

# Context-Aware Web Services Recommendation Based on User Preference Expansion

Yakun Hu<sup>1</sup>, Xiaoliang Fan<sup>1,2,3</sup>(✉), Ruisheng Zhang<sup>1</sup>,  
and Wenbo Chen<sup>1</sup>

<sup>1</sup> School of Information Science and Engineering,  
Lanzhou University, Lanzhou 730000, China

{huykl4, fanxiaoliang, zhangrs, chenwb}@lzu.edu.cn

<sup>2</sup> Fujian Key Laboratory of Sensing and Computing for Smart City,  
School of Information Science and Engineering, Xiamen University,  
Xiamen 361000, China

<sup>3</sup> State Key Laboratory for Novel Software Technology,  
Nanjing University, Nanjing 210023, China

**Abstract.** Context-Aware Recommender System is known to not only recommend items or services similar to those already rated with the highest score, but also consider the current contexts for personalized Web services recommendation. Specifically, a key step for CARS methods refers to previous service invocation experiences under the similar context of the user to make Quality of Services prediction. Existing works either considered the influence of regional correlations on user preference, or combined the location-aware context with the matrix factorization method. However, the user preference expansion triggered by instant update of user location is not fully observed. For instance, when making Web service recommendation for a user, it is expected to be aware of rapid change of the user location immediately and the expansion of user preference as well. In this paper, we propose a Web services recommendation approach dubbed as CASR-UPE (Context-aware Web Services Recommendation based on User Preference Expansion). First, we model the influence of user location update on user preference. Second, we perform the context-aware similarity mining for updated location. Third, we predict the Quality of Services by Bayesian inference, and thus recommend the best Web service for the user subsequently. Finally, we evaluate the CASR-UPE method on WS-Dream dataset by evaluation matrices such as RMSE and MAE. Experimental results show that our approach outperforms several benchmark methods with a significant margin.

**Keywords:** Context awareness · Web service · Recommender system · QoS · Preference expansion

## 1 Introduction

Context-Aware Recommender System (CARS) for Web services aims to recommend Web services not only similar to those already rated with the highest score, but also that could combine the contextual information with the recommendation process [1, 4, 10].

In recent years, preliminary benefits have been seen in Web services recommendation considering various contextual factors [2, 3]. For example, temporal [7, 8], spatial [9, 12, 13] and social [5, 6, 11, 14, 16] contexts are widely extracted separately for personalized Web services recommendation.

Specifically, one of the key steps for CARS method is referring to previous service invocation experiences under the similar location of the current user to make Quality of Services (QoS) prediction [17]. Existing works mainly discussed the influence of regional correlations on user preference [14]. There are also several novel methods combining the location-aware contexts with matrix factorization methods [13]. However, the user preference expansion triggered by instant update of user location is not fully observed for personalized recommendation. For instance, when making recommendation for a user, we are expected to be aware of the rapid change of the user location immediately, and thus of the expansion of user preference as well.

In this paper, we propose a Web services recommendation approach dubbed as CASR-UPE. Our approach consists of three steps: (1) model the influence of user location update on user preference; (2) perform the context-aware similarity mining for updated location; (3) predict the QoS of Web services by Bayesian inference and recommend the best Web service for the user subsequently. Finally, we evaluate the CASR-UPE algorithm on WS-Dream dataset [17] by evaluation matrices of both RMSE and MAE. Experimental results show that our approach outperforms the six state-of-the-art benchmark methods with a significant margin.

Hereafter, the paper is organized as follows. Section 2 introduces the related works. Section 3 shows a motivating example. Section 4 gives the details of CASR-UPE method. Section 5 shows the experimental results and discussion. Finally, the general conclusion as well as perspectives in Sect. 6 closes this paper.

## 2 Related Works

Context-aware recommender system has gained significant momentum in recent years. The development of mobile devices and their crowd sensing capabilities have enabled the collection of rich contextual information on time and location [9, 11–13].

In the CARS methods, the temporal contexts [7, 8] have been widely used in conventional CARS methods. Another widely discussed context information is the location context [9, 12, 13], especially in LBSN [5, 6]. A location-aware services recommendation method is presented in [9, 10] by referring to previous service invocation experiences under the similar location with the current user. However, these methods merely consider the location as a filter to make recommendation to the current user. The influence of regional correlation on user preference is also considered in [13]. In addition, a location-based hierarchical matrix factorization (HMF) method [15] is proposed to perform personalized QoS prediction. In short, the above location-based service recommendation methods overlook the user preference expansion triggered by the instant update of user location.

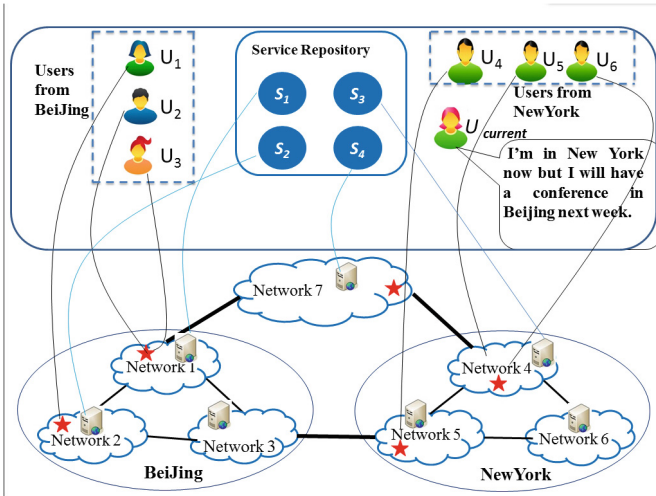
Furthermore, temporal location correlations [20, 21] have been studied for location recommendations in location-based social network (LBSN). However, the location

recommendation in LBSN is different from Web services recommendation and the temporal location effects seem to be not suitable for Web services recommendation.

Among the current works of considering location-aware context in CARS, we propose a Web services recommendation approach, considering the influence of user location update on user preference, and then have an updated location similarity mining.

### 3 A Motivating Example

Figure 1 is a scenario of use to recommend weather forecast services considering the user preference expansion. The upper part represents a Web services repository ( $S_1, S_2, \dots, S_n$ ) and many service users ( $u_1, u_2, \dots, u_m$ ), where services and users are distributed all over the world. Suppose  $S_1 = \text{“Weather China”}^1$ ;  $S_2 = \text{“Moji Weather China”}^2$ ;  $S_3 = \text{“US National Weather Service”}^3$ ;  $S_4 = \text{“Le Figaro météo in France”}^4$ . The underlying part illustrates the distributed networks. The curves link users and services to their corresponding geographic positions.



**Fig. 1.** A scenario of weather forecast services recommendation considering user preference expansion

Firstly, in this scenario of Fig. 1,  $U_{current}$  is in NYC at present time. As we all know, the accuracy of weather forecast services is highly relevant to the location and it is natural to believe that a user prefers the service either located in his/her city. So when making recommendations to  $U_{current}$  at present time, we should consider her current

<sup>1</sup> Weather China, <http://en.weather.com.cn/>.

<sup>2</sup> Moji Weather China, <http://www.moweather.com/>.

<sup>3</sup> US national weather service, <http://www.weather.gov/>.

<sup>4</sup> Le Figaro météo weather forecasting service, <http://www.lefigaro.fr/meteo/france/index.php>.

location (New York) and recommend  $S_3$  to her. However, if we know that the user will have a conference in Beijing next week (e.g. from his/her mobile phone calendar), we should consider the influence of location update (Beijing) on user preference and recommend  $S_1$  to  $U_{\text{current}}$  when recommending weather services for next week. In a word, when making recommendations to users, we should consider the influence of user location update on user preference.

Secondly, when making recommendations to  $U_{\text{current}}$ , we should also consider the set of users in the same location with  $U_{\text{current}}$  because of “the more similar between the current user and another user’s context (e.g. location), the more probability of the two users will have a same preference”. The user location update will lead to different sets of similar users. For instance,  $U_{\text{current}}$  is in New York at present time, her similar set of users is made of  $U_4, U_5, U_6$  (all from New York). But  $U_{\text{current}}$  will be in Beijing next week, her similar set of users will become  $U_1, U_2, U_3$  (all from Beijing). So we should also consider the influence of user location update on the set of similar users.

Above all, we consider the expansion of user preference triggered by instant update of user location when making recommendation to the current user.

## 4 CASR-UPE Algorithm: Context-Aware Web Services Recommendation Based on User Preference Expansion

### 4.1 Problem Definition

In order to help readers understanding our algorithm better, the following definitions are given.

We assume that there are a set of users  $U = \{u_1, u_2, \dots, u_n\}$  and a set of Web services  $S = \{s_1, s_2, \dots, s_m\}$ .  $u_i (1 \leq i \leq n)$  in the context-aware web services recommender system. A service user from  $U$  must have invoked a service from  $S$  at least once.

$L_{U,t} = \{l_{u_i,t}\} (1 \leq i \leq n)$  is the set of  $l_{u_i,t}$  that is the temporal location of user  $u_i$ .

$L_S = \{l_{s_k}\}$  is the set of  $l_{s_k}$ , which is the network location of the service  $s_k$ .

$R = \{r_{u_i,s_k}\}$  is the set of rating records on the Web service  $s_k$  by the user  $u_i$ , where  $(1 \leq i \leq n)$  and  $(1 \leq k \leq m)$ .

$\bar{R} = \{\bar{r}_1, \bar{r}_2, \dots, \bar{r}_i, \dots, \bar{r}_n\}$  is the set of mean rating of all Web services invoked by  $U = \{u_1, u_2, \dots, u_n\}$ .

When a service  $s_k$  is invoked by the user  $u_j$ , it will present a set of QoS properties. We will have  $Q_{i,j} = \langle q_1^{i,j}, q_2^{i,j}, \dots, q_k^{i,j} \rangle$ , which is a 1-tuple denoting service invocation records of  $s_k$  invoked by the user  $u_j$ , where  $q_l^{k,j} (1 \leq k \leq m, 1 \leq j \leq n)$  denotes the value of  $l$ -th property recorded during the invocation of  $s_k$  called by  $u_j$ .

### 4.2 Modeling the Influence of User Location Update on User Preference

We know that the user’s locations are different as time goes on. As described in Sect. 3, for region-related service (e.g. weather forecast services), the accuracy of

recommendation is highly relevant to the specific region in real-time. The impact of regional correlation on user preference is defined as:

$$P_{RCL(t)} = \begin{cases} 1 & \text{if web service is related to regio} \\ 0 & \text{if web service is not related to region} \end{cases}, \quad (1)$$

For region-irrelated services, it's also reasonable for users to have a preference to services near to his/her region because the network distance between users and services (mainly because of transfer delay) have an obvious effect on Internet application performance (e.g. response time and throughput). The network distance's influence on user preference can be defined as:

$$P_{NDL(t)} = P_0 \text{Dis}(l_{u_i,t}, l_{s_k})_{nor}, \quad (2)$$

Here we assign 1 to it according to our need.  $\text{Dis}(l_{u_i,t}, l_{s_k})$  is the network distance between  $l_{u_i,t}$  (the user's network location) and  $l_{s_k}$  (the service's network location). Also, network distance measurement technology can help us to get the  $\text{Dis}(l_{u_i,t}, l_{s_k})$ , which should be normalized as  $\text{Dis}(l_{u_i,t}, l_{s_k})_{nor}$  to have a same evaluation criterion

In addition, different weights are assigned to the impact of both regional correlation and network distance ( $w_1$  to  $P_{RCL(t)}$  and  $w_2$  to  $P_{NDL(t)}$ ). Thus,  $w_1$  and  $w_2$  will be used to represent the influence of user location update on user preference as follows:

$$P_{L(t)} = w_1 P_{RCL(t)} + w_2 P_0 \text{Dis}(l_{u_i,t}, l_{s_k})_{nor}, \quad (3)$$

Finally, we can have a data filtering results based on  $P_{L(t)}$  and get the services which correspond to the current preference of a user.

### 4.3 Context-Aware Similarity Mining for Updated Location

In this step, it is assumed that for location-based services recommendation, the more similar between the current user and another user's context (e.g. location), the more probability of the two users will have similar QoS on the same Web service. The set of similar users with the current user in terms of location will be got after this step.

We use the Euclidean distance later to describe the similarity between the two users' locations. The nearer the distance is, the more similar they are. The following formula is the presentation of Euclidean distance between  $l_{u_i,t}$  and  $l_{u_j,t}$ :

$$\text{Sim}(l_{u_i,t}, l_{u_j,t}) = \sqrt{\sum_{k=1}^N (l_{i,t,k} - l_{j,t,k})^2}, \quad (4)$$

Furthermore, we could calculate the distances between current user's location and other users' locations, thus and get the set of users who are the closest with the current user.

#### 4.4 QoS Predication and Services Recommendation

In the final step, we use the Bayesian inference to make QoS prediction and services recommendation based on the past invocation records filtered from the above steps. The formula of Bayesian inference is defined as:

$$P(OS = 1|s_i) = \frac{P(s_i|OS = 1) * P(OS = 1)}{P(s_i)}, \quad (5)$$

In order to explain our formula, we give an example in Table 1. In the example, we suppose that a threshold (e.g.  $q = 0.7$ ) just for the explanation of Bayesian inference (When we have experiment, we will set different values of  $q$  to get different results so that we can find the  $q$  which lead to the best result). That is to say, if  $QoS > 0.7$ , we will say the service satisfies the user who invoked it, while if  $QoS < 0.7$ , we will say the service is not satisfied. We use 1 to donate “satisfied” while 0 to donate “not satisfied”.

**Table 1.** Example of Bayes inference

Record	QoS	OS
$\langle s_1, u_1, 1 \rangle$	0.85	1
$\langle s_1, u_1, 2 \rangle$	0.75	1
$\langle s_1, u_1, 3 \rangle$	0.45	0
$\langle s_2, u_1, 1 \rangle$	0.80	1
$\langle s_2, u_1, 2 \rangle$	0.50	0
$\langle s_2, u_1, 3 \rangle$	0.60	0
$\langle s_3, u_1, 1 \rangle$	0.75	1
$\langle s_3, u_1, 2 \rangle$	0.55	0

Table 1 shows an example of service invocation records, where each triple  $s_i, u_j, n$  represents a  $n$ -th service invocation of  $s_i$  by the user  $u_j$ . According to formula (6), the  $P(OS = 1|s_i)$  donates the prediction QoS of the current user to the Web service  $s_i$ ,  $P(OS = 1)$  donates the probability of the satisfactory ones in all the web service,  $P(s_i|OS = 1)$  donates the probability of Web service  $s_i$  in the satisfactory ones. The maximum result represents the best service. Thus, we can recommend the *top n* Web services to the current user. The approaches to calculate  $P(OS = 1|s_i)$  are:

$$P(OS = 1|s_1) = \frac{P((s_1|OS = 1) * P(OS = 1))}{P(s_1)} = \frac{\frac{1}{2} * \frac{1}{2}}{\frac{3}{8}} = \frac{2}{3}$$

$$P(OS = 1|s_2) = \frac{P((s_2|OS = 1) * P(OS = 1))}{P(s_2)} = \frac{\frac{1}{4} * \frac{1}{2}}{\frac{3}{8}} = \frac{1}{3}$$

$$P(OS = 1|s_3) = \frac{P((s_3|OS = 1) * P(OS = 1))}{P(s_3)} = \frac{\frac{1}{4} * \frac{1}{2}}{\frac{3}{8}} = \frac{1}{2}$$

Finally, we can make the QoS prediction for the user  $s_1$  by 2/3. Later, according to the results of each Web service, we could rank the value from the higher to the lower. Hence, we conclude that  $s_1$  would be recommended to the current user compared with  $s_2$  and  $s_3$ . The entire procedure of CASR-UPE algorithm is shown as follows.

**Algorithm: Context-aware Web Services Recommendation based on User Preference Expansion (CASR-UPE)**

```

Input: q, u, t, dataset
    // q is the threshold of the QoS; u is the test user;
    t is the time of recommendation; dataset is the training
dataset. //
Output: A group of values of MAE and RMSE with different
QoS threshold q.
1. Start
2. When (t == t1)
3.  $P_{L(t_1)} = w_1 P_{RCL(t_1)} + w_2 P_0 \text{Dis}(l_{u,t_1}, l_{s_k})_{\text{nor}}$ ;
4. // get the set  $P_{L(t_1)}$  of Web services corresponding
to user current preference determined by  $L(t_1)$ , the lo-
cation of the user at time  $t_1$ //
5. dataset2= filtered( $P_{L(t_1)}$ , dataset);
6. //get the filtered dataset2 according to the pref-
erence set  $P_{L(t_1)}$ //
7.  $U_{L(t_1)} = \text{Sim}(u)$ ;
8. //get similar user set  $U_{L(t_1)}$  in location //
9. dataset3=filtered( $U_{L(t_1)}$ , dataset2);
10. //get the filtered dataset3 according to the
similar set  $U_{L(t_1)}$  //
11. for (different q)
12. preQoS =Bayesian (q, dataset3);
13. end for
14. mae = MAE (preQoS); rmse=RMSE(preQoS);
15. When (t == t2)
16.  $P_{L(t_2)} = w_1 P_{RCL(t_2)} + w_2 P_0 \text{Dis}(l_{u,t_2}, l_{s_k})_{\text{nor}}$ ;
17. // get the set  $P_{L(t_2)}$  of Web services cor-
responding to user current preference //
18. dataset2= filtered( $P_{L(t_2)}$ , dataset);
19. // get the filtered dataset2 according
to the preference set  $P_{L(t_2)}$ //
20.  $U_{L(t_2)} = \text{Sim}(u)$ ;
21. // get similar user set  $U_{L(t_2)}$  in location //
22. dataset3=filtered( $U_{L(t_2)}$ , dataset2);
23. //get the filtered dataset3 according to the
similar set  $U_{L(t_2)}$  //
24. for (different q)
25. preQoS =Bayesian (q, dataset3);
26. end for
27. mae = MAE (preQoS); rmse=RMSE(preQoS);
28. // For different preQoS, we can calculate MAE
and RMSE values of q. //
29. End When

```

## 5 Experiments

In this section, we choose six algorithms to compare with CASR-UPE algorithm on WS-Dream dataset by evaluation metrics of both MAE and RMSE.

### 5.1 Datasets and Data Processing

WS-Dream [17] dataset <sup>5</sup> is adopted in our experiments, which contains 1,542,884 Web services invocation records executed by 150 distributed service users on 100 Web services. Approximately, every user invokes a Web service 100 times. Each invocation record contains 6 parameters: IP address, WSID (ID of web service), RTT (round-trip time), Data Size, Response HTTP Code, and Response HTTP Message. Since Response HTTP Code and Message are highly related, here we omit the property Response HTTP Code.

The raw data must be normalized before use. Gaussian approach is used to normalize QoS data, due to its well-balanced distribution. The normalization rule for Response HTTP Message is as follows: if the message is “OK”, the normalized value is 1, otherwise it is 0. The normalization rule for RTT and Data Size is defined as:

$$r_1^{k,j} = 0.5 + \left( r_1^{k,j} - \bar{r}_1^j \right) / (2 * 3\sigma_j), \quad (6)$$

Where  $\sigma_j$  is the standard deviation of user  $u_j$ 's QoS data on the  $l$ -th property and  $\bar{q}_l^k$  denotes the arithmetic mean of QoS data collected from user  $u_j$  on the  $l$ -th QoS property. Now we can simulate the feedback of a user after invoking a Web service by evaluating the overall QoS of a service. The weight QoS formula can be described as:

$$QoS = w_1 * v_{RTT} + w_2 * v_{DataSize} + w_3 * v_{RHTTPMessage}, \quad (7)$$

Where  $w_1, w_2$  and  $w_3$  are set to 0.35, 0.05 and 0.6 respectively according to their different significance. For example, Response HTTP Message shows that whether the invocation succeeded so it is a fundamental property and can be set 0.6. Thus the properties of RTT and Data Size are not that important and they can be set 0.35 and 0.05 respectively.

All experiments were developed with Matlab. They were performed on a Lenove desktop computer with the following configuration: Intel Core i5 2.50 GHz CPU, 2 GB RAM with the Windows 7 operating system.

### 5.2 Evaluation Metrics

The evaluation metrics [17] we use in our experiments are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE):

---

<sup>5</sup> WS-Dream dataset, <http://www.wsdream.net/dataset.html>.



$$MAE = \frac{\sum_{u,s} |Q_{u,s} - \hat{Q}_{u,s}|}{N}, \quad (8)$$

$$RMSE = \frac{\sqrt{\sum_{u,s} (Q_{u,s} - \hat{Q}_{u,s})^2}}{N}, \quad (9)$$

In the formulas (8) and (9),  $Q_{u,s}$  denotes actual QoS values of a Web service  $s$  observed by the user  $u$ ,  $\hat{Q}_{u,s}$  represents the predicted QoS values of service  $s$  for the user  $u$ , and  $N$  denotes the number of predicted value.

### 5.3 Evaluation

**Comparative Algorithms.** Six algorithms are compared with our CASR-UPE in this paper:

- **RBA** (Recommendation by all): recommend web services to a user collected by all users without a filtering.
- **UPCC** [22]: recommend web services collected from the users sharing the similar preference with the current user (PCC based on user profiles).
- **IPCC** [23]: recommend web services similar to the ones the current user preferred in the past (PCC based on services).
- **CASR** [9]: make recommendation based on the service invocation experiences under similar location context with the current user.
- **ITRP-WS** [24]: ITRP-WS considers the time decay effects in UPCC.
- **CASR-UP** [25]: make recommendation considering the user preference determined by user's location.

**Performance Comparison.** Figure 2 shows the results of MAE and RMSE for different algorithms. The results are generated in different threshold  $q$  (from 0.65 to 0.95 and the interval is 0.025) in the ratio 14:1 of training dataset and test dataset. From Fig. 2, we could make the conclusion that: (1) the MAEs and RMSEs of CASR-UPE are much better than other five algorithms when the threshold  $0.725 \leq q \leq 0.925$ ; (2) it is abnormal when the threshold  $q = 0.95$ ; and (3) When  $q \leq 0.725$ , the MAEs and RMSEs of the algorithms remain almost invariable. We can also see that the best  $q$  is 0.775. In Sect. 5.4, we will further explain the reason of both (2) and (3). In general, the results demonstrate that the significant of CASR-UPE algorithm in recommending web services considering the user preference expansion.

Figure 3 shows the average MAE/RMSE results of the six algorithms in different ratios (8:7, 9:6, 10:5, 11:4, 12:3, 13:2, and 14:1). What we could learn from the results are: (1) as the ratio of training and test data increases the MAE and RMSE results of six algorithms decrease; (2) in different ratios, the results of the CASR-UPE algorithm also performs better than the other six algorithms; and (3) CASR-UPE performs worse than CASR-UP in the ratio of 8:7, 9:6 and 10:5 adopting RMSE as the evaluation metric. In Sect. 5.4, we will further explain reasons for those three results above. In general,

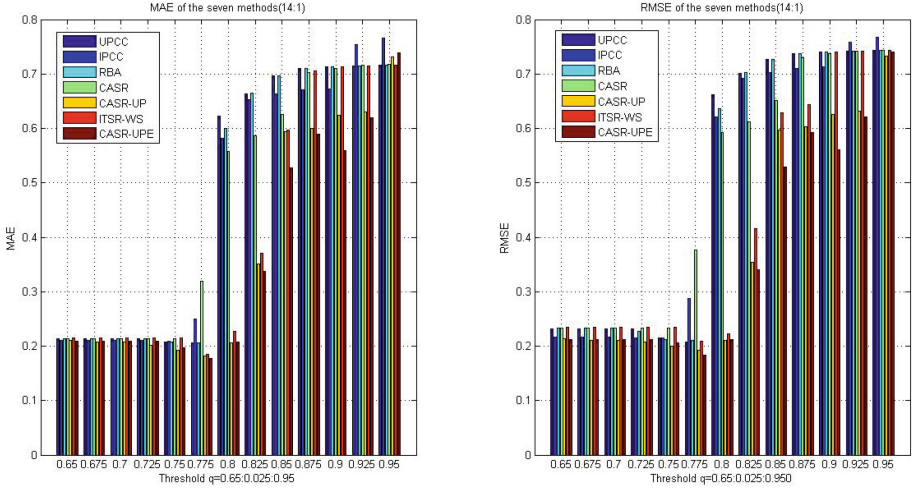


Fig. 2. MAE and RMSE results of compared methods (14:1)

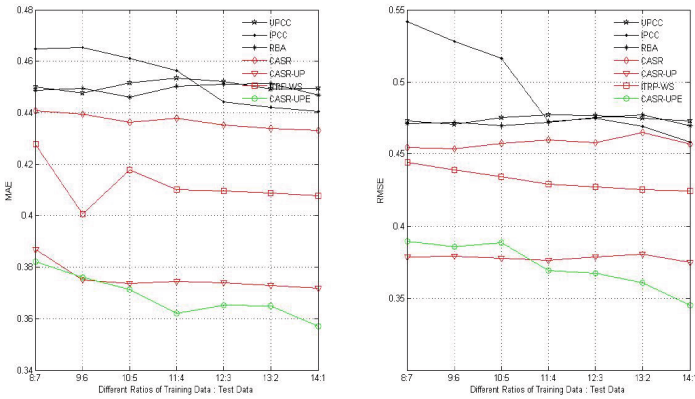


Fig. 3. MAE and RMSE results of compared methods (in various ratios)

the results in different ratios demonstrate that the significant of CASR-UPE algorithm in recommending web services considering the user preference expansion.

## 5.4 Discussion

In this subsection, we will discuss two aspects in our experiments to further explain the results in 5.3.

**Trade-off Parameters:** From the results of Fig. 2, we can infer that: (1) The MAEs and RMSEs of CASR-UPE are smaller than other algorithms when the threshold  $0.725 \leq q \leq 0.925$ , but why  $q = 0.95$  is an exception? By analyzing CASR-UPE

method, we can find that after selecting some Web services according to the user dynamic preference, the invocation records of these selected services are more useful. As the threshold  $q$  rises up to 0.95, most of positive services will be excluded and the results will be in a high and abnormal value. (3) Why is that when  $q \leq 0.725$ , the MAEs and RMSEs of the algorithms remain almost invariable? When the threshold  $q$  decreases, the request of QoS will decrease and many negative services will be included. When  $q$  is low enough, all Web services will be included, thus the MAEs and RMSEs remain invariable. (4) We could conclude that the threshold  $q$  for the calculated probability is highly relevant to the result. If  $q$  is too low, many negative Web services will be included, while if  $q$  is too high, many positive Web services will be excluded.

Figure 3 shows the influence of different ratios on the MAEs and RMSEs results. When the ratio of training dataset and test dataset arises, more data is used to train the algorithm and few data is used to test the results. Thus, the accuracy will be better. However, why does CASR-UPE perform worse than CASR-UP in the ratio of 8:7, 9:6 and 10:5 adopting RMSE as the evaluation metrics? The possible reason may come from the randomly changed training dataset when the ratio of training data: test data decreases.

**Impact of User Preference Expansion.** Comparing with other algorithms not considering user preference expansion, we got the results of the impact of user preference expansion on recommendation accuracy. The results shown in the Figs. 2 and 3 collectively demonstrate that: (1) the combination of the influence of user location update on user preference get a better recommendation; and (2) the updated location similarity mining could also further improve the accuracy of recommendation.

## 6 Conclusion

In this paper, we propose the CASR-UPE algorithm, for modeling the influence of user location update on user preference and performing updated location similarity mining. Finally, the experiments results show that CASR-UPE algorithm improves predictive accuracy and outperforms the compared methods.

Despite the significant progress of user preference expansion in context-aware Web services recommendation, there still remain numerous avenues to explore. Our future works include: (1) incorporate novel context properties, such as social context (interpersonal interest similarity, interpersonal influence among social network, etc.) to improve more personalized recommendation; and (2) focus on the correlations between context properties, such as temporal-spatial correlations to improve the accuracy of QoS prediction.

**Acknowledgment.** This work is supported by the grants from Natural Science Foundation of China (No. 61300232); Ministry of Education of China “Chunhui Plan” Cooperation and Research Project (No. Z2012114, Z2014141); Funds of State Key Laboratory for Novel Software Technology, Nanjing University (KFKT2014B09); Fundamental Research Funds for the Central Universities (lzujbky-2015-100); and China Telecom Corp. Gansu Branch Cuiying Funds (lzudxcy-2014-6).

## References

1. Truong, H., Dustdar, S.: A survey on context-aware web service systems. *Int. J. Web Inf. Syst.* **5**(1), 5–31 (2009)
2. Dourish, P.: What we talk about when we talk about context. *Pers. Ubiquit. Comput.* **8**(1), 19–30 (2003)
3. Jannach, D., Dortmund, T.U., Friedrich, G.: Tutorial: recommender systems. In: *International Joint Conference on Artificial Intelligence, Beijing* (2013)
4. Staiano, J., Oliver, N., Lepri, B., de Oliveira, R., Caraviello, M.: Money walks: a human-centric study on the economics of personal mobile data. In: *2014 ACM Conference on Ubiquitous Computing*, pp. 583–594. ACM, Seattle (2014)
5. Rossi, L., Musolesi, M.: It's the way you check-in: identifying users in location-based social network. In: *2nd ACM Conference on Online Social Networks*, pp. 215–226. ACM, Dublin (2014)
6. Lima, A., Musolesi, M.: Spatial dissemination metrics for location-based social networks. In: *2012 ACM Conference on Ubiquitous Computing*, pp. 972–979. ACM, Pittsburgh (2012)
7. Papapetrou, P., Roussos, G.: Social context discovery from temporal app use patterns. In: *2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pp. 397–402. ACM, Seattle (2014)
8. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: *33rd international ACM SIGIR Conference on Research and development in information retrieval*, pp. 210–217. ACM, Geneva (2010)
9. Xiong, H., Zhang, D., Gauthier, V.: Predicting mobile phone user locations by exploiting collective behavioral patterns. In: *IEEE 9th International Conference on Ubiquitous Intelligence and Computing (UIC 2012)*, pp. 164–171. IEEE Press (2012)
10. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 217–253. Springer, New York (2011)
11. Guo, B., Chen, C., Zhang, D., Yu, Z., Chin, A.: Mobile crowd sensing and computing: when participatory sensing meets participatory social media. *IEEE Commun. Mag.* (2015)
12. Yu, Z., Feng, Y., Xu, H., Zhou, X.: Recommending travel packages based on mobile crowdsourced data. *IEEE Commun. Mag.* **52**(8), 56–62 (2014)
13. Yin, H., Cui, B., Chen, L.: Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data* **38** (2014)
14. Yang, D., Zhang, D., Yu, Z.: Fine-grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs. In: *2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 479–488. ACM, Zurich (2013)
15. Yu, Q., Zheng, Z., Wang, H.: Trace norm regularized matrix factorization for service recommendation. In: *2013 IEEE International Conference on Web Services*, pp. 34–41. IEEE press, Santa Clara Marriott (2013)
16. Cao, B., Liu, J., Tang, M., Zheng, Z., Wang, G.: Mashup service recommendation based on user interest and social network. In: *IEEE International Conference on Web Services*, pp. 99–106. IEEE Press, Santa Clara Marriott (2013)
17. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware web service recommendation by collaborative filtering. *IEEE Trans. Serv. Comput.* **4**(2), 140–152 (2010)
18. McCarthy, J.: Notes on formalizing context. In: *International Joint Conference on Artificial Intelligence*, pp. 555–560 (1993)

19. Brézillon, P.: Task-realization models in contextual graphs. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) *CONTEXT 2005*. LNCS (LNAI), vol. 3554, pp. 55–68. Springer, Heidelberg (2005)
20. Gao, H., Tang, J., Hu, X., Liu, H.: Modeling temporal effects of human mobile behavior on location-based social networks. In: *ACM International Conference on Information and Knowledge Management*, pp. 1673–1678. ACM, Maui (2013)
21. Hu, B., Jamali, M., Ester, M.: Spatio-temporal topic modeling in mobile social media for location recommendation. In: *IEEE International Conference on Data Mining*, pp. 1073–1078. IEEE press (2013)
22. Breese, J., Heckerman, D., Kadie, C.: *Empirical Analysis of Predictive Algorithms For Collaborative Filtering*. Morgan Kaufmann Publishers Inc., San Francisco (1998)
23. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: An open architecture for collaborative filtering of netnews. In: *ACM Conference on Computer Supported Cooperative Work*, pp. 175–186 (1994)
24. Cui, X., Yin, G., Han, Q., Dong, Y.: An improved time-effectiveness reliability prediction approach of web service. *J. Comput. Inf. Syst.* **10**(4), 1365–1374 (2014)
25. Fan, X., Hu, Y., Zhang, R.: Context-aware web services recommendation based on user preference. In: *IEEE Asia-Pacific Services Computing Conference*, pp. 55–61. IEEE press, Fuzhou (2014)