Spatial-Temporal Data-driven Service Recommendation with Privacy-preservation

Lianyong Qi[1], Xuyun Zhang[2], Shancang Li[3], Shaohua Wan[4,*], Yiping Wen[5], Wenwen Gong[6]

[1] School of Information Science and Engineering, Qufu Normal University, China
[2] Department of Electrical and Computer Engineering, University of Auckland, New Zealand
[3] Computer Science and Creative Technologies Department, University of the West of England, UK
[4] School of Information and Safety Engineering, Zhongnan University of Economics and Law, China
[5] School of Computer Science and Engineering, Hunan University of Science and Technology, China
[6] College of Information and Electrical Engineering, China Agricultural University, China

{lianyongqi@gmail.com, xuyun.zhang@auckland.ac.nz, Shancang.Li@uwe.ac.uk, shaohua.wan@ieee.org, ypwen81@gmail.com, wen.gong@cau.edu.cn}
Correspondence should be addressed to: Shaohua Wan (shaohua.wan@ieee.org)

Abstract. The ever-increasing popularity of web service sharing communities have produced a considerable amount of web services that share similar functionalities but vary in Quality of Services (QoS) performances. To alleviate the heavy service selection burden on users, lightweight recommendation ideas, e.g., Collaborative Filtering (CF) have been developed to aid users to select their preferred services. However, existing CF methods often face two challenges. First, service QoS is often context-aware and hence depends on the spatial and temporal information of service invocations heavily. While it requires challenging efforts to integrate both spatial and temporal information into service recommendation decision-making process simultaneously. Second, the location-aware and time-aware QoS data often contain partial sensitive information of users, which raise an emergent privacy-preservation requirement when performing service recommendations. In view of above two challenges, in this paper, we integrate the spatial-temporal information of QoS data and Locality-Sensitive Hashing (LSH) into recommendation domain and bring forth a location-aware and time-aware recommendation approach considering privacy concerns. At last, a set of experiments conducted on well-known WS-DREAM dataset show the feasibility of our approach.

Keywords: Service recommendation, Spatial-Temporal QoS, Locality-Sensitive Hashing, Privacy-preservation, Collaborative Filtering.

1 Introduction

With the increasing maturity and popularization of web of things, many software developers or vendors have begun to encapsulate their software tools or program interfaces into lightweight web services that can be accessed easily, and register them in various service sharing communities or platforms, e.g., ProgrammableWeb (PW)[1] and api-platform.com[2]. In this situation, a service community often contains a considerable amount of candidate web services that share same or similar functions but vary in QoS (Quality of Services) performances [1-2], which impacts the service selection decisions of users especially when a user does not have much detailed knowledge of candidate web services.

In this situation, various recommendation techniques, e.g., Collaborative Filtering (CF) have been introduced to aid the users' service selection decision-makings. Typically, CF technique can filter out massive services not preferred by a user and only recommend appropriate new services that satisfy the user's preferences; thus the search space of optimal recommended services is reduced significantly and the user's service selection burden is decreased accordingly. Benefiting from the easy-to-understand and domain-independent characteristics, CF has been widely adopted in various recommender systems [3-5], such as news recommendation, spot recommendation and E-commerce recommendation. Typically, by evaluating past service execution records (e.g., historical QoS data of candidate services), CF can effectively predict user preferences and then make accurate recommendations.

However, in the dynamic network environment, service QoS data are often not fixed but fluctuant with service execution context [6-8]. For example, a long waiting time is possible if we invoke a web service at its busy time; besides, a user in USA often invokes a web service hosted in USA quickly while invokes another web service hosted in China slowly. In other words, service QoS often depends on the spatial and temporal information of each service invocation. This location-aware and time-aware QoS variation brings a new challenge for accurate service recommendation as it is often hard to precisely predict the future QoS performance of candidate services.

Furthermore, the location-aware and time-aware QoS data often contain partial information that is sensitive to users [9-10], e.g., a larger response time for invoking a cloud storage service often indicates that a user is transmitting a big file. In this situation, the recommender systems need to secure the sensitive QoS data when analyzing the QoS data for accurate recommendation decision-makings. In other words, the recommendation process should be not only accuracy-guaranteed but also privacy-aware. However, it requires challenging efforts to achieve the above two goals simultaneously as there is often a tradeoff between the released QoS data and the recommendation accuracy.

Considering the above challenges, we take into consideration the spatial-temporal information of QoS and the Locality-Sensitive Hashing (LSH) technique and then propose a location-aware and time-aware recommendation solution with a high capability of privacy-preservation. LSH has a good property of "similarity retain": two neighboring points will be assigned the same index with high probability. Therefore, through LSH, we can convert sensitive user QoS data into less-sensitive user indices and then uses user indices to make further recommendations. This way, user privacy can be protected and meanwhile recommendation accuracy can also be guaranteed. Overall, the major academic contributions of our work are two-fold.

[1] https://www.programmableweb.com/
[2] https://api-platform.com/

(1) We combine the spatial-temporal information of historical QoS data and the LSH technique together and then introduce a recommendation solution named $Rec_{st\text{-}LSH}$. Different from the traditional CF approaches that use original and sensitive QoS data for similar friend findings, LSH employs less-sensitive user indices to search for similar friends of a target user. Therefore, private user information is protected. In other words, $Rec_{st\text{-}LSH}$ can not only output an accurate recommended list but also protect the sensitive user information hidden in historical QoS data.

(2) A set of experiments are designed based on a popular QoS dataset, i.e., WS-DREAM that contains the spatial-temporal QoS data of 4532 web services invoked by 142 users distributed in different countries. Experiment results prove the effectiveness and efficiency of our proposal.

The reminder of this paper is structured as below. In Section 2, we summarize the related work associated with the spatial-temporal and privacy-aware service recommendation problem. In Section 3, the spatial-temporal data-driven recommendation problems with privacy-preservation is formalized. In Section 4, we integrate the spatial-temporal information of historical QoS data and the LSH technique into recommendation decision-making process and propose a novel service recommendation approach named $Rec_{st\text{-}LSH}$. In Section 5, a set of experiments on real-world QoS dataset, i.e., WS-DREAM are carried out to evaluate the performances of $Rec_{st\text{-}LSH}$. Section 6 concludes the whole paper and analyzes the possible improvements in future work.

2 Related Work

Next, we review the research progress of spatial-temporal service recommendations with privacy-preservation from two perspectives: spatial-temporal data-driven recommendation and privacy-preserving service recommendation.

(1) Spatial-temporal data-driven recommendation

In [1], the correlation between service QoS and service invocation time is investigated by transforming the user-service QoS matrix into the product of two sub-matrices, i.e., user-time sub-matrix and time-service sub-matrix. After that, the derived two sub-matrices are employed to make time-aware service recommendations. Most recommender systems take the all-time statistics of user-service usage patterns as the recommendation bases, while neglect the time-aware QoS evolution in the dynamic network environment. Such a static QoS measurement often cannot reflect the latest QoS variation trend very well. In view of this shortcoming, in [2-3], Latent Dirichlet Allocation (LDA) technique is used to extract time-aware QoS evolution patterns of web services, after which a service popularity-based recommendation method is brought forth. In [4-6], traditional CF-based recommendation approaches are improved by adding a time-aware user similarity or item similarity; this way, time-aware QoS variations are successfully integrated into the recommendation decision-making process to pursue higher recommendation accuracy.

In [7], the authors investigate the correlation between service QoS and user location; afterwards, Nonnegative Tensor Factorization (NTF) technique is introduced into the traditional CF recommendations to improve the accuracy of location-based recommended POIs (Point-of-Interest). In [8], both user location and service location are taken into consideration and finally, only those similar users who are physically close to a target user are returned to make appropriate recommendations.

In [9], the authors propose a temporal decay model for user similarity in traditional CF approaches and then recruit the spatial-temporal correlations hidden in historical QoS data to extract user preferences and perform preferences-aware service recommendations. In [10], the authors utilize the spatial-temporal information generated from past service invocations to analyze user behaviors and then determine the neighboring users of a target user who have similar behavior patterns. In [11], the authors analyze the common influence of location and time factors on service QoS performances. Through experiments deployed on a real-world QoS dataset, the authors reveal the QoS data of spatially-close user-service pairs are correlated with high probability. Motivated by this observation, the authors put forward a Matrix Factorization-based recommendation approach. In [12], the users and candidate services are grouped by their respective locations; afterwards, a time weight and a location weight are assigned to user similarity or item similarity to improve the service recommendation accuracy. In [13], the authors investigate the multi-dimensional service recommendation problems where the QoS data with multiple dimensions (including time and location) are modeled as a tensor; afterwards, tensor decomposition technique is used to find out the component matrices of the QoS tensor; finally, the derived component matrices are employed to predict unknown QoS values and then make optimal recommendation decisions.

Therefore, existing research often focuses on either location-aware service recommendation or time-aware service recommendation or spatial-temporal service recommendation. However, they often lack the capability of protecting the private information of users contained in spatial-temporal QoS data. Although partial research work, e.g., [14] considers both time-aware QoS data and privacy-preservation requirement, they overlook the location factor that plays a key role in most recommender systems.

(2) Privacy-preserving service recommendation.

Obfuscation technique is adopted in [15] to protect the sensitive QoS data by adding a random number; thus the obfuscated QoS data are recruited to approximately calculate user similarity. This way, user privacy is protected; however, recommendation accuracy is decreased to some extent due to the inherent tradeoff between data availability and data privacy. In [16], the authors randomly divide each QoS data into multiple segments with little privacy; afterwards, the QoS segments are used to calculate item similarity and make recommendations. However, this privacy protection technique is only applicable to the specific PCC (Pearson Correlation Coefficient)-based similarity calculation; besides, much storage cost is incurred by the divided QoS segments. In [17], the authors propose a cloud-assisted differentially private video recommendation system based on distributed online learning technique, to secure the sensitive data involved in recommendation process. However, the time complexity of differential privacy is often high; besides, the accumulated noise data by differential privacy techniques can decrease the data availability and further influence the accuracy of the final recommended list.

In our previous work [18-22], various hash techniques are used to achieve the goal of privacy protection in distributed recommendation process, e.g., LSH technique in [18-20], SimHash technique in [21] and MinHash technique in [22]. These hash techniques can effectively balance the recommendation performances in terms of accuracy and privacy-preservation. Although the above literatures can protect user privacy involved in recommendation process, they often suppose service QoS is fixed, without considering the QoS fluctuation incurred by the frequent change of time and location. Motivated by this fact, a novel recommendation solution named $Rec_{st\text{-}LSH}$ is brought forth as elaborated in the following sections.

3 Formulation and Problem Specification

In this section, we formulate the spatial-temporal service recommendation problems with privacy that we focus on in this paper. Inspired by the three-dimensional user-service-time QoS representation in [14], we model the spatial-temporal QoS data as a four-dimensional vector QoS = (user, service, time, location) depicted intuitively in Fig.1. In Fig.1, different locations are remarked in distinct colors; each QoS data is represented by a four-dimensional point $q_{i,j,k,d}$ where dimensions i, j, k, d denote user, service, time and location, respectively.

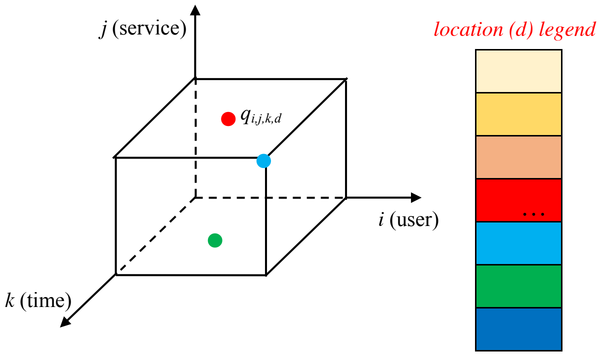

Fig.1 Four-dimensional QoS model.

For formal discussions in the rest of paper, we make the following specifications.

(1) web service set. The set of candidate web services in a service sharing community, denoted by $WS\_Set = \{ws_1, \ldots, ws_n\}$

(2) user set. The set of users who have executed any web service in WS_Set previously, denoted by $U\_Set = \{u_1, \ldots, u_m\}$.

(3) time slot set. The set of time slots when a web service was invoked by users in U_Set, denoted by $T\_Set = \{t_1, \ldots, t_P\}$. For simplicity, we assume that the time slots for $ws_1, \ldots, ws_n$ are the same.

(4) user location. user u's locations where u invoked a service, denoted by $LOC\_Set = \{loc_1, \ldots, loc_Q\}$.

(5) service quality. Service QoS is denoted by a four-dimensional $q_{i,j,k,d}$ that indicates the quality of service ws_j ($\in WS\_Set$) by user u_i ($\in U\_Set$) from location loc_d ($\in LOC\_Set$) at time slot t_k ($\in T\_Set$).

(6) target user. u_{target} ($\in U\_Set$) denotes a target user who requests recommended service items.

Generally, the historical QoS data $q_{i,j,k,d}$ ($1 \le i \le m$, $1 \le j \le n$, $1 \le k \le P$, $1 \le d \le Q$) are recorded by different platforms or servers. These distributed QoS data are the major bases for the subsequent recommendation decision-makings. So according to the above formal specifications, we can describe the spatial-temporal data-driven recommendation problems with privacy-preservation as follows: a recommender system integrates and analyzes the historical QoS data $q_{i,j,k,d}$ distributed across multiple platforms, and then returns u_{target} a set of new services that may be preferred by u_{target}, during which the sensitive QoS data $q_{i,j,k,d}$ in different platforms are secure. To solve the above challenge, we propose a novel spatial-temporal data-driven recommendation approach named $Rec_{st\text{-}LSH}$ in Section 4.

4 Our Solution: $Rec_{st\text{-}LSH}$

In this section, we clarify the details of our proposed $Rec_{st\text{-}LSH}$ approach. The basic idea behind $Rec_{st\text{-}LSH}$ is: first, we build time-aware user indices (with little even no privacy) according to the user-service-time QoS matrix; second, according to the less-sensitive user indices, determine the similar users of u_{target}; third, weight each similar user according to his or her location information; at last, we predict the unknown QoS in the user-service-time QoS matrix and make appropriate recommendations. Concretely, $Rec_{st\text{-}LSH}$ approach includes four steps, as elaborated in Fig.2. Here, please note that Step-1 is executed by each platform or server hosting sensitive QoS data; while Step-2 ~ Step-4 are executed by the recommender system.

Step-1: Time-aware user indices building. According to the user-service-time QoS matrix M, build L time-aware user index tables Tb_1, …, Tb_L with less privacy based on LSH technique.

Step-2: Similar user finding for u_{target}. According to the less-sensitive user index tables Tb_1, …, Tb_L, look for the similar users of u_{target} and put them into set $Sim\_U(u_{target})$.

Step-3: Similar user weighting. For each user u_i in set $Sim\_U(u_{target})$, we weight him or her according to the distance between u_i and u_{target}.

Step-4: Service quality prediction & service recommendation. In the user-service-time matrix, according to the known QoS data of the derived similar users of u_{target}, we predict the unknown QoS data of u_{target} and finally return u_{target} a service $ws_{optimal}$ with the optimal QoS prediction value at current time slot.

Fig.2 Four steps of $Rec_{st\text{-}LSH}$ approach

Step-1: Time-aware user indices building.

In this step, we build less-sensitive user indices based on the original user-service-time QoS matrix (denoted by M). As specified in Section 2, the QoS matrix involves four dimensions, i.e., user, service, time and location, which is often difficult to describe with a matrix. Considering this, we simply depict the QoS matrix with two dimensions (i.e., service and time): each user u_i corresponds to a service-time QoS matrix $M(u_i)$ (see equation (1)) and u_i's location information is not considered temporally.

$$M(u_i) = \begin{array}{c} \\ ws_1 \\ \vdots \\ ws_n \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & \cdots & t_P \end{array} \\ \begin{bmatrix} q_{i,1,1} & \cdots & q_{i,1,P} \\ \vdots & \ddots & \vdots \\ q_{i,n,1} & \cdots & q_{i,n,P} \end{bmatrix} \end{array} \tag{1}$$

Next, we generate a $P*n$ LSH function matrix, denoted by LSH_M as in (2) where $x_{i,j}$ is randomly chosen from range [-1, 1]. Then we calculate the product of matrices $M(u_i)$ and LSH_M by (3), whose results are represented by an $n*n$ matrix A. In matrix A, we only record the elements in the diagonal line and calculate their sum as in (4). Next, we transform the sum value in (4) into a binary value $sum^{\#}$ in (5) according to the transformation rule of LSH.

$$LSH\_M = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{P,1} & \cdots & x_{P,n} \end{bmatrix} \tag{2}$$

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix} = M(u_i) * LSH\_M = \begin{bmatrix} q_{i,1,1} & \cdots & q_{i,1,P} \\ \vdots & \ddots & \vdots \\ q_{i,n,1} & \cdots & q_{i,n,P} \end{bmatrix} * \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{P,1} & \cdots & x_{P,n} \end{bmatrix} \tag{3}$$

$$sum = \sum_{i=1}^{n} a_{i,i} \tag{4}$$

$$sum^{\#} = \begin{cases} 1, & \text{if } sum > 0 \\ 0, & \text{if } sum \le 0 \end{cases} \tag{5}$$

Next, we repeat the above process to generate r LSH_M matrices, i.e., LSH_M_1, …, LSH_M_r, and derive r $sum^{\#}$ values, i.e., $sum_1^{\#}$, …, $sum_r^{\#}$. Then the r-dimensional vector $H_i = (sum_1^{\#}, \ldots, sum_r^{\#})$ can be regarded as the index of user u_i. Thus, the projections from the original and sensitive QoS matrix of u_i ($1 \le i \le m$), i.e., $M(u_i)$ to the less-sensitive index value H_i form a hash table (denoted by Tb). In the reminder of our approach, the recommender system only uses less-sensitive user indices H_1, …, H_m for recommendation decisions without revealing the real and sensitive QoS values in original matrix $M(u_1)$, …, $M(u_m)$. Therefore, user privacy involved in the recommendation process is secured. Concrete details are specified in Algorithm 1.

Algorithm 1: ***user_index building***

Input:

$u_1, \ldots, u_m$: user set
r: number of LSH functions
$ws_1, \ldots, ws_n$: candidate service set
M: user-service-time QoS matrix

Output:

$H_1, \ldots, H_m$: time-aware user indices

```
1  for i = 1 to m do
2      generate service-time QoS matrix M(u_i) from M
3  end for
4  for i = 1 to P do
5      for j = 1 to n do
6          x_{i,j} = random (-1, 1)
7      end for
8  end for
9  for i = 1 to m do
10     calculate matrix A by (3)
11     calculate sum by (4)
12     calculate sum^# by (5)
13 end for
14 repeat Lines 4~13 r times
15 for i = 1 to m do
16     return H_i
17 end for
```

Step-2: Similar user finding for u_{target}.

In Step-1, we have obtained the index value H_i for each user u_i ($1 \leq i \leq m$). According to the similar object judgment rule in LSH, we can draw a conclusion that u_i and u_{target} are similar if their indices are equal, i.e., $H_i = H_{target}$. However, LSH is a probability-aware search manner; therefore, the above similar user search condition is often a bit rigid and may produce false-negative search results. In view of this shortcoming, we relax the search condition of $H_i = H_{target}$. Concretely, we repeat Step-1 multiple times to generate L ($L > 1$) hash tables $Tb_1, \ldots, Tb_L$. If $H_i = H_{target}$ holds in any of the L hash tables, then the two users u_i and u_{target} could be regarded as similar. In other words, the equation in (6) holds.

$$u_i \text{ is similar with } u_{target} \text{ iff } H_i = H_{target} \text{ holds in any } Tb_z \ (z = 1, \ldots, L) \qquad \textbf{(6)}$$

Next, we search for all the similar users of the target user u_{target} through (6) and put them into a new set $Sim\_U(u_{target})$. The pseudo code of Algorithm 2 specifies the details of this step.

Algorithm 2: ***Similar user finding for u_{target}***

Input:
$u_1, \ldots, u_m$: user set
$H_1, \ldots, H_m$: user indices
L: number of LSH tables
u_{target}: a target user
Output:
$Sim\_U(u_{target})$

```
repeat Algorithm 1 to generate L hash tables Tb_1, …, Tb_L
for i = 1 to m do
    for z = 1 to L do
        if H_i = H_target holds in Tb_z
            then do put u_i into Sim_U(u_target)
                    break
        end if
    end for
end for
return Sim_U(u_target)
```

Step-3: Similar user weighting.

In Step-2, we have derived a set of similar users of u_{target}, i.e., the users in set $Sim\_U(u_{target})$. Assume $Sim\_U(u_{target})$ include K similar users of u_{target}, denoted by $u_1, \ldots, u_K$, respectively. The next question is to rank their respective importance in the subsequent QoS missing-value prediction. A promising way is to rank and weight all the similar users of u_{target} according to their location information as location is a key factor that influences the service running quality heavily (discussed in Section 1).

Concretely, we use the Sigmod function in (7) to model the weights of the K similar users of u_{target}. In (7), w_i means the weight of u_i $(1 \leq i \leq K)$, $D(u_i, u_{target})$ measures the distance between the locations of u_i and u_{target}. Concrete weight function is presented in Fig.3. As the weight sum of K similar users is equal to 1, we replace w_i by $w_i / \sum w_i$.

$$w_i = 1 - \frac{1}{1+e^{-D(u_i,\, u_{target})}} \tag{7}$$

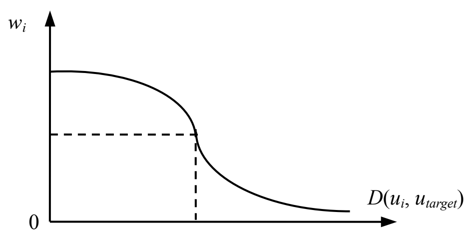

Fig. 3 Location-based weighting for similar users of u_{target}

Step-4: Service quality prediction & service recommendation.

In this step, we use the similar users (i.e., u_1, …, u_K) of u_{target} obtained in Step 2 and the user weights obtained in Step 3 to predict the missing QoS data in the user-service-time matrix M. In concrete, for u_{target}, its missing QoS data on service ws_j at time slot t_k, i.e., $q_{target,j,k}$ can be calculated by (7). At last, we choose the web service $ws_{optimal}$ with optimal predicted quality by (8) and return $ws_{optimal}$ to u_{target}.

$$q_{target,j,k} = \sum_{u_i \in Sim\_U(u_{target})} w_i * q_{i,j,k} / K \quad (7)$$

$$ws_{optimal} = \{ws_j \mid ws_j \in WS\_Set \text{ and } q_{target,j,k} = \text{optimal}\{ q_{target,j,k} \}\} \quad (8)$$

Thus, with the above four steps of $Rec_{st\text{-}LSH}$, a recommender system can produce accurate recommended results for u_{target} in a privacy-preserving way due to the following two reasons: first, we predict the missing service QoS of a target user at a time slot according to the known QoS of the similar users at the same time slot, which ensures a relatively high recommendation accuracy; second, the similar users of u_{target} are determined based on less-sensitive user indices H_i, …, H_m, instead of sensitive QoS matrix M, which guarantee the capability of $Rec_{st\text{-}LSH}$ in terms of privacy protection.

5 Experiments

5.1 Experiment Settings

We take the well-known QoS dataset WS-DREAM as our experiment data source. This dataset contains the historical QoS data of 4532 services monitored by 142 users at 64 time slots. The user location information can be obtained by analyzing the user IP. Therefore, this dataset contains the spatial-temporal information of QoS and hence can be used to prove the feasibility of $Rec_{st\text{-}LSH}$. We compare $Rec_{st\text{-}LSH}$ with three state-of-the-art ones: $SerRec_{time\text{-}LSH}$ [5], $SerRec_{distri\text{-}LSH}$ [18] and benchmark *UPCC* [23]. Experiment parameters are specified in Table 1.

Experiments were running on a pc with 2.60 GHz processor and 8.0 GB memory. The software configurations are Windows 10 and Python 3.6. Tests are repeated 100 times and average running results are reported finally for display. Detailed running results are represented in the next subsection.

Table 1. Specifications of experiment parameters

symbol	meaning	values
m	number of users	142
n	number of web services	4532
P	number of time slots	64
L	number of hash tables	4, 6, 8, 10
r	number of hash functions	4, 8, 12, 16
β	density of QoS matrix	1%

5.2 Experiment Results

Profile-1: Accuracy comparison of four approaches.

Accuracy of recommended service items is a core criterion to evaluate the performance of different recommender systems. Here, we measure the accuracy (through RMSE, smaller is better) of four solutions. Parameter L = 10, r = 4. Experiment comparison results are presented in Fig.4.

We can observe from Fig.4 that the RMSE values of the three LSH-based solutions (i.e., $Rec_{st\text{-}LSH}$, $SerRec_{time\text{-}LSH}$ and $SerRec_{distri\text{-}LSH}$) are all lower than that of *UPCC* due to the inherent characteristics of LSH in term of similar object search. Moreover, the RMSE value of $Rec_{st\text{-}LSH}$ is lower than those of the three competitive approaches, which indicates that our solution can return more accurate recommended list compared to other approaches. This is because in $Rec_{st\text{-}LSH}$ approach, both spatial and temporal information hidden in historical QoS data is taken into consideration to analyze users' potential preferences, which is beneficial for recommending "more accurate" candidate services to the target users.

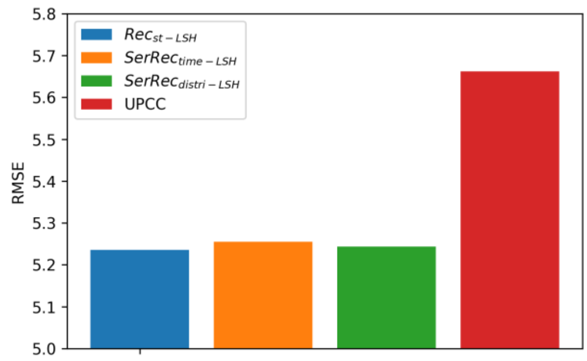

Fig.4 Accuracy comparisons (RMSE)

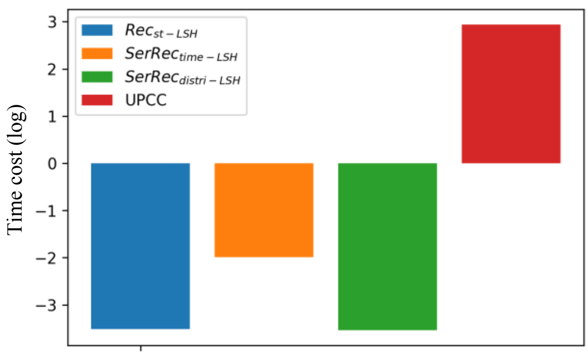

Fig.5 Efficiency comparison

Profile-2: Efficiency comparison of four approaches.

Recommendation efficiency is a key factor that influences users' willingness to use a recommender system. Next, the time costs of different approaches are evaluated. Parameter L = 10, r = 4. Test results are displayed in Fig.5. As shown in Fig.5, the computational cost of *UPCC* solution is much higher than the rest three ones as frequent and time-consuming user similarity calculation is necessary in *UPCC*. While in the three LSH-based solutions (i.e., $Rec_{st\text{-}LSH}$, $SerRec_{time\text{-}LSH}$ and $SerRec_{distri\text{-}LSH}$), user indices or service indices are built offline beforehand; as a result, the efficiency is improved significantly, which is rendered in Fig.5 (the log values of time costs are all negative). The comparison result means that our $Rec_{st\text{-}LSH}$ approach can response to the users' recommendation requests quickly.

Profile-3: Recall comparisons.

Recall is another criterion to evaluate the performance of a recommender system. In this profile, we also compare the recall value of $Rec_{st\text{-}LSH}$ and compare it with another two approaches, i.e., $SerRec_{distri\text{-}LSH}$ and *UPCC* (here, we do not test the recall value of $SerRec_{time\text{-}LSH}$ because this approach does not involve the step of "similar user" finding, instead, it focuses on the "similar time slot" finding). Here, we choose the Top-30 similar users of a target user and evaluate the recall of different approaches based on their respective returned 30 users. Parameters L = 10, r = 4. Test results are presented in Fig.6. We can see from Fig.6 that *UPCC* is the benchmark for recall measurement, therefore, its recall value is 100%. Our $Rec_{st\text{-}LSH}$ achieves 49.56% recall compared to 21.62% of $SerRec_{distri\text{-}LSH}$ as $Rec_{st\text{-}LSH}$ considers more useful context information than $SerRec_{distri\text{-}LSH}$.

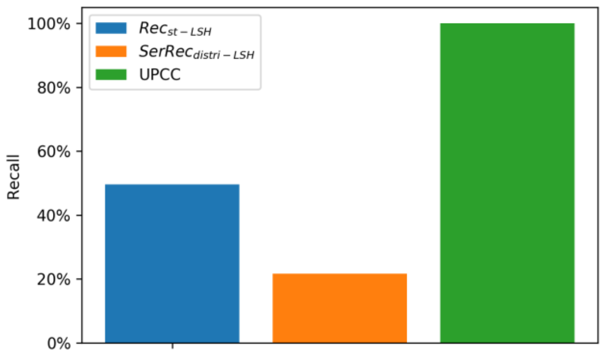

Fig.6 Recall comparison

Profile-4: Accuracy of $Rec_{st\text{-}LSH}$ with respect to L and r.

As two inherent parameters of LSH technique, L and r may affect the recommendation performance of $Rec_{st\text{-}LSH}$ approach. Inspired by this fact, we measure the correlation between the accuracy of $Rec_{st\text{-}LSH}$ and L-r pairs. Experiment parameter settings: r = 4, 8, 12, 16; L = 4, 6, 8, 10. Concrete test results are demonstrated in Fig.7.

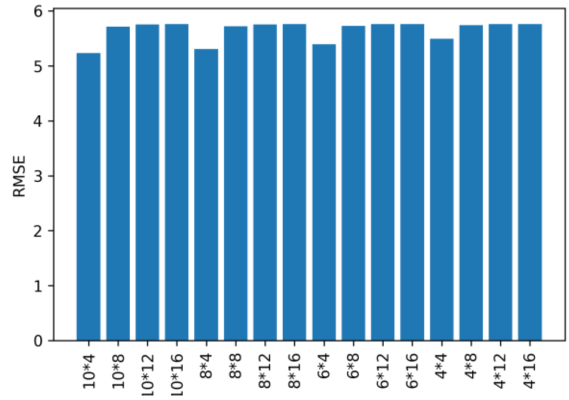

Fig.7 RMSE of $Rec_{st\text{-}LSH}$ w.r.t. (L, r)

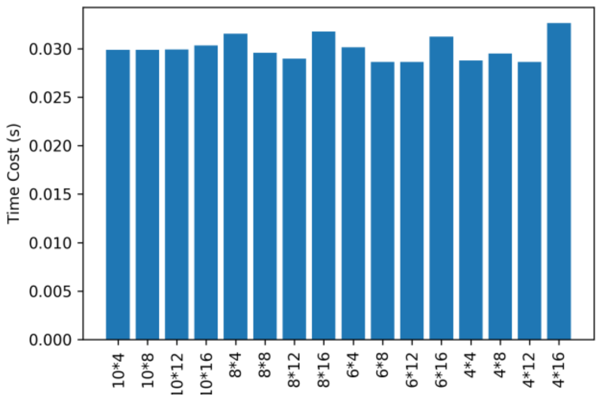

Fig.8 Time cost of $Rec_{st\text{-}LSH}$ w.r.t. (L, r)

The comparison results in Fig.7 indicate that the RMSE value increases when r (number of functions) grows. This is due to the fact that the dataset density is very low (1%). Therefore, when the similar user search condition becomes narrow (i.e., large r), it is hard to find the similar users of a target user, which also influences the recommendation accuracy.

Profile-5: Efficiency of $Rec_{st\text{-}LSH}$ with respect to L and r.

According to LSH, parameters L and r are two factors involved in the process of LSH-based recommendation solutions. Motivated by the above analyses, we investigate the correlation between the efficiency of $Rec_{st\text{-}LSH}$ and L-r pairs. Parameter settings: r = 4, 8, 12, 16; L = 4, 6, 8, 10. Fig.8 shows the concrete test results. Experiment data reveal that the efficiency of $Rec_{st\text{-}LSH}$ is not affected by the parameters L and r much as most of the computational tasks are finished offline; as a consequence, the needed time cost is often small enough.

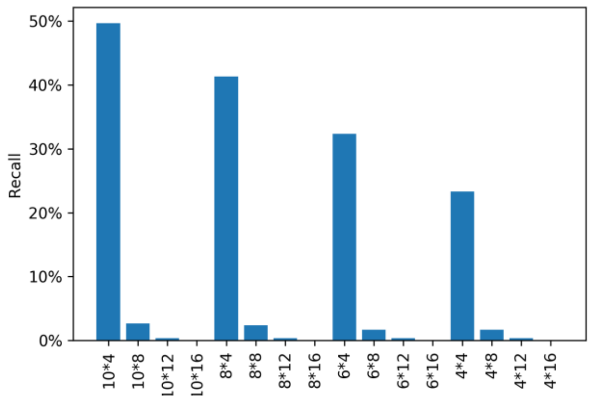

Fig.9 Recall of $Rec_{st\text{-}LSH}$ w.r.t. (L, r)

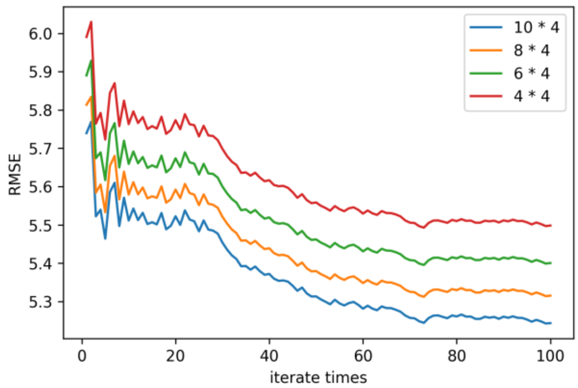

Fig.10 RMSE convergence of $Rec_{st\text{-}LSH}$ w.r.t. (L, r)

Profile-6: Recall of *Rec*$_{st\text{-}LSH}$ with respect to *L* and *r*.

Parameters L and r are two factors that may influence the recall of a recommender system. Inspired by this observation, we test the recall values under different parameter settings. Parameter settings: r = 4, 8, 12, 16; L = 4, 6, 8, 10. Fig.9 shows the concrete test results. Experiment data reveal that the recall value of $Rec_{st\text{-}LSH}$ declines with the growth of r and the decrease of L. This is due to the fact that a larger r or a smaller L often implies a more rigid search condition of similar users in the LSH theory; as a result, it is much harder to find similar users of a target user when r becomes larger or when L becomes smaller.

Profile-7: RMSE Convergence of *Rec*$_{st\text{-}LSH}$ with respect to *L* and *r*.

In the experiment setting, we claim that our experiments are repeated 100 times. Such a conclusion is drawn from the test results in Fig.10. We can see from the figure that the RMSE values of $Rec_{st\text{-}LSH}$ become convergent when the experiments are executed 100 times.

6 Conclusions

The ever-increasing popularity of web services as well as their produced big volume of QoS data call for various lightweight recommendation solutions including CF to reduce the service selection cost of users. While existing CF solutions mainly face the challenges of context-aware QoS fluctuation and privacy leakage. In view of these challenges, we integrate the spatial-temporal QoS data and the LSH technique into recommendation domain to pursue more accurate recommended results with privacy-preservation. At last, experiments conducted on well-known WS-DREAM dataset prove the advantages of our approach.

Service-oriented systems are often environment-aware [24-27] and hence probably generate multi-dimensional quality data [28-33]. Therefore, we will continue to refine the proposed recommendation model by integrating additional context information besides time and location. In addition, the location partition manner in this paper is a bit coarse. In the future, we will continue to investigate more fine-grained user location measurement manner.

References

1. Y. Zhang, Z. Zheng, M. R. Lyu, WSPred: a time-aware personalized qos prediction framework for web services. *IEEE International Symposium on Software Reliability Engineering*, pp. 210-219, IEEE, New York (2011).
2. K. Huang, Y. Fan, and W. Tan. Recommendation in an evolving service ecosystem based on network prediction. *IEEE Transactions on Automation Science and Engineering*, 11(3): 906-920, 2014.
3. Y. Zhong, Y. Fan, K. Huang, W. Tan, J. Zhang. Time-aware service recommendation for mashup creation. *IEEE Transactions on Services Computing*, 8(3): 356-368, 2015.
4. Y. Hu, Q. Peng, X. Hu, R. Yang. Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering. *IEEE Transactions on Services Computing*, 8(5): 782-794, 2015.
5. L. Qi, R. Wang, S. Li, Q. He, X. Xu, C. Hu. Time-aware Distributed Service Recommendation with Privacy-preservation. *Information Sciences*, 480: 354-364, 2019.
6. L. Qi, X. Xu, W. Dou, J. Yu, et al. Time-aware IoE service recommendation on sparse data, *Mobile Information Systems*, Article ID 4397061, 12 pages, 2016.
7. L. Yao, Q.Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, Q. He. Context-aware Point-of-Interest recommendation using tensor factorization with social regularization. *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1007-1010, 2015.
8. M. Tang, Y. Jiang, J. Liu, X. Liu. Location-Aware Collaborative Filtering for QoS-Based Service Recommendation. *IEEE International Conference on Web Services*, pp. 202-209, IEEE, New York (2012).
9. X. Fan, Y. Hu, R. Zhang, W. Chen, P. Brézillon. Modeling temporal effectiveness for context-aware web services recommendation. *IEEE International Conference on Web Services*, pp. 225-232, IEEE, New York (2015).
10. W. He, L. Cui, G. Ren, Q. Li, T. Li. A new paradigm for personalized mashup recommendation based on dynamic contexts in mobile computing environments. *SCIENTIA SINICA Informationis*, 46(6): 677-697, 2016.

11. X. Wang, J. Zhu, Z. Zheng, W. Song, Y. Shen, M. R. Lyu. A spatial-temporal qos prediction approach for time-aware web service recommendation. *ACM Transactions on the Web*, 10(1), Article 7, 25 pages, 2016.
12. C. Yu, L. Huang. A web service qos prediction approach based on time- and location-aware collaborative filtering. *Service Oriented Computing and Applications*, 10: 135-149, 2016.
13. S. Wang, Y. Ma, B. Cheng, F. Yang, R. N. Chang. Multi-dimensional qos prediction for service recommendations. *IEEE Transactions on Services Computing*, 2016. DOI: 10.1109/TSC.2016.2584058.
14. Q. Lin, R. Wang, S. Li, Q. He, X. Xu, C. Hu. Time-aware distributed service recommendation with privacy-preservation. *Information Sciences*, vol. 480, pp. 454-464, 2019.
15. J. Zhu, P. He, Z. Zheng, M. R. Lyu. A privacy-preserving qos prediction framework for web service recommendation. *IEEE International Conference on Web Services*, pp. 241-248, IEEE, New York (2015).
16. D. Li, C. Chen, Q. Lv, L. Shang, et al. An algorithm for efficient privacy-preserving item-based collaborative filtering. *Future Generation Computer Systems*, 55: 311-320, 2016.
17. P. Zhou, Y. Zhou, D. Wu, H. Jin. Differentially private online learning for cloud-based video recommendation with multimedia big data in social networks. *IEEE Transactions on Multimedia*, 18(6): 1217-1229, 2016.
18. L. Qi, X. Zhang, W. Dou, Q. Ni. A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. *IEEE Journal on Selected Areas in Communications*, 35(11): 2616-2624, 2017.
19. W. Gong, L. Qi, Y. Xu. Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. *Wireless Communications and Mobile Computing*, 2018: 1-8, Article ID 3075849, 2018.
20. C. Yan, X. Cui, L. Qi, X. Xu, X. Zhang. Privacy-aware data publishing and integration for collaborative service recommendation. *IEEE ACCESS*, 6: 43021-43028, 2018.
21. Y. Xu, L. Qi, W. Dou, J. Yu. Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment. *Complexity*, 2017: 1-9, Article ID 3437854, 2017.
22. L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, J. Chen. A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. *Future Generation Computer Systems*, 88: 636-643, 2018.
23. Y. Xia, S. Qu, S. Wan. Scene guided colorization using neural networks. *Neural Computing and Applications*, pp. 1-14, 2018.
24. J. S. Breese, D. Heckerman and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” *The Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, USA, pp. 43-52, 1998.
25. Z. Gao, D. Wang, S. Wan, H. Zhang, Y. Wang. Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval. *Future Generation Computer Systems*, 94: 641-653, 2019.
26. S. Ding, S. Qu, Y. Xi, S. Wan. A long video caption generation algorithm for big video data retrieval. *Future Generation Computer Systems*, 93: 583-595, 2019.
27. Y. Wang, Z. Cai, Z. Zhan, Y. Gong, X. Tong. An Optimization and Auction-Based Incentive Mechanism to Maximize Social Welfare for Mobile Crowdsourcing. *IEEE Transactions on Computational Social Systems*, 6(3): 414-429, 2019.
28. C. Luo, T. Li, Y. Huang, H. Fujita. Updating three-way decisions in incomplete multi-scale information systems, *Information Sciences*, 476: 274-289, 2019.

29. S. Wang, T. Li, C. Luo, H. Chen, H. Fujita. Domain-wise approaches for updating approximations with multi-dimensional variation of ordered information systems. *Information Sciences*, 478: 100-124, 2019.
30. Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong. An Incentive Mechanism with Privacy Protection in Mobile Crowdsourcing Systems. *Computer Networks*, 102: 157-171, 2016.
31. Z. Yu, T. Li, N. Yu, Y. Pan, H. Chen, B. Liu. Reconstruction of hidden representation for robust feature extraction. *ACM Transactions on Intelligent Systems and Technology*, 10 (2): 18, 2019.
32. J. Liu, R. M. Rodríguez, L. Martínez. New trends of information fusion in decision making. *Information Fusion*, 29 (C): 87-88, 2016.
33. B. Zhao, Y. Wang, Y. Li, Y. Gao, X. Tong. Task Allocation Model Based on Worker Friend Relationship for Mobile Crowdsourcing. *Sensors*, 19(4): 921, 2019.