IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2018, accepted July 30, 2018, date of publication August 3, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863050

Privacy-Aware Data Publishing and Integration for Collaborative Service Recommendation

CHAO YAN[1], XINCHUN CUI[1], LIANYONG QI[1,2], XIAOLONG XU[3], AND XUYUN ZHANG[4]

[1]School of Information Science and Engineering, Qufu Normal University, Rizhao, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210008 , China
[3]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[4]Department of Electrical and Computer Engineering, The University of Auckland, Auckland 1010, New Zealand

Corresponding author: Lianyong Qi (lianyongqi@gmail.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61702277 and Grant 61702299, in part by the Shandong Province Social Science Planning Research Project under Grant 16CTQJ02, in part by the New Zealand Marsden Funds, and in part by the UoA Faculty Research Development Fund under Grant 3714668.

ABSTRACT Item-based collaborative filtering (i.e., ICF) technique has been widely recruited to make service recommendations in the big data environment. However, the ICF technique only performs well when the data for service recommendation decision-making are stored in a physically centralized manner, while they often fail to recommend appropriate services to a target user in the distributed environment where the involved multiple parties are reluctant to release their data to each other due to privacy concerns. Considering this drawback, we improve the traditional ICF approach by integrating the locality-sensitive hashing (LSH) technique, to realize secure and reliable data publishing. Furthermore, through integrating the published data with little privacy across different platforms, appropriate services are recommended based on our suggested recommendation approach named ICF_{LSH}. At last, simulated experiments are conducted on WS-DREAM data set. Experiment results show that ICF_{LSH} performs better than the competitive approaches in terms of service recommendation accuracy, efficiency, and the capability of privacy-preservation.

INDEX TERMS Item-based collaborative filtering, data publishing and integration, service recommendation, privacy-preservation, locality-sensitive hashing.

I. INTRODUCTION

Motivated by the advantages of IoT (Internet of Things) [1]–[3], an increasing number of enterprises and organizations have begun to encapsulate their developed products into various easy-to-access web services (e.g., foods, movies, places of interest) [4], [5]. Thus remote users can browse, compare and select their preferred services from massive candidates. While a challenge is raised when a target user has massive candidate services because the target user often has little knowledge of the functional or non-functional properties of candidate services. Under this circumstance, the collaborative recommendation techniques, e.g., the well-known item-based collaborative filtering (i.e., ICF) [6], are recently introduced to help the target user to find out appropriate services that suit the personalized preferences of the target user.

Existing ICF technique is often applicable to the service recommendation scenarios where the historical service quality data used to make recommendation decisions are stored in a centralized manner [7]. However, they often fall short in recommending target users their preferred services when the data for recommendations are distributed across multiple platforms. This is due to the following two reasons. First, in the distributed environment, a platform dares not release its stored private data to the public [8], which blocks the data publishing and cross-platform data sharing. Second, the cross-platform data integration often requires additional communication cost [9], especially when the data recorded in each platform are updated or increased frequently. In other words, the cross-platform recommendation efficiency is often not high enough to satisfy the urgent service recommendation requests from partial users.

Considering the abovementioned challenges, the ICF technique is boosted by integrating Locality-Sensitive Hashing (LSH), an efficient and privacy-preserving similar neighbor search technique; furthermore, a recommendation method named ICF_{LSH} is suggested based on ICF and LSH,. Benefiting from the inherent advantages of LSH in terms of privacy-preservation and high efficiency, ICF_{LSH} can help a recommender system to find out the optimal service recommendation solution, in a privacy-preserving and efficient manner.

Concretely, our contributions are two-fold.

(1) The traditional ICF technique is improved by integrating the LSH technique so as to encourage the privacy-preserving data publishing and integration in the distributed environment. Furthermore, according to ICF and LSH, an approach named ICF_{LSH} is suggested to cope with the service recommendation problems with privacy concerns in the multi-source environment.

(2) To prove the advantages of the suggested ICF_{LSH} approach, simulated experiments are designed based on the well-known dataset, i.e., *WS-DREAM*. The positive results from experiments demonstrate that the ICF_{LSH} approach realizes a better tradeoff or balance among recommendation accuracy and scalability while protecting the private information of users, compared to other competitive approaches.

The structure of paper is organized as follows. Related work is presented in Section 2. An intuitive example is introduced in Section 3 for motivation specification. In Section 4, we formulate the ICF recommendation problem in the distributed environment. In Section 5, we first introduce the LSH technique briefly; afterwards, we specify the details of suggested ICF_{LSH} approach. Experiments are performed in Section 6 and conclusions are drawn in Section 7.

II. RELATED WORK

Next, we introduce the current research status of Collaborative Filtering-based service recommendation from the following two aspects: memory-based Collaborative Filtering and model-based Collaborative Filtering.

A. MEMORY-BASED COLLABORATIVE FILTERING

User-based Collaborative Filtering (i.e., UCF) and item-based Collaborative Filtering (i.e., ICF) are put forward for high-quality recommended results in [6] and [10], respectively. To merge their advantages, Jiang *et al.* [11] suggest a hybrid CF-based service recommendation approach named *WSRec*. However, the service quality is often not fixed, but dynamic. Concretely, service quality is often influenced by the network context information such as time and location; therefore, to make accurate service recommendations, the time factor and location factor are taken into considerations in [12] and [13]. Besides, the target user's service selection decisions are often related to the subjective user preferences. Therefore, preference information is added in [4] to accommodate the recommendation requests of personalized users. While existing approaches suffer from the low efficiency and scalability when the decision-making data for recommendations update frequently. Beside, existing approaches fit the centralized recommendation scenarios more, while fail to tackle the distributed recommendation issues [14].

B. MODEL-BASED COLLABORATIVE FILTERING

Different from the memory-based CF, the model-based CF often first builds a service quality prediction and service recommendation model offline based on historical decision data generated from past invocations. Afterwards, according to the model, a recommender system makes appropriate service recommendations. Currently, the popular model-based recommendation techniques include Matrix Factorization (MF) [15], Latent Dirichlet Allocation (LDA) [16], tensors [17], [18], and so on. These model-based CF approaches are often time-efficient due to the offline model training process. While similar to memory-based CF introduced in subsection 2.A, existing model-based CF approaches seldom consider the privacy concerns raised in the distributed environment [19].

Through the above analyses, a conclusion can be drawn that existing CF-based recommendation approaches often fall short in protecting the private information of users during the cross-platform service recommendation process. Considering this drawback, we improve the traditional ICF technique through integrating the LSH technique and further bring forth a novel service recommendation approach, i.e., ICF_{LSH}. The details of ICF_{LSH} will be specified in the following sections.

III. MOTIVATION

In Fig.1, an example is presented to illustrate the ICF recommendation in the distributed environment. Concretely, web service ws_1 was invoked by users $\{u_{1,1}, u_{1,2}, u_{2,1}\}$ in Amazon, web service ws_2 was invoked by users $\{u_{1,2}, u_{2,1}, u_{2,2}\}$ in IBM. Then according to ICF, the first step is to calculate the item similarity $sim(ws_1, ws_2)$ between services ws_1 and ws_2. However, the calculation of $sim(ws_1, ws_2)$ faces the following two challenges:

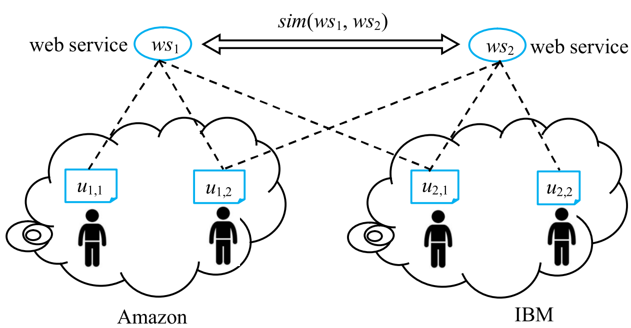


FIGURE 1. Distributed item-based CF recommendation: an example.

(1) To protect sensitive user information, Amazon often dares not release its data (i.e., $u_{1,1}$'s and $u_{1,2}$'s observed quality data of service ws_1) to IBM, vice versa, which renders the calculation of $sim(ws_1, ws_2)$ infeasible.

(2) The volume of data in both Amazon and IBM may update frequently with time elapsing, which requires frequent calculation of $sim(ws_1, ws_2)$. As a result, the time cost is often high and hence fails to meet the emergent requests from target users.

To tackle the above issues, an improved ICF recommendation approach, i.e., ICF_{LSH} is suggested for privacy-aware and scalable recommendation problems.

IV. PROBLEM FORMALIZATION

Next, we formalize the ICF recommendation problem in the distributed environment as a five-tuple $ICF(ws_{target}, P, U, WS, q)$, where

(1) ws_{target} denotes the target web service, i.e., one of the services invoked by the target user.

(2)$P = \{p_1, \ldots, p_z\}$: $p_k(1 \leq k \leq z)$ denotes k-th distributed platform which provides k-th part of ws_{target}'s historical quality data. For example, z is equal to 2 in Fig.1.

(3) $U = \{U_1, \ldots, U_z\}$: U_k $(1 \leq k \leq z)$ denotes users in platform p_k. To ease the following discussion, we assume that each platform has m users, i.e., $U_k = \{u_{k,1}, \ldots, u_{k,m}\}$ where $u_{k-i}(1 \leq i \leq m)$ denotes i-th user on platform p_k.

(4) $WS = \{ws_1, \ldots, ws_n\}$: $ws_i(1 \leq i \leq n)$ denotes the i-th web service. For each element in set WS, its historical quality data are distributed, i.e., multi-sourced.

(5) q denotes a service quality criterion such as *throughput*. $ws.q$ denotes the quality of q of service ws; here, it is supposed that the collected decision-making data are all trustworthy [20] and real values [21]–[25].

V. A SERVICE RECOMMENDATION APPROACH BASED ON ICF AND LSH: ICF_{LSH}

Concrete process of our suggested privacy-preserving recommendation approach, i.e., ICF_{LSH} is introduced in this section. Concretely, LSH technique is first introduced in subsection 5.A; afterwards, the ICF_{LSH} approach is specified step by step in subsection 5.B.

A. LOCALITY-SENSITIVE HASHING

As an effective similar neighbor search technique [26], LSH can guarantee (a) two neighbors x and y will be mapped into the same bucket after hash operations; (b) two non-neighbors x and y will be mapped into different buckets after hash operations.

Any function (or a family of hash functions) that satisfies the above two conditions are called a LSH function. Concretely, $h(.)$ can be regarded as a qualified LSH function if and only if the inequations in (1)-(2) hold simultaneously. Here, $d(x, y)$ represents the distance between points x and y; $h(x)$ denotes the hash value (or index) of x; $P(X)$ denotes the probability (assume positive [27], [27]–[31]) that event X holds, $\{d_1, d_2, p_1, p_2\}$ are a set of thresholds. If the conditions in (1) and (2) hold simultaneously, then we can conclude that $h(.)$ is a LSH function and $h(.)$ is (d_1, d_2, p_1, p_2)-sensitive.

$$\text{If } d(x, y) \leq d_1, \quad \text{then } P(h(x) = h(y)) \geq p_1 \tag{1}$$
$$\text{If } d(x, y) \geq d_2, \quad \text{then } P(h(x) = h(y)) \leq p_2 \tag{2}$$

B. ICF_{LSH}: LSH-BASED ICF RECOMMENDATION APPROACH

Concretely, as Fig.2 shows, the suggested ICF_{LSH} approach consists of three steps.Note that ws_{target} represents a service invoked by the target user. Other symbols to be used in this paper as well as their meanings can be found in Table 1.

Step-1: Building service sub-indices offline. For web service ws, according to ws's historical quality data in platform p_k $(1 \leq k \leq z)$, build ws's sub-index $H_k(ws)$ based on a family of hash functions $h_{k,1}(.)$, …, $h_{k,r}(.)$.

Step-2: Generating service indices by merging service sub-indices offline. For service ws, merge its sub-indices $H_1(ws)$, …, $H_z(ws)$ derived in Step 1 to be an index $H(ws)$ offline.

Step-3: Service recommendation. For target service ws_{target}, find its similar services online based on its index $H(ws_{target})$ and the indices of other services. Then the optimal similar services are recommended to the target user.

FIGURE 2. Three steps of the ICF_{LSH} approach.

TABLE 1. Symbol specifications.

symbols	specifications
p_1, …, p_z	distributed platforms
$u_{k,1}$, …, $u_{k,m}$	users in platform p_k (we assume that the number of users in different platforms are the same)
ws_1, …, ws_n	candidate web services
$h(ws)$	hash value of service ws
$H_k(ws)$	service ws's sub-index corresponding to platform p_k
$H(ws)$	service ws's index
b_1, …, b_t	buckets in a hash table
$Table_1$, …, $Table_L$	L hash tables
$h_1(.)$, …, $h_r(.)$	in each hash table, we assume that each platform is corresponding to r hash functions

Step-1: Building user sub-indices offline.

Lets consider a web service ws and a platform p_k which has m users $u_{k,1}, \ldots, u_{k,m}$. Then ws's historical quality data (over quality dimension q) in platform p_k can be specified by the vector $\overrightarrow{ws(k)} = (ws.q_{k,1}, \ldots, ws.q_{k,m})$ (here, we assume that the service quality data are fixed values [32], [36]). Next, according to the LSH theory, we calculate the hash value of $\overrightarrow{ws(k)}$, i.e., $h(\overrightarrow{ws(k)})$ through the equation in (3) [37]. Please note that $\overrightarrow{v}$ is a vector $(v_1, \ldots, v_m)$ where each dimensional value $v_j(1 \leq j \leq m)$ is randomly selected from the domain $[-1, 1]$; "$a \circ b$" represents the dot product between vectors a and b.

$$h(\overrightarrow{ws(k)}) = \begin{cases} 1 & \text{if } \overrightarrow{ws(k)} \circ \overrightarrow{v} > 0 \\ 0 & \text{if } \overrightarrow{ws(k)} \circ \overrightarrow{v} \leq 0 \end{cases} \tag{3}$$

As LSH is a probability-based similar neighbor search technique, one hash function $h(.)$ is often not enough. So we select a hash function family $\{h_{k,1}(.), \ldots, h_{k,r}(.)\}$ and then derive $H_k(ws) = (h_{k,1}(\overrightarrow{ws(k)}), \ldots, h_{k,r}(\overrightarrow{ws(k)}))$. Thus $H_k(ws)$ can be regarded as ws's sub-index corresponding to platform p_k. As $\overrightarrow{ws(k)}$ is stored in p_k, sub-index $H_k(ws)$ can be built offline by p_k; this way, the time cost can be reduced significantly.

Step-2: Generating service indices by merging service sub-indices offline.

In Step 1, we have obtained the web service ws's sub-index $H_k(ws)$ corresponding to platform $p_k (1 \leq k \leq z)$. Next, we merge the z sub-indices into a complete index for service ws offline, i.e., $H(ws) = (H_1(ws), \ldots, H_z(ws))$. For each candidate web service $ws_i (1 \leq i \leq n)$, we can obtain its complete index $H(ws_i)$ through Step 1 and Step 2. With the mapping relationships between services and their complete indices (in the form of vector [38], [39]), a hash table is formed offline.

Step-3: Service recommendation.

According to the LSH theory, if two services have the same hash values, then they are similar with large probability. Therefore, the services which are similar to ws_{target} can be found based on their respective complete indices. Concretely, service ws_x is similar to ws_{target} iff the condition in (4) holds.

$$H(ws_x) = H(ws_{target}) \tag{4}$$

However, if condition in (4) does not hold, we cannot conclude that ws_x is not similar to ws_{target} because LSH is a probability-based neighbor search technique. In view of this, we create L hash tables instead of only one hash table, i.e., $Table_1, \ldots, Table_L$, and then modify condition (4) to be condition (5). Namely, service ws_x is similar to ws_{target} iff condition (5) holds.

$$\exists Table_T (T \in \{1, \ldots, L\}), \quad \text{s.t. } H(ws_x) = H(ws_{target}) \tag{5}$$

If a similar service of ws_{target}, denoted by ws_x, has never been invoked by the target user, then we put it into set *Sim_set*. Specifically, if the target user has multiple target services, we repeat (3)-(5) to find all the qualified similar services and put them into *Sim_set* (if a service emerges multiple times in *Sim_set*, then the average service quality is adopted). Next, we select the optimal service from *Sim_set*. Concretely, if condition (6) holds, then ws_x is the optimal and recommended to the target user. Here, q denotes a quality criterion of web services. More formally, our proposed ICF_{LSH} approach is introduced in detail by the pseudo code.

$$ws_x.q = \text{optimal}(ws_i.q \mid ws_i \in Sim\_set) \tag{6}$$

Algorithm 1 ICF_{LSH}

Inputs: $P = \{p_1, \ldots, p_z\}$: distributed platforms
$U_k = \{u_{k-1}, \ldots, u_{k-m}\}$: user set in platform p_k
$WS = \{ws_1, \ldots, ws_n\}$: candidate web services
q: a quality dimension of web services
ws_{target}: target services
Output: ws_x: optimal service recommended to the target user

/* **Step 1: Building service sub-indices offline** */
for $k = 1$ to z **do**
  for $g = 1$ to r **do**
    for $j = 1$ to m **do**
      $h_{kgj} = random[-1, 1]$
    end for
    $\overrightarrow{v_g} = (h_{kg1}, \ldots, h_{kgm})$
    for $i = 1$ to n **do**
      $\overrightarrow{ws_{i(k)}} = (ws_i.q_{k,1}, \ldots, ws_i.q_{k,m})$
      if $\overrightarrow{ws_{i(k)}} \circ \overrightarrow{v_g} > 0$ // dot product
        then $h_{k,g}(\overrightarrow{ws_{i(k)}}) = 1$
        else $h_{k,g}(\overrightarrow{ws_{i(k)}}) = 0$
      end if
    end for
  end for
  for $i = 1$ to n **do**
    $H_k(ws_i) = (h_{k,1}(\overrightarrow{ws_{i(k)}}), \ldots, h_{k,r}(\overrightarrow{ws_{i(k)}}))$
  end for
end for
/* **Step 2: Generating service indices by merging service sub-indices offline** */
for $i = 1$ to n **do**
  $H(ws_i) = (H_1(ws_i), \ldots, H_z(ws_i))$
end for
create hash table *Table* based on $H(ws_1), \ldots, H(ws_n)$
/* **Step 3 : Service recommendation** */
Sim_Set= Φ// similar service set of ws_{target}
for $T = 1$ to L **do**
  repeat Step-1 and Step-2 to create $Table_T$
  for each ws_{target} **do**
    find bucket b corresponding to $H(ws_{target})_T$
    if $b \neq Null$
    then put services in b into *Sim_Set*
    end if
  end for
end for
for $x = 1$ to $|$ *Sim_Set* $|$ **do**
  if ws_x is put into *Sim_Set* multiple times
  then ws_x's average service quality is adopted
  end if
end for
find candidate service ws_x with optimal service quality
return ws_x to the target user

VI. EXPERIMENT

A wide range of experiments are provided in this section to test the effectiveness and efficiency of the suggested ICF_{LSH} method in this paper, while protecting the private information of users in the distributed service recommendation scenarios.

A. EXPERIMENT CONFIGURATION

Experiments are based on *WS-DREAM* [40], i.e., a dataset that reports the quality values of 5825 real-world services observed by 339 users around the world. The 5825 web services are located in different countries, so we consider each country as an independent platform. The service quality matrix is very dense. So we randomly remove partial entries from the matrix and then predict the missing quality values, through which we can calculate the accuracy of the final recommended results. Moreover, only one quality dimension, i.e., *response time*, is recruited here.

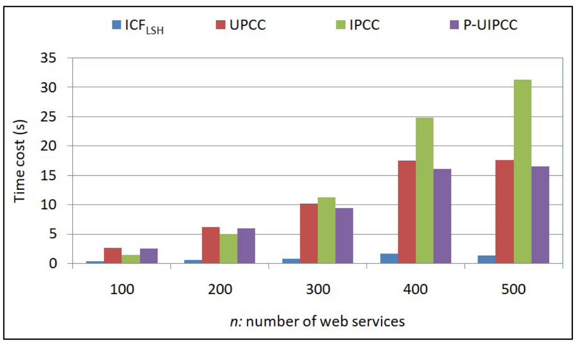
Concretely, we test the *time cost* and *MAE* to measure the service recommendation performances. Moreover, to validate the advantages of our ICF_{LSH} approach, we compare ICF_{LSH} with another three recommendation approaches: *UPCC* [10], *IPCC* [41] and *P-UIPCC* [42]. Here, *UPCC* and *IPCC* are two benchmark service recommendation approaches, while *P-UIPCC* approach utilizes the data obfuscation technique to protect the sensitive service quality data of users.

The experiments were conducted on a PC with Intel Core i5 3.2GHz processors and 16.0 GB RAM. The machine is running under Windows 10 (64 bits) and Python 3.6.1. Each test was run 100 times and their average values were adopted at last.

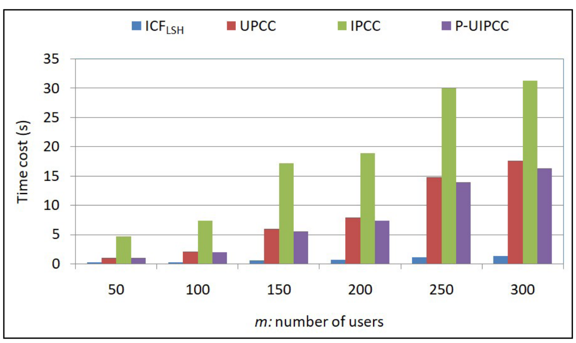
B. RESULTS

The following three profiles are tested and compared in our experiments. Here, m and n denote the number of users and number of services, respectively; L and r denote the number of LSH tables and the number of LSH functions, respectively.

Profile 1 (Time Cost of Four Approaches): The time cost values of different recommendation approaches are tested, respectively. Parameter values are set as below: $m = 50, 100, 150, 200, 250, 300$; $n = 100, 200, 300, 400, 500$; $T = 10$; $r = 10$. Concrete experiment data are presented in Fig.3.



(a)

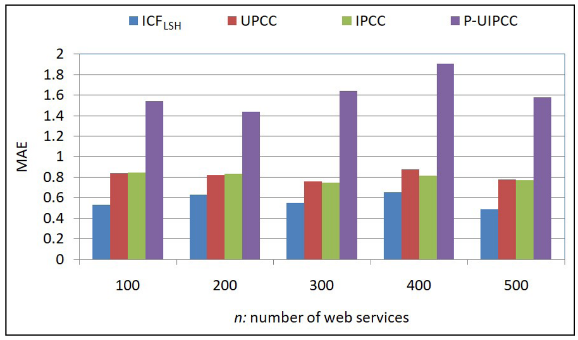


(b)

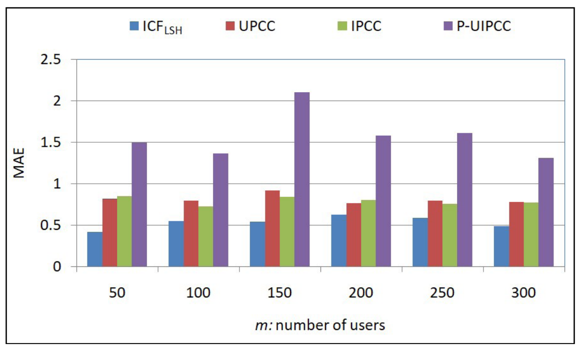
FIGURE 3. **Time cost of four competitive approaches. (a) $m = 300$. (b) $n = 500$.**

As Fig.3 indicates, the efficiency of suggested ICF_{LSH} approach is relatively high and performs better than those of *UPCC*, *IPCC* and *P-UIPCC* approaches. This is because in our proposal, most jobs (e.g., service indices building) is done offline, while the remaining job (e.g., online similar service search) can be finished quickly as its time complexity is near O(1). The low time cost means that our ICF_{LSH} approach can be applied on the service recommendation situations when a quick response is required by the users.

Profile 2 (Accuracy of Four Approaches): Note that the accuracy of the final recommended results is measured with *MAE* (the smaller the better). In this profile, we measure and compare the accuracy values of four competitive approaches. Parameter values are set as below: $m = 50, 100, 150, 200, 250, 300$; $n = 100, 200, 300, 400, 500$; $T = 10$; $r = 10$. Concrete experiment data are presented in Fig.4.



(a)



(b)

FIGURE 4. **Recommendation accuracy of three approaches. (a) $m = 300$. (b) $n = 500$.**

As can be seen from Fig.4, the accuracy value of *P-UIPCC* approach is the lowest (i.e., *MAE* value is the largest), because the recommendation data are already obfuscated. While our proposed ICF_{LSH} approach outperforms those of the other three approaches. This is because in our proposal, only the "most similar" services are returned and recruited to make service recommendation; therefore, the accuracy is increased. Moreover, the recommendation accuracy of ICF_{LSH} does not vary sharply with the growth of m or n, which is another observation from Fig.4.

Profile 3 (Recommendation Accuracy of ICF_{LSH} w.r.t. L and r): According to the LSH theory, the recommendation

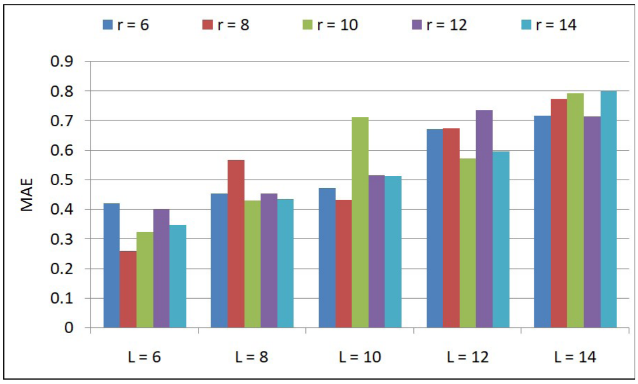


FIGURE 5. Accuracy value of ICF_{LSH} w.r.t. L and r.

accuracy of our proposed ICF_{LSH} approach is correlated with L and r. Next, we investigate the potential correlations between the accuracy value of our suggested ICF_{LSH} approach and the parameters L and r. The experiment parameters are set as below: $m = 300$; $n = 500$; $L = 6, 8, 10, 12, 14$; $r = 6, 8, 10, 12, 14$. Concrete experiment data are presented in Fig.5. From the figure we can see that the accuracy of ICF_{LSH} decreases when L grows; the reason is: when L grows, the search condition for neighbors would become looser and hence lead to lower recommendation accuracy.

C. TIME COMPLEXITY ANALYSES

In this subsection, we analyse the time complexity of ICF_{LSH} approach. Suppose that m and n represent the number of users and the number of services, respectively; L and r represent the number of LSH tables and the number of LSH functions, respectively.

(1) Step-1:**Building service sub-indices offline.** In this step, we need to build the index $H(ws)$ offline for each service ws, whose time complexity is O(1). Then we calculate the target service's index based on the r hash functions in each of the L hash tables, whose time complexity is $O(m * r * L)$. Afterwards, we search for the target service's neighbours online based on the service indices in the L hash tables, whose time complexity is $O(L)$. Therefore, the complexity of Step-1 is $O(m * r * L)$.

(2) Step-2:**Generating service indices by merging service sub-indices offline.**

In this step, for each web service ws, we need to merge its z sub-indices into a complete index offline. Therefore, the time complexity of Step-2 is O(1).

(3) Step-3:**Service recommendation.**

In this step, we first search for the similar services of a target service based on the L hash tables, whose time complexity is $O(L)$. Afterwards, according to the obtained similar services (at most $n - 1$ services), we predict the missing quality of a service, whose time complexity is $O(n)$. As there are at most $n - 1$ target services, the complexity of Step-3 is $O(n * (n + L)) = O(n^2)(L \ll n)$.

Through the above analyses, a conclusion can be drawn that the time complexity of our proposal is $O(n^2 + m * r * L)$.

D. FURTHER DISCUSSIONS

In our proposal, only one service quality criterion (e.g., *response time*) is considered; while in the actual situations, multiple dimensions are more common [43]–[46]. So in the future, we plan to extend our work by introducing multiple dimensions as well as the weight information which is often recruited to indicate the dimension importance [29], [47], [48]. Besides, we only consider the quality dimensions whose values are real and continuous, e.g., *response time*, while neglecting some other dimensions whose values are discrete [49], [50] binary [51], fuzzy [52] and correlated [53]–[55]. So in the future, we will investigate how to integrate the quality dimensions with different data types.

VII. CONCLUSION

In this paper, item-based Collaborative Filtering (ICF) technique is improved by integrating the LSH technique, to realize the secure data publishing and efficient data integration in the distributed service recommendation environment. Furthermore, we suggest an approach named ICF_{LSH} to help a recommender system to make privacy-preserving service recommendation decisions. To test the effectiveness and efficiency of our proposed ICF_{LSH} approach, a set of experiments are designed and deployed on the well-known real-world service quality dataset, i.e., *WS-DREAM*. Experiment results reveal that our proposal performs better than the other competitive recommendation approaches in terms of time cost and accuracy while protecting the private information of users in the distributed environment. In the future work, we will continue to refine ICF_{LSH} by taking into consideration the data types and weight of different quality dimensions. Besides, multiple forms of quality data will also be investigated.

REFERENCES

[1] S. Li, L. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[2] T. Wang, M. Z. A. Bhuiyan, G. Wang, A. Rahman, J. Wu, and J. Cao, "Big data reduction for a smart city's critical infrastructural health monitoring," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 128–133, Mar. 2018.

[3] X. Wang, L. T. Wang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[4] K. K. Fletcher and X. F. Liu, "A collaborative filtering method for personalized preference-based service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Jun./Jul. 2015, pp. 400–407.

[5] S. Li, T. Tryfonas, G. Russell, and P. Andriotis, "Risk assessment for mobile systems through a multilayered hierarchical Bayesian network," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1749–1759, Aug. 2016.

[6] K.-Y. Chung, D. Lee, and K. J. Kim, "Categorization for grouping associative items using data mining in item-based collaborative filtering," *Multimedia Tools Appl.*, vol. 71, no. 2, pp. 889–904, 2014.

[7] L. Qi *et al.*, "Structural balance theory-based E-commerce recommendation over big rating data," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2016.2602849.

[8] T. Wang *et al.*, "Fog-based storage technology to fight with cyber threat," *Future Gener. Comput. Syst.*, vol. 83, pp. 208–218, Jun. 2018.

[9] X. Wang, L. T. Yang, H. Liu, and M. J. Deen, "A big data-as-a-service framework: State-of-the-art and perspectives," *IEEE Trans. Big Data*, to be published, doi: .10.1109/TBDATA.2017.2757942.

[10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.

[11] C. Jiang, R. Duan, H. K. Jain, S. Liu, and K. Liang, "Hybrid collaborative filtering for high-involvement products: A solution to opinion sparsity and dynamics," *Decis. Support Syst.*, vol. 79, pp. 195–208, Nov. 2015.

[12] L. Qi, X. Xu, W. Dou, J. Yu, Z. Zhou, and X. Zhang, "Time-aware IoE service recommendation on sparse data," *Mobile Inf. Syst.*, vol. 2016, Nov. 2016, Art. no. 4397061.

[13] G. Ference, M. Ye, and W.-C. Lee, "Location recommendation for out-of-town users in location-based social networks," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 721–726.

[14] T. Wang *et al.*, "Data collection from WSNs to the cloud based on mobile fog elements," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.07.031.

[15] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.

[16] F. Li, "Backward solutions to Neumann and Dirichlet problems of heat-conduction equation," *Appl. Math. Comput.*, vol. 210, pp. 211–214, Apr. 2009.

[17] X. Wang, L. T. Yang, J. Feng, X. Chen, and M. J. Deen, "A tensor-based big service framework for enhanced living environments," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 36–43, Nov./Dec. 2016.

[18] L. T. Yang, X. Wang, X. Chen, J. Han, and J. Feng, "A tensor computation and optimization model for cyber-physical-social big data," *IEEE Trans. Sustain. Comput.*, to be published.

[19] S. Li, J. Chen, T. Spyridopoulos, P. Andriotis, R. Ludwiniak, and G. Russell, "Real-time monitoring of privacy abuses and intrusion detection in android system," in *Proc. HCI*, vol. 22, 2015, pp. 379–390.

[20] T. Wang *et al.*, "A comprehensive trustworthy data collection approach in sensor-cloud system," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2018.2811501.

[21] C. Ma and J. Zhang, "On formability of linear continuous-time multi-agent systems," *J. Syst. Sci. Complex.*, vol. 25, no. 1, pp. 13–29, 2012.

[22] G.-L. Tian, M. Wang, and L. Song, "Variable selection in the high-dimensional continuous generalized linear model with current status data," *J. Appl. Statist.*, vol. 41, no. 3, pp. 467–483, 2014.

[23] H. Wu, Y. Ren, and F. Hu, "Continuous dependence property of BSDE with constraints," *Appl. Math. Lett.*, vol. 45, pp. 41–46, Jul. 2015.

[24] L. L. Liu, "Continued fractions and the derangement polynomials of types A and B," *Ars Combinatoria*, vol. 125, pp. 321–330, 2016.

[25] F. Hu, "The modulus of continuity theorem for G-Brownian motion," *Commun. Statist., Theory Methods*, vol. 46, no. 7, pp. 3586–3598, 2017.

[26] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *VLDB*, vol. 99, no. 6, pp. 518–529, 1999.

[27] H. Dong and C. Yin, "Complete monotonicity of the probability of ruin and de Finetti's dividend problem," *J. Syst. Sci. Complex.*, vol. 25, no. 1, pp. 178–185, 2012.

[28] K. C. Yuen and C. Yin, "Asymptotic results for tail probabilities of sums of dependent and heavy-tailed random variables," *Chin. Ann. Math., B*, vol. 33, no. 4, pp. 557–568, 2012.

[29] Y.-F. Wang, C.-C. Yin, and X.-S. Zhang, "Uniform estimate for the tail probabilities of randomly weighted sums," *Acta Math. Appl. Sinica, English Ser.*, vol. 30, no. 4, pp. 1063–1072, 2014.

[30] F. Hu, Z. Chen, and P. Wu, "A general strong law of large numbers for non-additive probabilities and its applications," *Statist., J. Theor. Appl. Statist.*, vol. 50, no. 4, pp. 733–749, 2016.

[31] Y. Wang and C. Yin, "Approximation for the ruin probabilities in a discrete time risk model with dependent risks," *Statist. Probab. Lett.*, vol. 80, pp. 1335–1342, Sep. 2010.

[32] X. Du and Z. Zhao, "On fixed point theorems of mixed monotone operators," *Fixed Point Theory Appl.*, vol. 2011, p. 563136, Dec. 2011.

[33] X. Li and Z. Zhao, "On a fixed point theorem of mixed monotone operators and applications," *Electron. J. Qualitative Theory Differ. Equ.*, vol. 2011, no. 94, pp. 1–7, 2012.

[34] D. Kong, L. Liu, and Y. Wu, "Best approximation and fixed-point theorems for discontinuous increasing maps in Banach lattices," *Fixed Point Theory Appl.*, vol. 2014, p. 18, Dec. 2014.

[35] X. Zhang, L. Liu, and Y. Wu, "Fixed point theorems for the sum of three classes of mixed monotone operators and applications," *Fixed Point Theory Appl.*, vol. 2016, p. 49, Dec. 2016.

[36] Z. Zhao, "Existence of fixed points for some convex operators and applications to multi-point boundary value problems," *Appl. Math. Comput.*, vol. 215, no. 8, pp. 2971–2977, 2009.

[37] Y. Ioannidis *et al.* (2015). *Data Mining and Query Log Analysis for Scalable Temporal and Continuous Query Answering*. [Online]. Available: http://www.optique-project.eu/

[38] Z. Gao and W. Wang, "Large and moderate deviations for a renewal randomly indexed branching process," *Statist. Probab. Lett.*, vol. 116, pp. 139–145, Sep. 2016.

[39] Z. Gao and Y. Zhang, "Limit theorems for a supercritical Poisson random indexed branching process," *J. Appl. Probab.*, vol. 53, no. 1, pp. 307–314, 2016.

[40] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world Web services," *IEEE Trans. Services Comput.*, vol. 7, no. 1, pp. 32–39, Jan./Mar. 2014.

[41] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186.

[42] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving QoS prediction framework for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Jun./Jul. 2015, pp. 241–248.

[43] M. Wang and G.-L. Tian, "Robust group non-convex estimations for high-dimensional partially linear models," *J. Nonparam. Statist.*, vol. 28, no. 1, pp. 49–67, 2016.

[44] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *J. Appl. Statist.*, vol. 43, no. 5, pp. 796–809, 2016.

[45] P. Wang and L. Zhao, "Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds," *Nonlinear Anal.*, vol. 130, pp. 1–17, Jan. 2016.

[46] S. Li, G. Oikonomou, T. Tryfonas, T. M. Chen, and L. D. Xu, "A distributed consensus algorithm for decision making in service-oriented Internet of things," *IEEE Trans Ind. Informat.*, vol. 10, no. 2, pp. 1461–1468, May 2014.

[47] S. Yang, Z.-A. Yao, and C.-A. Zhao, "The weight distributions of two classes of p-ary cyclic codes with few weights," *Finite Fields Appl.*, vol. 44, pp. 76–91, Mar. 2017.

[48] J. Cai, "An implicit sigma (3) type condition for heavy cycles in weighted graphs," *Ars Combinatoria*, vol. 115, pp. 211–218, 2014.

[49] H. Liu and F. Meng, "Some new generalized Volterra-Fredholm type discrete fractional sum inequalities and their applications," *J. Inequal. Appl.*, vol. 2016, p. 213, Sep. 2016. doi: 10.1186/s13660-016-1152-7.

[50] P. Li and G. Ren, "Some classes of equations of discrete type with harmonic singular operator and convolution," *Appl. Math. Comput.*, vol. 284, pp. 185–194, Jul. 2016.

[51] B. Zhang, "Remarks on the maximum gap in binary cyclotomic polynomials," *Bull. Math. Soc. Sci. Math. Roumanie*, vol. 59, no. 1, pp. 109–115, 2016.

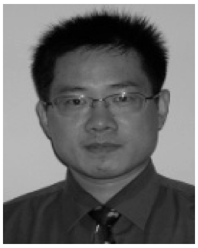
[52] L. Wang, "The fixed point method for intuitionistic fuzzy stability of a quadratic functional equation," *Fixed Point Theory Appl.*, vol. 2010, 2010, Art. no. 107182.

[53] L. L. Liu and D. Ma, "Some polynomials related to Dowling lattices and X-Stieltjes moment sequences," *Linear Algebra Appl.*, vol. 533, pp. 195–209, Nov. 2017.

[54] L. Li, F. Meng, and Z. Zheng, "Oscillation results related to integral averaging technique for linear Hamiltonian systems," *Dyn. Syst. Appl.*, vol. 18, no. 3, pp. 725–736, 2009.

[55] A. Xu and N. Ding, "Semidualizing bimodules and related Gorenstein homological dimensions," *J. Algebra Appl.*, vol. 15, no. 10, p. 1650193, 2016.

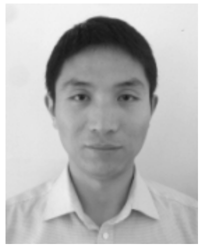
CHAO YAN received the master's degree from the Insititute of Computing Technology, Chinese Academy of Sciences, China, in 2006. He is currently a Lecturer with the School of Information Science and Engineering, Qufu Normal University, China. His research interests include recommender systems and services computing.



XINCHUN CUI received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China, in 2011. He is currently a Professor with the School of Information Science and Engineering, Qufu Normal University, China. His main research interests include information security and computational neuroimaging.

LIANYONG QI received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. He is currently an Associate Professor with the School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, China. He has published over 40 peer review papers in international journals and conferences. His research interests include recommender systems and services computing.

XIAOLONG XU received the B.Sc. and M.Sc. degrees in computer science from the Nanjing University of Information Science and Technology in 2010 and 2013, respectively, and the Ph.D. degree from Nanjing University, China, in 2016. He was a Research Scholar with Michigan State University, USA, from 2017 to 2018. He is currently a Lecturer with the School of Computer and Software, Nanjing University of Information Science and Technology. He has published over 30 peer review papers in international journals and conferences.



XUYUN ZHANG received the B.S. and M.E. degrees in computer science from Nanjing University, China, in 2011 and 2008, respectively, and the Ph.D. degree from the University of Technology, Sydney, Australia, in 2014. He was a Post-Doctoral Fellow with the Machine Learning Research Group, NICTA (currently Data61, CSIRO), Australia. He is currently a Lecturer with the Department of Electrical and Computer Engineering, The University of Auckland, New Zealand. His primary research interests include IoT and smart cities, big data, cloud computing, scalable machine learning and data mining, data privacy and security, and Web service technology.