Modeling Temporal Effectiveness for Context-aware Web Services Recommendation

Xiaoliang Fan, Yakun Hu, Ruisheng Zhang, Wenbo Chen
School of Information Science and Engineering
Lanzhou University
Lanzhou, P.R. China
{fanxiaoliang; huyk14; zhangrs; chenwb}@lzu.edu.cn

Patrick Brézillon
Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
Paris, France
patrick.brezillon@lip6.fr

Xiaoliang Fan
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, P.R. China
xiaoliang.fan@gmail.com

***Abstract*—Context-Aware Recommender System (CARS) aims to not only recommend services similar to those already rated with the highest score, but also provide opportunities for exploring the important role of temporal, spatial and social contexts for personalized web services recommendation. A key step for temporal-based CARS methods is to explore the time decay process of past invocation records to make the Quality of Services (QoS) prediction. However, it is a nontrivial task to model the temporal effects on web services recommendation, due to the dynamic features of contextual information in view of temporal spatial correlations. For instance, in location-aware services recommendation, the user's geographical position would change very frequently as time goes on. In this paper, we propose a Context-Aware Services Recommendation based on Temporal Effectiveness (CASR-TE) method. Inspired by existing time decay approaches, we first present an enhanced temporal decay model combining the time decay function with traditional similarity measurement methods. Then, we model temporal spatial correlations as well as their impacts on the user preference expansion. Finally, we evaluate the CASR-TE method on WS-Dream dataset by evaluation matrices of both RMSE and MAE. Experimental results show that our approach outperforms several benchmark methods with a significant margin.**

Keywords-context awareness; web services; recommender system; QoS; temporal effectiveness.

I. INTRODUCTION

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

Quality of Service (QoS) describes the non-functional characteristics of web services, including response time, throughput, availability, etc. Preliminary benefits have been seen in recommending web services when taking contextual factors into account [2-4]. Specifically, temporal [5-8], spatial [4, 9, 11] and social [10] contexts are extracted separately for personalized web services recommendation.

A key step for temporal-based web services recommendation methods is to consider the decay process of time-effectiveness from past invocation records to make the QoS prediction of web services [13]. For instance, a longer timespan of service invocation may imply a deviation of QoS value, due to the shutdown of services or network failures. However, most existing works overlook dynamic features of temporal contexts, and specifically ignore the correlations between temporal and spatial contexts. For example, in location-aware services recommendation, the user's geographical position would change continuously over time.

In this paper, we propose a Context-Aware Services Recommendation based on Temporal Effectiveness (CASR-TE) method. Our approach consists of four steps: (1) model an enhanced temporal decay method to combine the time decay function with traditional similarity measurement methods; (2) mine the location-aware similarity with temporal effects; (3) explore temporal spatial correlations as well as their impacts on the user preference expansion; and (4) predict the QoS of web services by Bayesian inference. Finally, we evaluate the CASR-TE algorithm on WS-Dream dataset [19] by evaluation matrices of both RMSE and MAE. Experimental results show that our approach outperforms six state-of-the-art benchmark methods with a significant margin.

Hereafter, the paper is organized as follows. Section II introduces the related works. Section III shows a motivating example. Section IV gives the details of CASR-TE method. Section V shows the experimental results and many discussions. Finally, the general conclusion and perspective in Section VI closes this paper.

II. RELATED WORKS

Context-aware recommender system (CARS) [1] has gained significant momentum in recent years. There are many different approaches of obtaining, representing and modeling contextual factors in recommender systems, due to the fact that context is a complex notion with almost infinite dimension [12, 13]. For instance, the knowledge of a recommender system about the contextual factors was classified into three categories [1]: fully observable, partially observable and unobservable.

A key step for modeling personalized web services recommendation is that contextual factors are extracted from invocation records of web services. First, the temporal contexts [5-8] have been widely used in conventional CARS methods. The "user-service-time" triadic relations are represented in [5] to analyze latent features in

978-1-4673-7272-5/15 $31.00 © 2015 IEEE
DOI 10.1109/ICWS.2015.39

recommendation by a three-dimensional tensor. In addition, time decay function is used to compute time weights for different services [6, 7]. However, most works overlook dynamic features of the temporal impact on web services recommendation, and specifically ignore the correlations between temporal and spatial contexts. For instance in location-aware services recommendation, the user's geographical position would change continuously over time.

The second widely discussed context information is location or spatial contexts [4, 9, 11]. A context-aware services recommendation (CASR) method is presented in [4] by referring to previous service invocation experiences under the similar location with the current user. The regional correlation's influence on user preference is also considered in [9]. In addition, a location-based hierarchical matrix factorization (HMF) method [11] is proposed to perform personalized QoS prediction.

Specifically, in location-based social network, temporal spatial correlations [14, 15] have been well studied for location recommendations. For example, four temporal aggregation strategies are presented in [14] to explore correlations between a user's check-in time and the corresponding check-in location. However, the temporal effects for location recommendation, such as power-law distribution and temporal consecutiveness, seem to be not directly suitable for web services recommendation.

Among the current works of modeling temporal effects in CARS, we present an enhanced temporal decay model combining the time decay function with similarity measurement methods, then model temporal spatial correlations as well as their impacts on user preference expansion.

III. A MOTIVATING EXAMPLE

Fig. 1 shows a context-aware web services recommender system as a scenario of use. The motivation is to recommend weather forecast services according to the geographical position of a specific user as time goes on. Consider that the system includes 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= "US National Weather Service[1]", S_2= "NYC Severe Weather[2]"; S_3 = "Weather China[3]"; S_4 = "Le Figaro météo in France[4]", S_5 = "Moji Weather China[5]"). Since the accuracy of weather forecast services are highly relevant to the specific region in real-time, it is natural to believe that a user prefers the service either located in his country (i.e., at home), or near the place he will migrate few days later (i.e., for a mission).

There are three layers in Fig. 1: 1) *service layer* represents the service repository available; 2) *spatial layer* illustrates migratory positions of a specific user (u_1). The curves between the first two layers link services and their physical locations correspondingly. For instance, S_1 locates in New York City and S_5 is a weather forecast service from Beijing; 3) *temporal layer* shows how the user moves from one location to another as time goes on. The arrows between the spatial and temporal layers correspond to the temporal spatial correlations of a specific user. For instance, u_1 is at New York City on January 12th, later flies to Beijing for a conference until January 16th, and finally takes a 3-day holiday in Paris.

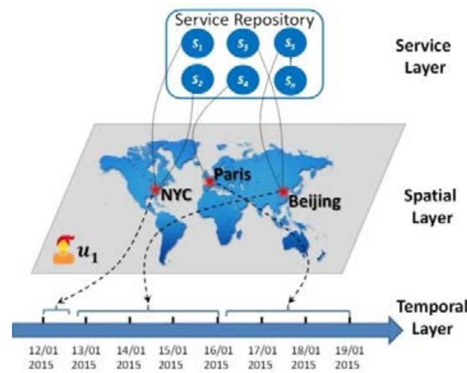

Figure 1. A scenario of weather forecast services recommendation.

In this scenario, for example in Fig. 1, u_1 is originally from NYC, thus she tends to select the native services (S_1 or S_2) when she is at home. However, the user's location could change over time from one place to another along with the movement of the user. Fig. 1 shows that u_1 would prefer the weather forecast services in China instead (S_3 or S_5) because she is aware of attending a conference in Beijing from Jan. 13th to 16th. Thus there is an awareness of temporal spatial correlations as time goes on.

It is interesting to notice that the temporal effectiveness plays a very important role to recommend personalized web services. First, the decay process of time effectiveness exists in web services recommendation. For example, the service S_1 would become inactive or not available over time, thus the impact of rating scores of S_1 will decay gradually. Second, the user preference expansion may happen when the user's location is changed over time. For instance, the observation that u_1 would prefer S_1 rather than S_3 on January 12th doesn't mean that S_1 is superior to S_3 all the time. We now elaborate on the proposed temporal effectiveness method for context-aware web services recommendation in the next section.

IV. CASR-TE ALGORITHM: CONTEXT-AWARE SERVICES RECOMMENDATION BASED ON TEMPORAL EFFECTIVENESS

In this section, we first introduce the problem definition in Section IV.A. Then, we present the four steps of proposed CASR-TE algorithm in Section IV.B, IV.C, IV.D, and IV.E.

[1] US national weather service, http://www.weather.gov/

[2] NYC Severe Weather, http://www1.nyc.gov/site/severeweather/index.page

[3] Weather China, http://en.weather.com.cn/

[4] Le Figaro météo weather forecasting service, http://www.lefigaro.fr/meteo/france/index.php

[5] Moji Weather China, http://www.moweather.com/

A. *Problem Definition*

Suppose a context-aware web services recommender system contains a set of users $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_\mathrm{n}\}$ and a set of web services $S = \{s_1, s_2, \ldots, s_m\}$. $u_i (1 \le i \le n)$ denotes a service user and he must have invoked a service from S at least once.

$W_{u_i,u_j} = \{s_1, s_2, \ldots, s_k\}$ is a set of commonly invoked web services by user u_i and u_j.

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

$R_t = \{r_{u_i,s_k,t}\}$ is the set of rating records on web service s_k by user u_i considering time decay t, where $(1 \le k \le m)$ and $(1 \le i \le n)$.

$\bar{R} = \{\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_i, \ldots, \bar{r}_n\}$ is the set of mean rating of all web services invoked by $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_\mathrm{n}\}$.

$L_{U,t} = \{l_{u_i,t}\} (1 \le i \le n)$ is the set of $l_{u_i,t}$ which is the temporal location of N dimensions 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.

$\mathrm{Q} = \{\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_\mathrm{k}\}$ is a set of QoS properties recording a service invocation.

When a service s_k is invoked by the user u_j, it will present a set of QoS properties. We will have $\mathrm{Q}_{\mathrm{i,j}} = <\mathrm{q}_1^{\mathrm{i,j}}, \mathrm{q}_2^{\mathrm{i,j}}, \ldots, \mathrm{q}_\mathrm{k}^{\mathrm{i,j}}>$, which is a *l*-tuple denoting service invocation records of s_k invoked by the user u_j, where $q_l^{k,j}$ $(1 \le k \le m, 1 \le j \le n)$ denotes the value of *l-th* property recorded during the invocation of s_k called by u_j.

B. *An Enhanced Temporal Decay Model*

Similarity measurement methods, such as Pearson Correlation Coefficient (PCC) and its variants [16, 17], are widely used in web services recommendation. However, those similarity models ignore the impact of timespan of service invocation, which is an important contextual factor because the timespan is highly related to the QoS performance of web services. For example, a longer timespan may imply a deviation of QoS value, due to service shutdown, network failures, etc.

Inspired by existing temporal decay model [6], we propose an enhanced temporal decay model by combining the decay function with similarity measurement methods. The fundamental principle of decay model is that more recent invocation from two users on the same service may have greater impacts on the user similarity measurement.

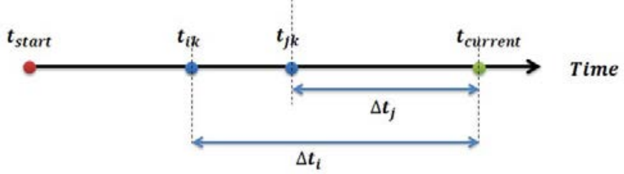

Figure 2. An example of temporal decay model.

As shown in Fig. 2, t_{ik} and t_{jk} are the time points when service s_k was invoked by user u_i and u_j. Δt_i is the time span between u_i's invocation on the service s_k and the current time $t_{current}$. Similarly, Δt_j is the time span between u_j's invocation on service s_k and the current time. We utilize $\Delta t = (\Delta t_i + \Delta t_j)/2$ to denote the factor of temporal decay. Thus, we consider that the contribution of s_k would decay manyfold with the increase of Δt. The decay function is defined as:

$$f(t_{ik}, t_{jk}) = e^{-\alpha |t_{current} - \Delta t|}, \tag{1}$$

where $\alpha \ge 0$ is a positive decay constant. Furthermore, we assume that $r_{u_i,s_k,t}$ describes results of the past rating records with temporal decay:

$$r_{u_i,s_k,t} = r_{u_i,s_k} f(t_{ik}, t_{jk}), \tag{2}$$

Here, we combine the decay function (1) with the similarity measurement method PCC [16] to describe the temporal decay model. Thus the PCC can be defined as:

$$\mathrm{sim}(\mathrm{u_i}, \mathrm{u_j}, \mathrm{t}) = \frac{\sum_{\mathrm{s_k} \in \mathrm{W_{u_i,u_j}}} \left(\mathrm{r_{u_i,s_k,t}} - \bar{\mathrm{r}}_{\mathrm{u_i}}\right)\left(\mathrm{r_{u_j,s_k,t}} - \bar{\mathrm{r}}_{\mathrm{u_j}}\right)}{\sqrt{\sum_{\mathrm{s_k} \in \mathrm{W_{u_i,u_j}}} \left(\mathrm{r_{u_i,s_k,t}} - \bar{\mathrm{r}}_{\mathrm{u_i}}\right)^2} \sqrt{\sum_{\mathrm{s_k} \in \mathrm{W_{u_i,u_j}}} \left(\mathrm{r_{u_j,s_k,t}} - \bar{\mathrm{r}}_{\mathrm{u_j}}\right)^2}}, \tag{3}$$

where w_{u_i,u_j} is a set of commonly invoked web services by the target user u_i and another user u_j, s_k is an arbitrary web service from w_{u_i,u_j}, and $\bar{r}_{u_i}$ represents the mean rating of all web services invoked by u_i. If U indicates the whole user set, we can select the users with the temporal decay similarity to constitute the similar user set $T(u_i)$ by the formula (3).

C. *Location-aware Similarity Mining with Temporal Effects*

For location-aware services recommendation, the more similar between the current user and another user's context, the more probability of the two users will have similar QoS on the same web service. However, the location information that a user submits its service invocation request may change over time.

When the current user's location is changed as time goes on, it is vital to mine users that are under the similar spatial contexts. We first capture the spatial information $S(t)$ of a user, based on the indications from calendar application of his/her mobile phone.

$$S(t) = Get(phone), \tag{4}$$

Second, we use the distance to describe the similarity of two users' location. The nearer the distance is, the more similar they are. We can use the Euclidean distance to describe the distance between $l_{u_i,t}$ and $l_{u_j,t}$:

$$Sim\left(l_{u_i,t}, l_{u_j,t}\right) = \sqrt{\sum_{k=1}^{N} (l_{i,t,k} - l_{j,t,k})^2}, \tag{5}$$

Finally, we calculate the distances between current user's location and the centroid location of each cluster users. Therefore, the nearest distance represents that the location of the users in the cluster is the closest with the location of the current user.

D. A Temporal Spatial Effectiveness Model on User Preference

According to the above two steps (Section IV.B and IV.C), we know that: 1) the contribution of past service invocations would decay during a long timespan; and 2) a user will be in different locations as time goes on. In this step, we aim to model the temporal spatial correlations when the user preference is expanded accordingly.

As what has mentioned in Section III, the accuracy of weather forecast services are highly relevant to the specific region in real-time (known as "regional correlation"). We assume $P_{RC\ S(t)}$ is the impact of regional correlation on user preference:

$$\mathrm{P}_{\mathrm{RC\ S(t)}} = \begin{cases} 1 & \text{if web service is related to region} \\ 0 & \text{if web service is not related to region} \end{cases}, \tag{6}$$

For region-irrelated services, because Internet application performance such as response time and throughput, are largely dependent on network distance between users and services (mainly because of transfer delay), it's also reasonable for users to have a preference to services near to his/her region.

We define $P_{ND\ S}$ as the network distance's influence on user preference:

$$P_{ND\ S(t)} = P_0 Dis(l_{u_i,t}, l_{s_k})_{nor}, \tag{7}$$

where P_0 is a constant and here we assign 1 to it according to our need. $Dis(l_{u_i,t}, l_{s_k})$ denotes the network distance between the user's network location $l_{u_i,t}$ and the service's network location l_{s_k}. Furthermore, the $Dis(l_{u_i,t}, l_{s_k})$ can be measured by network distance measurement technology. $Dis(l_{u_i,t}, l_{s_k})_{nor}$ is the normalization of $Dis(l_{u_i,t}, l_{s_k})$.

We also assign different weights to the impact of both regional correlation and network distance (w_1 to $P_{RC\ S(t)}$ and w_2 to $P_{ND\ S(t)}$). As a result, temporal spatial effectiveness on user preference $P_{S(t)}$ can be described as:

$$P_{S(t)} = w_1 P_{RC\ S(t)} + w_2 P_0 \mathrm{Dis}(\mathrm{l}_{u_i,t}, \mathrm{l}_{s_k})_{\mathrm{nor}}, \tag{8}$$

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

E. QoS Predication and Services Recommendation

Finally in this step, we use the past invocation records of web services dynamically from all of similar users and services to make QoS prediction and services recommendation accordingly.

We first use Bayesian inference for QoS predication, as we would like to consider not only the current situation, but also the past experience and knowledge. The formula of Bayesian inference is defined as:

$$P(OS = 1|s_i) = \frac{P(s_i|OS=1) * P(OS=1)}{P(s_i)}, \tag{9}$$

where $P(OS = 1|s_i)$ 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.

In order to explain our formula, we then set a threshold $q = 0.7$ for the QoS of a service. 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 I. 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 I shows an example of service invocation records, where each triple $\langle s_i, u_j, n\rangle$ represents a n-th service invocation of s_i by the user u_j. 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:

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

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

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

Finally, we can make the QoS prediction for the user s_1 by 2/3. Later, according to the results of each 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 detailed experimental results will be explained in Section V.

The entire procedure of CASR-TE algorithm is shown as follows.

Algorithm: Context-Aware Services Recommendation based on Temporal Effectiveness (CASR-TE)
Input: **q**, **u**, **t**, **dataset** // ***q*** *is the threshold of the QoS, different* ***q*** *will have an impact on the result;* ***u*** *is the test user and we will give a MAE/RMSE of the prediction result for the user;* ***t*** *is the*

time of recommendation; ***dataset*** *represents the training dataset.* //

Output: A group of values of MAE and RMSE with different QoS threshold **q**.

1. Start
2. **If (t ==t_1)**
3. // t_1 *represents the current time* //
4. $\mathrm{T1}(u_i)$= $\mathrm{simUPCC}(u_i, u_j, t_1, dataset)$
5. // *get the QoS similar users dataset* $T1(u_i)$ *using PPC with time decay* //
6. $\mathrm{LT1}(u_i)$ = LASM ($\mathrm{S}(t_1)$, $\mathrm{T1}(u_i)$);
7. // *get the dataset* $LT1(u_i)$ *of location-aware similar users according to the current location* $\mathrm{S}(t_1)$ *of the current user* //
8. $P_1 = P_{s(t_1)}$;
9. //*get the set* P_1 *of* web services corresponding to user current preference $P_{s(t_1)}$//
10. $\mathrm{PLT1}(u_i)$ = filtered(P_1, $\mathrm{LT1}(u_i)$);
11. //*get the filtered* dataset $\mathrm{PLT1}(u_i)$*according to the preference set* P_1//
12. for (different q)
13. preQoS =Beyesian (q, $\mathrm{PLT1}(u_i)$);
14. // *get the different predicted preQoSs of different q from the filtered* dataset $\mathrm{PLT1}(u_i)$ //
15. end for
16. mae = MAE (preQoS); rmse=RMSE(preQoS);
17. // *For different recQoSs, we can calculate different MAE and RMSE values of q. The smallest values of mae and rmse are the best.* //
18. **Else If (t ==t_2)**
19. // t_2 *represents the time after* t_1 //
20. $\mathrm{T2}(u_i)$ = $\mathrm{simUPCC}(u_i, u_j, t_2, dataset)$
21. // get the QoS similar users dataset $\mathrm{T2}(u_i)$ using PPC with time decay //
22. $\mathrm{LT2}(u_i)$ = LASM ($\mathrm{S}(t_2)$, $\mathrm{T2}(u_i)$);
23. // *get the dataset* $\mathrm{LT2}(u_i)$ *of location-aware similar users according to the current location* $\mathrm{S}(t_2)$ *of the current user* //
24. $P_2 = P_{s(t_2)}$;
25. //*get the set* P_2 *of* web services corresponding to user current preference $P_{s(t_2)}$//
26. $\mathrm{PLT2}(u_i)$ = filtered(P_2, $\mathrm{LT2}(u_i)$);
27. //*get the filtered* dataset $\mathrm{PLT2}(u_i)$*according to the preference set* P_2//
28. for (different q)
29. preQoS =Beyesian (q, $\mathrm{PLT2}(u_i)$);
30. // *get the different predicted preQoSs of different q from the filtered* dataset $\mathrm{PLT2}(u_i)$ //
31. end for
32. mae = MAE (preQoS); rmse=RMSE(preQoS);
33. // *For different recQoSs, we can calculate different MAE and RMSE values of q. The smallest values of mae and rmse are the best.* //
34. **End If**

V. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of proposed CASR-TE method. Seven compared algorithms including CASR-TE, are evaluated on WS-Dream dataset by evaluation metrics of both MAE and RMSE.

A. *Datasets and Data Processing*

We have adopted the WS-Dream [19] dataset 1[6], 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.

The raw data must be normalized before use. We adopted Gaussian approach 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:

$$\mathrm{q}_{\mathrm{l}}^{\mathrm{k,j}} = 0.5 + (\mathrm{q}_{\mathrm{l}}^{\mathrm{k,j}} - \overline{\mathrm{q}_{\mathrm{l}}^{\mathrm{j}}})/(2 * 3\sigma_{\mathrm{j}}), \tag{10}$$

where $\overline{\mathrm{q}_{\mathrm{k}}^{\mathrm{l}}}$ denotes the arithmetic mean of QoS data collected from user u_j on the *l*-th QoS property, σ_j is the standard deviation of user u_j's QoS data on *l*-th property, and $3\sigma_j$ is used according to the $3\text{-}\sigma_{\mathrm{j}}$ rule, which declares that the probability of the normalized value being in the range of $[0, 1]$ is approximately 99%.

Furthermore, since Response HTTP Code and Message are highly related, we omit the property Response HTTP Code in the evaluation of the overall QoS of a service. Thus the weight formula is:

$$\mathrm{QoS} = \mathrm{w}_1 * \mathrm{v}_{\mathrm{RTT}} + \mathrm{w}_2 * \mathrm{v}_{\mathrm{DataSize}} + \mathrm{w}_3 * \mathrm{v}_{\mathrm{RHTTPMessage}}, \tag{11}$$

where w_1, w_2 and w_3 are set to 0.35, 0.05 and 0.6 respectively according to their different significance. The threshold q is used to determine whether a service is deemed as satisfying can be set to different values according to different situations. By evaluating the overall QoS of a service, we can simulate the feedback of a user after invoking a service.

In addition, the simulation and experiment are developed by the MATLAB 2013, and conducted on a LENOVE E430c PC with Intel Core I5 2.50GHz CPU, 4GB RAM, and Windows 7 operating system.

B. *Evaluation Metrics*

In each category of WS-Dream dataset, we divide the dataset into 15 segments, in which various ratios of training and testing dataset will be discussed later. The evaluation

[6] WS-Dream dataset, http://www.wsdream.net/dataset.html

metrics [16] we use in our experiments are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE):

$$MAE = \frac{\sum_{u,s}|Q_{u,s} - \hat{Q}_{u,s}|}{N}, \tag{12}$$

$$RMSE = \frac{\sqrt{\sum_{u,s}\left(Q_{u,s} - \hat{Q}_{u,s}\right)^2}}{N}, \tag{13}$$

In the formula (12) and (13), $Q_{u,s}$ denotes actual QoS values of 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.

C. Evaluation

1) Comparative Algorithms

We conducted series of experiments to compare our *CASR-TE* algorithm with the following existing algorithms:

- **RBA** (Recommendation by all): the method recommends web service to a user collected by all users without a filtering.
- **UPCC** [16]: the method recommends services to a user collected by other users sharing the similar preference. The similarity between users is calculated by PCC based on user profiles.
- **IPCC** [17]: the method recommends services to a user similar to the ones the user preferred in the past. The similarity between services is calculated by PCC based on services.
- **CASR** [4]: the method makes recommendation based on the service invocation experiences under similar spatial context with the current user.
- **CASR-UP** [18]: the method makes recommendation considering the user preference determined by user's location.
- **ITRP-WS** [20]: the method considers the time decay effects in UPCC and makes recommendation for users.

2) Performance Comparison

For different algorithms, we firstly show the results of MAE and RMSE which are generated in different threshold q (from 0.65 to 0.95) in the ratio 14:1 of training dataset and test dataset in Fig. 3. The number of the neighbors in UPCC, IPCC, ITRP-WS and CASR-TE is set to 5. From Fig. 3, we could see that: 1) when the threshold $q \leq 0.925$, the accuracy of our CASR-TE is much better than other six algorithms; and 2) when the threshold $q = 0.95$, the accuracy of our CASR-TE is abnormal. The reason is that as the threshold q rises up to 0.95, most of positive services will be excluded, which results in the very high MAE and RMSE. In general, the results demonstrate that the significant of CASR-TE algorithm in recommending web services considering the temporal effectiveness.

In addition, we also show the average MAE/RMSE results of seven algorithms in different ratios (8:7, 9:6, 10:5, 11:4, 12:3, 13:2, and 14:1) of training dataset and testing dataset. The number of the neighborhoods in UPCC, IPCC, ITRP-WS and CASR-TE is set to 5. In Fig. 4, we can see that: 1) The MAE and RMSE results of seven algorithms generally speaking decrease as the ratio of training and test data increase. This means that more training data will help gain more accurate evaluation; and 2) in different ratios, the results of our CASR-TE algorithm performs better than the other six algorithms.

Finally in Fig. 5, we compare the results of UPCC, IPCC, ITRP-WS and CASR-TE with different numbers of neighbors (e.g., 5, 6, 7, 8, 9, 10 and 11) in the ratio 14:1 of training and test dataset. We can conclude that 1) the results of CASR-TE outperform three compared algorithms regardless of the number of neighbors; and 2) with the increase of the number of neighbors, the MAE/RMSE results become better gradually.

D. Discussion

Comparing the performance of the proposed CASR-TE algorithm with the six comparative algorithms, we will discuss two aspects in our experiments: adjustment of the trade-off parameters, and the impact of temporal effectiveness.

1) Trade-off parameters

Fig. 3 shows the accuracy of QoS prediction when the threshold q changes while the ratio of training and test part remains invariable (i.e. 14:1). As we can see, when q is changed, the results of MAE and RMSE will be different accordingly. From the results, we can infer that: **1)** when the threshold $q \leq 0.925$, the MAEs and RMSEs of CASR-TE are smaller than other algorithms; **2)** But why $q = 0.95$ is an exception? By analyzing our CASR-TE method, we can find that after selecting some web services according to the dynamic user preference, the invocation records of the selected services are more useful. As the threshold q rises, most of positive services will be excluded and the result becomes abnormal; **3)** We can also see that when $q \leq 0.725$, the MAEs and RMSEs of the algorithms remain almost invariable. We could explain that when the threshold q is low, many negative services will be included. When q is low enough, all web services will be included, thus the MAEs and RMSEs remain invariable; and **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 services will be included, while if q is too high, many positive services will be excluded. According to our algorithm, the best q is approximately 0.775.

Fig. 4 shows the influence of different ratios on the 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, so the accuracy will be better.

In addition, Fig. 5 shows the influence of neighbors' number on the results. We can see that the results of UPCC, IPCC, ITRP-WS are almost invariable when the N changes, but the results of CASR-TE become better when the N arises. It's no wonder that more neighbors will contribute to better results, but when N is large enough, the results will be bad instead because not very useful items will be included.

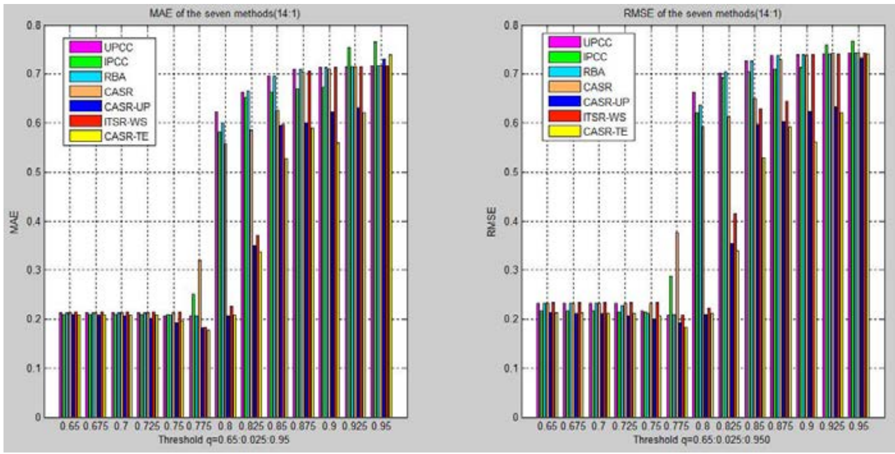

Figure 3. MAE and RMSE results of of compared methods (14:1).

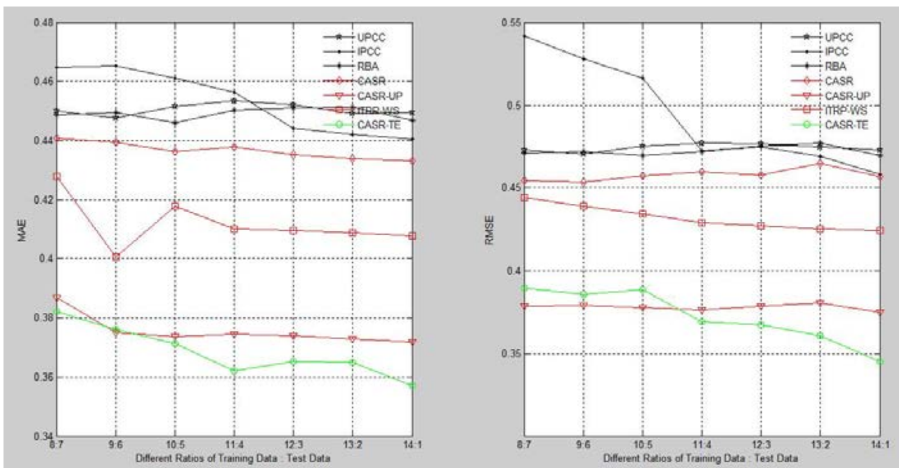

Figure 4. MAE and RMSE results of compared methods (in various ratios).

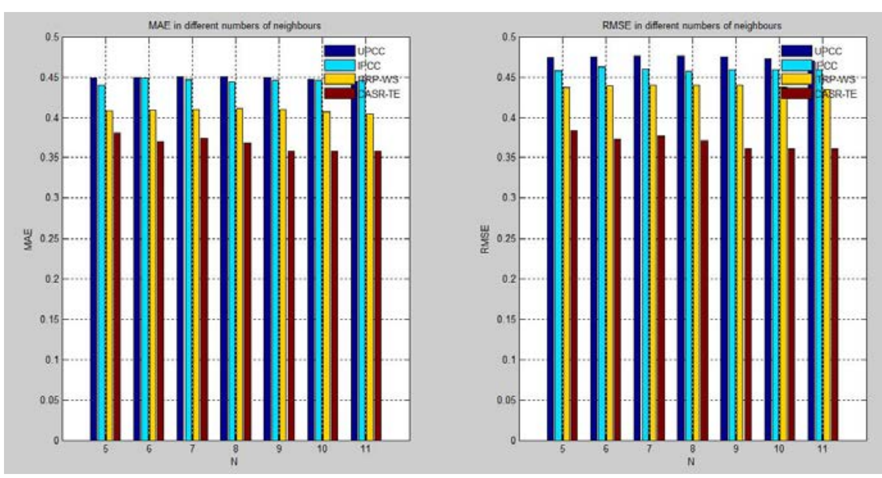

Figure 5. MAE and RMSE results of UPCC, IPCC, ITRP-WS and CASR-TE (in different numbers of neighbour).

2) *Impacts of Temporal Effectiveness*

We have obtained several preliminary results of the impact of temporal effectiveness on recommendation accuracy. The results shown in the Fig. 3, 4 and 5 collectively demonstrate that: 1) the temporal decay model is effectively integrated into the proposed context-aware web services recommendation model; and 2) the fusion of temporal spatial correlations on user preference could further improve the accuracy of recommendation.

VI. CONCLUSION

In this paper, we propose a *CASR-TE* method, for modeling enhanced temporal decay effectiveness and temporal spatial correlations. Also, the experiments results show that CASR-TE algorithm improves predictive accuracy and outperforms the compared methods.

Despite the significant progress of temporal effects in web service recommendation, there still remain numerous avenues to explore. According to the experimental results of Section V, our future works might include: 1) associating the time decay function with other similarity measurement methods; and 2) incorporating the socio-temporal correlations to improve the accuracy of QoS prediction in location-based social networks.

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); and China Telecom Corp. Gansu Branch Cuiying Funds (lzudxcy-2014-6)Fundamental Research Funds for the Central Universities (lzujbky-2015-100).

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, Context-aware recommender systems: Recommender systems handbook, Springer, US, 2011.

[2] P. Dourish, “What we talk about when we talk about context,” Personal and Ubiquitous Computing, vol. 8, no. 1, 2003, pp. 19–30, doi: 10.1007/s00779-003-0253-8.

[3] D. Jannach, T.U. Dortmund, and G. Friedrich, “Tutorial: Recommender Systems,” Proc. International Joint Conference on Artificial Intelligence Beijing (IJCAI 13), August 2013.

[4] L. Kuang, Y. Xia, and Y. Mao, “Personalized Services Recommendation Based on Context-Aware QoS Prediction,” Proc. IEEE International Conference on Web Services (ICWS 12), IEEE Press, June 2012, pp. 400-406, doi: 10.1109/ICWS.2012.12.1.

[5] W. Zhang, H. Sun, X. Liu, and X. Guo, “Incorporating Invocation Time in Predicting Web Service QoS via Triadic Factorization,” Proc. IEEE International Conference on Web Services (ICWS 14), IEEE Press, June 2014, pp.145-152, doi: 10.1109/ICWS.2014.32.

[6] Y. Hu, Q. Peng, and X. Hu, “A Time-Aware and Data Sparsity Tolerant Approach for Web Service Recommendation,” Proc. IEEE International Conference on Web Services (ICWS 14), IEEE Press, June 2014, pp. 33-40, doi:10.1109/ICWS.2014.18.

[7] Y. Zhang, Y Fan, K. Huang, W. Tan, and J. Zhang, “Time-Aware Service Recommendation for Mashup Creation in an Evolving Service Ecosystem,” Proc. IEEE International Conference on Web Services (ICWS 14), IEEE Press, June 2014, pp. 25-32, doi:10.1109/ICWS.2014.17.

[8] X. Zhang, K. He, J. Wang, C. Wang, G. Tian, and J. Liu, “Web Service Recommendation Based on Watch list via Temporal and Tag Preference Fusion,” Proc. IEEE International Conference on Web Services (ICWS 14), IEEE Press, June 2014, pp. 281-288, doi: 10.1109/ICWS.2014.49.

[9] M. Tang, Y. Jiang, J. Liu, and X. Liu, “Location-Aware Collaborative Filtering for QoS-Based Service Recommendation,” Proc. IEEE International Conference on Web Services (ICWS 12), IEEE Press, June 2012, pp. 202-209, doi: 10.1109/ICWS.2012.61.

[10] B. Cao, J. Liu, M. Tang, Z. Zheng, amd G. Wang, “Mashup Service Recommendation based on User Interest and Social Network,” Proc. IEEE International Conference on Web Services (ICWS 13), IEEE Press, June 2013, pp. 99-106, doi: 10.1109/ICWS.2013.23.

[11] P. He, J. Zhu, Z. Zheng, and J. Xu, “Location-based Hierarchical Matrix Factorization for Web Services Recommendation,” Proc. IEEE International Conference on Web Services (ICWS 14), IEEE Press, June 2014, pp. 297-304, doi: 10.1109/ICWS.2014.51.

[12] J. McCarthy, “Notes on formalizing context,” Proc. International Joint Conference on Artificial Intelligence (IJCAI 93), vol. 1, 1993, pp. 555-560.

[13] P. Brézillon, “Task-realization models in contextual graphs,” Proc. Modeling and Using Context: 5th International and Interdisciplinary Conference (Context 05). Lecture Notes in Computer Science, vol. 3554/2005, 2005, pp. 55-68, doi: 10.1007/11508373_5.

[14] H. Gao, J. Tang, X. Hu, and H. Liu, “Modeling Temporal Effects of Human Mobile Behavior on Location-Based Social Networks,” Proc. ACM International Conference on Information and Knowledge Management (CIKM 13), ACM, 2013, pp. 1673-1678, doi:10.1145/2505515.2505616.

[15] B. Hu, M. Jamali, and M. Ester, “Spatio-Temporal Topic Modeling in Mobile Social Media for Location Recommendation,” Proc. IEEE International Conference on Data Mining (ICDM 13), IEEE press, 2013, pp. 1073-1078,doi: 10.1109/ICDM.2013.139.

[16] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, San Francisco, CA: Morgan Kaufmann Publishers Inc., 1998.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens, “An open architecture for collaborative filtering of netnews,” Proc. ACM Conference on Computer Supported Cooperative Work (CSCW 94), 1994, pp. 175-186, doi: 10.1145/192844.192905.

[18] X. Fan, Y. Hu, and R. Zhang, “Context-aware Web Services Recommendation based on User Preference,” Proc. IEEE Asia-Pacific Services Computing Conference (APSCC 2014), December, 2014, IEEE press, pp. 55-61, doi: 10.1109/APSCC.2014.20.

[19] Z Zheng, H Ma, MR Lyu, and I King, “QoS-aware Web Service Recommendation by Collaborative Filtering,” IEEE Transactions on Services Computing, vol. 4, no. 2, 2010, pp. 140-152, doi: 10.1109/TSC.2010.52.

[20] X. Cui, G. Yin, Q. Han, and Y. Dong, “An Improved Time-effectiveness Reliability Prediction Approach of Web Service,” Journal of Computational Information Systems, vol. 10, no. 4, 2014, pp. 1365–1374, doi: 10.12733/jcis9139.