} = \frac{\sum_{ts \in ts_D} sim(u_i, u_{i1})_{ts}}{|ts_D|} \tag{7}$$

where ts_D is calculated by Algorithm 2. And the definition of $sim(u_i, u_{i1})_{ts}$ is a pairing of $sim(s_j, s_{j1})_{ts}$:

$$sim(u_i, u_{i1})_{ts} = \frac{2|S_{u_i, u_{i1}}|}{|S_{u_i}| + |S_{u_{i1}}|} sim^*(u_i, u_{i1})_{ts} \tag{8}$$

Based on (7) and (8), we can get the U_{topK} , which denotes the top k neighbors of u_i .

F. QoS prediciton

After calculation of U_{topK} , a weighted aggregation of perceptions on the target service of every user in U_{topK} is used to make prediction for the final QoS values, defined as follows:

$$q_{u_i, s_j, ts_{target}} = \frac{\sum_{u \in U_{topK}} sim(u_i, u)_{ts_{target}} \times q_{u, s_j, ts_{target}}}{\sum_{u \in U_{topK}} sim(u_i, u)_{ts_{target}}} \tag{9}$$

IV. EXPERIMENTS

In this section, experiments are conducted to evaluate our temporal influences-aware web service QoS prediction method.

A. Setup

The dataset used in our experiments is from the well-known real-world web service Dataset WSDREAM dataset 2 [6], which contains QoS records of service invocations on 4500 web services from 142 users at 64 continuous time intervals. So, the dataset is presented by a $64 \times 4500 \times 64$ user-service-time matrix in memory. Besides only the Response Time data is employed in our experiments.

To evaluate the performance of our method, we compare our temporal influences-aware collaborative filtering with dynamic time intervals (TACF-DT) method with two other service recommendation methods, which are the well-known hybrid CF method WSpred [7] and the time-aware collaborative filtering method with the searching space parameter d=2 denoted by $TACF_{d=2}$ [2].

B. Evaluation

To assess the prediction accuracy, Mean Absolute Error (MAE) is used. MAE is a measurement of the deviation of predictions from their actual QoS values, which can be defined as:

$$MAE = \frac{1}{N}\sum_{i,j,ts}(Aq_{u_i,s_j,ts} - Pq_{u_i,s_j,ts}) \qquad (10)$$

where $Aq_{u_i,s_j,ts}$ is the actual QoS values and $Pq_{u_i,s_j,ts}$ is predicted QoS values of s_j for u_i at time interval ts, and N is the total number of predicted values. Obviously, a smaller MAE indicates a more accurate prediction.

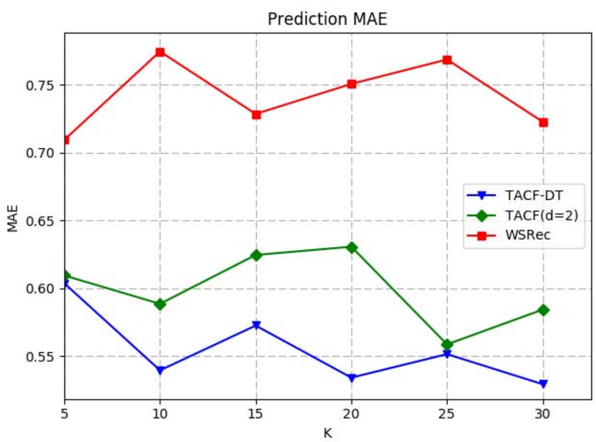

Figure 1. MAE for all methods under different K

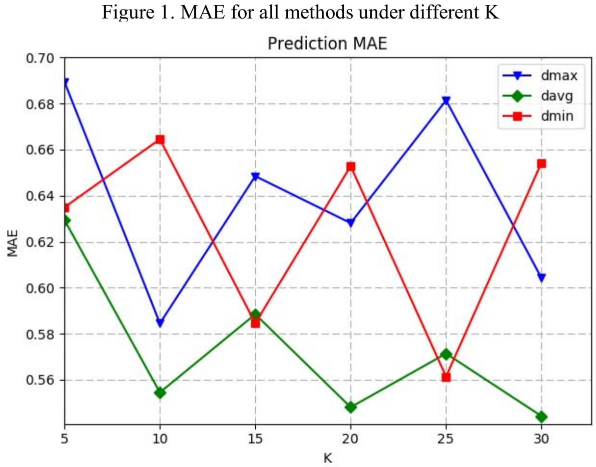

Figure 2. MAE for TACF-DT under different d

C. Evaluation and comparison

Firstly, we study the impact of K. Figure 1 illustrates the MAE of all methods under different value of parameter K in Top-K recommendation. Our TACF-DT method outperforms both WSpred and $TACF_{d=2}$ methods in all the conditions. Then, we study the impact of parameter d. In algorithm 1, there is parameter d, which is the threshold of QoS change amount at time interval searching. To study the impact of parameter d, we set TopK = 5, 10, 15 …30 and the value of $d = d_{min}$, d_{max} and d_{avg} respectively. Figure 2 illustrates the experiment results. We can conclude that the method achieves best prediction accuracy when $d = d_{avg}$.

D. Discussion

According to the experiment results above, we can reach a conclusion that our TACF-DT method indeed enhances the QoS prediction accuracy especially when QoS values of services change periodically.

V. Conclusion

In this paper, we have proposed a novel temporal influences-aware collaborative filtering method which designs an enhanced temporal influences-aware similarity measurement algorithm to predict QoS values. Comprehensive experiments show that the method outperforms other QoS-based recommendation methods. In future work, we will incorporate time-decay function in our temporal influences-aware similarity measurement and experiment on more datasets to determine the optimal parameter values.

Acknowledgment

This work was supported by NSFC (61571066), NSFC (61472047).

References

[1] Y. Hu, Q. Peng and X. Hu, "A Time-Aware and Data Sparsity Tolerant Approach for Web Service Recommendation," 2014 IEEE International Conference on Web Services, Anchorage, AK, 2014, pp. 33-40. doi: 10.1109/ICWS.2014.18.

[2] C. Yu and L. Huang, "Time-Aware Collaborative Filtering for QoS-Based Service Recommendation," 2014 IEEE International Conference on Web Services, Anchorage, AK, 2014, pp. 265-272. doi: 10.1109/ICWS.2014.47.

[3] Z. Zheng, H. Ma, M.R. Lyu, and I.King, "WSRec: a collaborative filtering based web service recommendation system," Proc. IEEE International Conference on Web Services (ICWS09), 2009, pp. 437-444.

[4] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for qos-based service recommendation," in Web Services (ICWS), 2012 IEEE 19th International Conference on. IEEE, 2012, pp. 202–209.

[5] X. Chen, X. Liu, Z. Huang, and H. Sun, "RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized Web service recommendation," Proc. IEEE International Conference on Web Services (ICWS 10), 2010, pp. 9-16.

[6] X. Fan, Y. Hu, R. Zhang, W. Chen, P. Brézillon and X. Fan, "Modeling Temporal Effectiveness for Context-Aware Web Services Recommendation," 2015 IEEE International Conference on Web Services, New York, NY, 2015, pp. 225-232.

[7] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," Proc. IEEE International Conference on Web Services (ICWS 07), 2007, pp. 439-446.