RESEARCH **Open Access**

LSH-based private data protection for service quality with big range in distributed educational service recommendations

Chao Yan[1], Xuening Chen[2] and Qinglei Kong[3*]

Abstract

Service recommendation has become a promising way to extract useful or valuable information from big educational data collected by various sensors and distributed in different platforms. How to protect the private user data in each cluster during recommendation processes is an interesting but challenging problem in the educational domain. A locality-sensitive hashing (LSH) technique has recently been employed to achieve the abovementioned privacy-preservation goal. However, traditional LSH-based recommendation approaches often suffer from low accuracy when the service quality data recruited in recommendations vary in a big range. Considering this drawback, we modify the traditional LSH technique in this paper to make it applicable to the service quality data with a big range, so as to pursue privacy-preserving and an accurate recommended list. Finally, a wide range of experiments are conducted based on the distributed dataset, i.e., *WS-DREAM*. Experiment results show that our approach can protect the private data in education (e.g., student information in universities) and performs better than other state-of-the-art ones in terms of accuracy and efficiency.

Keywords: Recommender system, Cluster, Sensor, Service quality with big range, Privacy preservation

1 Introduction

In the age of Internet of Things (IoT), tremendous sensors embedded in various smart devices (e.g., smart phone and PDA) have produced a considerable amount of data and stored in different platforms, which form the so called big data [1–3]. In this situation, service recommendation has become one of the most effective techniques to extract valuable information from massive data in various domains, e.g., education. Typically, through analyzing the historical educational service usage data (e.g., service quality left by educational users) distributed in different platforms, a recommender system can mine the potential interest or preferences of a target user, through lightweight recommendation techniques (e.g., collaborative filtering (CF)). Then according to user preferences, the recommender system can return appropriate services to the target user, so as to satisfy his/her complex business requirements [4, 5].

However, from the perspective of educational users, the historical service quality data (stored in different clusters) used to make recommendation decisions are generally sensitive as they often contain some private information of users. Therefore, in order to protect business secrets and obey laws, it is necessary for a recommender system to protect the sensitive service quality data (e.g., quality of a service ever invoked by an educational user), especially when the data are collected via remote sensors and stored in distributed clusters [6, 7].

As an efficient and privacy-preserving technique for a similar neighbor search, the locality-sensitive hashing (LSH) [8] technique has recently been introduced into a service recommendation domain to protect the private information of users [9–11]. Typically, a recommender system first utilizes the LSH technique to build user indices with little privacy (even no privacy) and then search for the similar neighbors of a target user based on the pre-built user indices; this way, similar neighbors of the target user can be found easily, without revealing the sensitive service quality data generated from past service invocations. Finally, appropriate services are selected and

* Correspondence: 229339904@qq.com
[3]School of Medical Information Engineering, Jining Medical University, Rizhao, China
Full list of author information is available at the end of the article

recommended to the target user based on the derived similar neighbors of the target user, so as to complete the LSH-based service recommendation process.

Generally, the traditional LSH technique performs well in privacy-preserving service recommendations. However, when the service quality data vary in a big range (e.g., the *throughput* in *WS-DREAM* dataset [12], whose value range is [0.004, 1000] kbps), the accuracy of LSH-based service recommended results decreases significantly and hence cannot satisfy the target users' quick recommendation requirements. In view of this challenge, we improve the traditional LSH technique, so as to adapt the LSH-based recommendation approaches to the sensitive service quality data with a big range (e.g., *throughput* = 20 kbps when an educational user invoked a service). This way, we can enlarge the applicability of the LSH technique in privacy-preserving service recommendations and meanwhile improve the recommendation accuracy.

In summary, our major contributions are twofold.

(1) We recognize the limitation of the traditional LSH technique in terms of recommendation accuracy when the service quality data for recommendations vary in a big range. Furthermore, we improve the traditional LSH-based recommendation approaches to adapt to the service quality with a big range.

(2) A set of experiments are designed and conducted on a well-known service quality dataset, i.e., *WS-DREAM*, to validate the effectiveness and efficiency of our proposed privacy-preserving service recommendation approach. Experiment results show that our proposal outperforms other competitive approaches in terms of recommendation accuracy, when the historical service quality data are varied in a big range.

The rest of this paper is structured as below. Related work is presented in Section 2. In Section 3, a novel privacy-preserving service recommendation approach based on the improved LSH technique, named Rec_{LSH^*}, is proposed to protect the private service quality data with a big range. Experiment comparisons are presented in Section 4. Finally, in Section 5, we conclude the paper and discuss the possible research directions in the future.

2 Related work

In the IoT age, privacy preservation has become a hot research topic in various domains including service recommendations. Many researchers have concentrated on the privacy-preserving service recommendation problems and put forward their respective resolutions. In this section, we will introduce the related work from the following four aspects, respectively.

2.1 Anonymization

Anonymity is an effective approach to achieve the data security and privacy-preservation goals in many research domains. Through anonymization, multiple entries in a dataset become similar and indiscriminative; this way, the private information of users can be protected. As a variant of an anonymization technique, K-anonymity is recently recruited in [13] to generalize the real geographical information of users so as to protect the users' location privacy. One shortcoming of K-anonymity is that it is hard to determine the optimal value of parameter K. Besides, the data availability after K-anonymity is reduced significantly when K grows; therefore, the recommendation accuracy would be decreased accordingly.

2.2 Encryption

As a classic data protection mechanism, encryption is widely applied in various domains. Many variants of an encryption technique are put forward in recent years, such as homomorphic encryption [14]. Although these approaches can protect the sensitive information of users, the encryption cost and time delay are often high; therefore, they are often not suitable for the lightweight service recommendation scenarios where quick responses from recommender systems are required.

2.3 Data obfuscation

In [15], the real service quality data are obfuscated by adding a random value and then the obfuscated data are recruited to make service recommendations; this way, the private information hidden in real service quality data can be protected well. However, as the data used to make recommendations are obfuscated, the recommendation accuracy is reduced accordingly. Similarly, in [16], the differential privacy (DP) technique is employed to protect the user privacy when making recommendation decisions. However, the time cost is often high due to the high time complexity of DP. Besides, when the available data for recommendations are updated frequently, the accumulated noise incurred by DP will become larger; in this situation, the recommendation accuracy would be decreased accordingly.

2.4 Locality-sensitive hashing (LSH)

In refs. [3–6, 9–11], the LSH technique is combined with user-based CF to build user indices offline so as to make efficient and privacy-preserving service recommendations. Similarly, the LSH technique is recruited in [17] to build service indices offline so as to accelerate the similar service search process. Although the LSH technique can be recruited to protect the private information of users, the LSH-based service recommendation approaches often fail to generate an accurate recommended result when the service quality data used to make recommendation decisions are varied in a big range.

With the above analyses, we can conclude that LSH is a promising technique to achieve the goal of efficient and privacy-preserving service recommendations. However, the traditional LSH technique still fails to produce an accurate recommended result while protecting the private service quality data with a big range. Considering this drawback, we improve the traditional LSH technique in this paper to make it adapt to the service quality varied in a big range, so as to improve the applicability and the accuracy of LSH-based recommendation approaches.

3 Privacy-preserving web service recommendations based on service quality data with a big range

In this section, we introduce the details of our proposed privacy-preserving service recommendation approach, i.e., Rec_{LSH*}. Concretely, in Section 3.1, we briefly introduce the rationale of LSH to be used in our approach; in Section 3.2, we first improve the traditional LSH technique to make it applicable to the service quality with a big range, and then search for the neighbors of a target user based on the improved LSH technique; finally, in Section 3.3, we make service recommendations based on the target user's neighbors derived in Section 3.2.

3.1 Locality-sensitive hashing (LSH)

LSH is an effective and efficient technique to make similar neighbor search. The rationale behind LSH is as follows [8]: if two points are close enough, then they will be still neighbors with high probability after a LSH mapping; on the contrary, if two points are far away from each other, then they will not be neighbors with high probability after a LSH mapping. In summary, LSH is essentially a probability-based similar neighbor search technique. Additionally, the similar neighbors of a target user can be discovered based on the indices (with little privacy) after LSH, instead of the real service quality data; this way, the private information of users can be protected very well.

3.2 Improved LSH-based neighbor search based on service quality with a big range

As discussed in Section 1, the traditional LSH technique cannot guarantee high accuracy when the private service quality data vary in a big range. The reason is that LSH is essentially a probability-based neighbor search technique and the probability of successful neighbor search decreases when the service quality data for recommendations fluctuate significantly. In view of the above reason analyses, we first transform the service quality with a big range into a corresponding service quality with a small range through linear mappings; afterwards, we utilize the transformed

service quality with a small range to make similar neighbor search based on LSH. Next, we introduce the concrete processes of service quality transformation and similar neighbor search.

To facilitate the further discussions, we only consider one service quality dimension with a big range, denoted by q, instead of multiple dimensions in many application domains [18–27]; for a web service ws, its real quality over q is denoted by $ws.q$; the minimal and maximal quality values of q of all the candidate services are denoted by MIN_q and MAX_q, respectively (thus a big range $[\text{MIN}_q, \text{MAX}_q]$ can be obtained through statistic offline), while ws's quality over q after transformation is denoted by $T(ws.q)$. Next, we introduce how to transform the real service quality $ws.q$ ($\in[\text{MIN}_q, \text{MAX}_q]$) into its corresponding value $T(ws.q)$ ($\in[0, 1]$) (here, we select the small range $[0, 1]$ due to its popularity in the data normalization applications).

Concretely, if q is a positive quality dimension (the larger the better, e.g., *throughput*), then the normalization process can be finished by (1) where $ws.q \in [MIN_q, MAX_q]$ holds. While if q is a negative quality dimension (the smaller the better, e.g., *response time*), then the normalization process can be finished by (2) where $ws.q \in [MIN_q, MAX_q]$ holds. Thus, through the transformation equations in (1) and (2), we can obtain $T(ws.q)$ whose value falls into the range $[0, 1]$. Then according to the $T(ws.q)$ values of all candidate services invoked by all users in the past, we can make similar neighbor search for a target user (denoted by u_{target}) based on the LSH technique, in an efficient and privacy-preserving way. Next, we introduce the concrete process of LSH-based neighbor search.

$$T(ws.q) = \begin{cases} \dfrac{ws.q - \text{MIN}_q}{\text{MAX}_q - \text{MIN}_q} & \text{if } \text{MAX}_q \neq \text{MIN}_q \\ 1 & \text{if } \text{MAX}_q = \text{MIN}_q \end{cases}$$
$$(1)$$

$$T(ws.q) = \begin{cases} \dfrac{\text{MAX}_q - ws.q}{\text{MAX}_q - \text{MIN}_q} & \text{if } \text{MAX}_q \neq \text{MIN}_q \\ 1 & \text{if } \text{MAX}_q = \text{MIN}_q \end{cases}$$
$$(2)$$

Suppose there are n candidate services $\{ws_1, ..., ws_n\}$ and m users $\{u_1, ..., u_m\}$. Then for each user u_i ($1 \leq i \leq m$), the quality of his/her invoked services can be denoted by the n-dimensional vector $\vec{u_i} = (T(ws_1.q), ..., T(ws_n.q))$. Specifically, $T(ws_j.q) = 0$ ($1 \leq j \leq n$) if user u_i has never invoked service ws_j before. Afterwards, we can transform the sensitive vector $\vec{u_i}$ into its corresponding hash value $h(u_i)$ with little privacy, based on the LSH function $h(.)$ in (3). Here, $\vec{v}$ is an n-dimensional vector $(v_1, ..., v_n)$ where v_j is a random

value in the range $[-1, 1]$; symbol "∘" denotes the dot product of two vectors.

$$h(u_i) = \begin{cases} 1 & \text{if } \overrightarrow{u_i} \circ \overrightarrow{v} > 0 \\ 0 & \text{if } \overrightarrow{u_i} \circ \overrightarrow{v} \le 0 \end{cases} \qquad (3)$$

We repeat the above hash process (for user u_i) r times based on the randomly selected r hash functions $\{h_1(.), ..., h_r(.)\}$. Afterwards, an r-dimensional vector $H(u_i) = (h_1(u_i), ..., h_r(u_i))$ is obtained, which can be regarded as the index for user u_i according to the traditional LSH theory. Then all the users and their respective indices form a hash table (denoted by *Hash_Tb*). With the derived hash table, we can make quick neighbor search for the target user u_{target}. Concretely, if the indices of u_{target} and u_i are equal in the hash table *Hash_Tb*, i.e., $H(u_i) = H(u_{\text{target}})$, then we can conclude that u_i is a neighbor of u_{target}.

However, LSH is a probability-based neighbor search technique; therefore, one hash table is often not enough to guarantee to find out all the similar neighbors of a target user. In other words, "false-negative" (i.e., similar neighbors of the target user are overlooked by mistake) search results are possible. Therefore, in order to reduce the "false-nega-tive" probability, multiple hash tables are often needed. Concretely, we repeat the abovementioned hash table building process L times to get L hash tables $\{Hash\_Tb_1, ..., Hash\_Tb_L\}$. Then we relax the neighbor search condi-tion in the last paragraph as follows: if $H(u_i) = H(u_{\text{target}})$ holds in any hash table $Hash\_Tb_x$ $(1 \le x \le L)$, then u_i can be regarded as a neighbor of u_{target}. This way, we can obtain more neighbors (denoted by set *Neighbor_set*) of the target user.

Here, we utilize the user index $H(u_i)$ $(1 \le i \le m)$, in-stead of the transformed service quality vector $\overrightarrow{u_i}$, to make similar neighbor search due to the following two reasons. First of all, $H(u_i)$ contains less private informa-tion of users compared to $\overrightarrow{u_i}$. Second, the L hash tables can be built offline; therefore, the neighbor search process based on $H(u_i)$ and $H(u_{\text{target}})$ would be finished quickly, which can accelerate the speed of subsequent recommendation process significantly. Therefore, with the help of user indices, we can search for the neighbors of a target user in a privacy-preserving and efficient way. Next, we introduce how to predict the missing service quality and make appropriate service recommendations based on the derived neighbors in *Neighbor_set*.

3.3 Service recommendation
For each service ws_j $(1 \le j \le n)$ never invoked by u_{target}, its missing service quality (after quality transformation) over dimension q, i.e., $T(ws_j.q)$, can be predicted by the equation in (4). Here, set $Neighbor\_set_j$ includes all the neighbors of u_{target} who have ever invoked service ws_j

before; $T_u(ws_j.q)$ denotes the quality over dimension q of service ws_j invoked by user u.

$$T(ws_j.q) = \frac{1}{|Neighbor\_set_j|} \overset{*}{\underset{u \in Neighbor\_set_j}{\sum}} T_u(ws_j.q) \qquad (4)$$

With the predicted service quality $T(ws_j.q)$ after quality transformation, we can reversely calculate the predicted service quality before quality transformation, i.e., $ws_j.q$, by (5) (for those positive quality dimensions) and (6) (for those negative quality dimensions). These two equations are de-duced from (1) and (2), respectively. Then for each service never invoked by u_{target}, its missing service quality can be predicted by (5) and (6). Finally, we select the service with the optimal predicted quality and recommend it to u_{target}.

$$ws_j.q = T(ws_j.q)^* (\text{MAX}_q - \text{MIN}_q) + \text{MIN}_q \qquad (5)$$

$$ws_j.q = \text{MAX}_q - T(ws_j.q)^* (\text{MAX}_q - \text{MIN}_q) \qquad (6)$$

4 Experiments
4.1 Experiment configurations
To simulate the private data protection scenarios in the distributed environment, our experiments are conducted on a well-known distributed service quality dataset named *WS-DREAM* [12]. The dataset consists of the quality data (i.e., *response time* and *throughput*) of 5825 services invoked by 339 users around the world. In this paper, we only focus on the service quality with a big range; therefore, the quality dimension *throughput* whose values belong to the big range [0.004 kbps, 1000 kbps] is considered in our experiments. Because the quality values of *response time* are varied in a small range [0.001 s, 19.999 s], therefore, *response time* is not a quality dimension that we are interested in and hence is not considered here.

We test the time cost and MAE (mean absolute error, the smaller the better) to evaluate the performances of our proposed service recommendation approach, i.e., Rec_{LSH^*}. Furthermore, to validate the advantages of Rec_{LSH^*}, we compare our proposal with the following four recommendation approaches:

(1) $DistSR_{LSH}$ [28]: our previous work which employs the traditional LSH technique to search for the similar users of a target user, in a privacy-preserving way.

(2) *UPCC*: user-based collaborative filtering which recruits the PCC (Pearson correlation coefficient) as the similarity measurement.

(3) *P-UIPCC* [15]: data obfuscation technique (each sensitive service quality value is added a random

number) is recruited to protect the private service quality.

(4) *PPICF* [29]: sensitive service quality data are divided into several quality segments with little privacy, and then the quality segments are employed to make service recommendations.

The experiments were deployed on a Dell laptop with a 2.80-GHz CPU and 2.0-GB RAM. Experiments are running under Windows XP and JAVA 1.5. Each experiment is repeated 10 times, and the average results are reported.

4.2 Experiment results

Concretely, four profiles are tested and compared in our experiments. Here, β ($\in$[0%, 100%]) denotes the density of user-service quality matrix; r and L represent the number of hash functions and the number of hash tables recruited in our approach, respectively.

4.2.1 Profile 1: Recommendation accuracy comparisons of five approaches (w.r.t. β)

In this profile, we test the accuracy of our proposed Rec_{LSH*} approach and compare it with the other four approaches. The parameters are set as follows: β is varied from 5 to 25%, $r = L = 10$ holds. The experiment results are shown in Fig. 1. As Fig. 1 indicates, the recommendation accuracy of the Rec_{LSH*} approach outperforms the other four approaches, because the LSH technique is improved and recruited in Rec_{LSH*} to adapt the service quality data with a big range, while the other four approaches do not discriminate the service quality dimensions with different value ranges; therefore, the

recommendation accuracy is decreased when the service quality for recommendations varies in a big range. Another observation from Fig. 1 is that in our Rec_{LSH*} approach, the recommendation accuracy does not render a linear variation tendency with the density of user-service quality matrix, i.e., β.

4.2.2 Profile 2: Recommendation efficiency comparisons of five approaches (w.r.t. β)

Efficiency or scalability is another criterion to evaluate the performances of a recommender system. Therefore, in this profile, we test and compare the efficiencies of different service recommendation approaches. The experiment parameter settings are as follows: the matrix density β is varied from 5 to 25%, $r = L = 10$ holds. The concrete experiment results are presented in Fig. 2.

As Fig. 2 shows, the recommendation efficiencies of Rec_{LSH*} and $DistSR_{LSH}$ approaches are better than the other three approaches; this is because the LSH technique is adopted in both Rec_{LSH*} and $DistSR_{LSH}$, while LSH is rather efficient in searching for the neighbors of a target user as most jobs (e.g., the building of user index tables) can be done offline. Besides, Rec_{LSH*} consumes a little more time than $DistSR_{LSH}$ as the service quality data with a big range need to be transformed in Rec_{LSH*}. However, as demonstrated in Fig. 2, our proposed Rec_{LSH*} approach is still efficient enough compared to the *UPCC*, *P-UIPCC*, and *PPICF* approaches.

4.2.3 Profile 3: Recommendation accuracy of the Rec_{LSH*} approach w.r.t. r and L.

As introduced in Section 3.1, LSH is essentially a probability-based neighbor search technique; therefore,
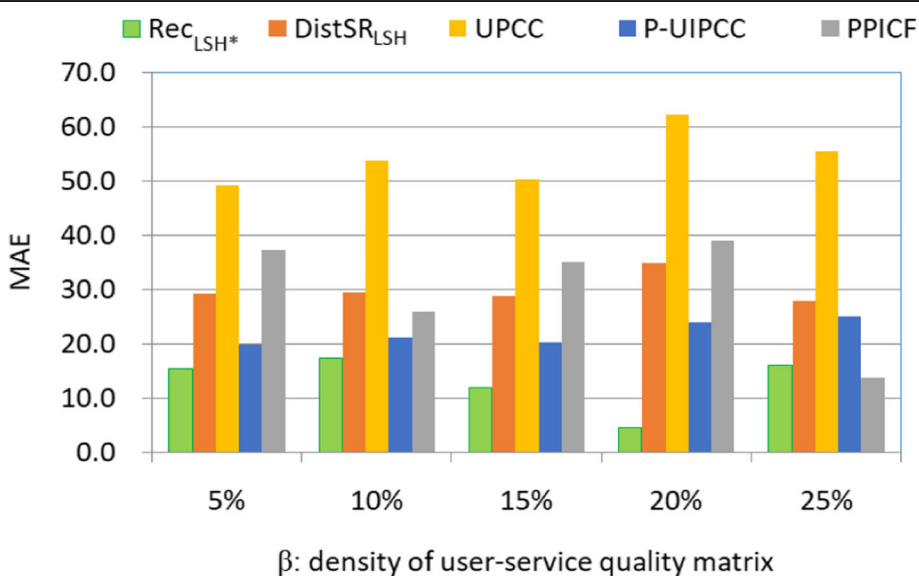

Fig. 1 Recommendation accuracy comparison (w.r.t. β). Detailed legend: *Rec_{LSH*}, DistSR_{LSH}, UPCC, P-UIPCC, PPICF*
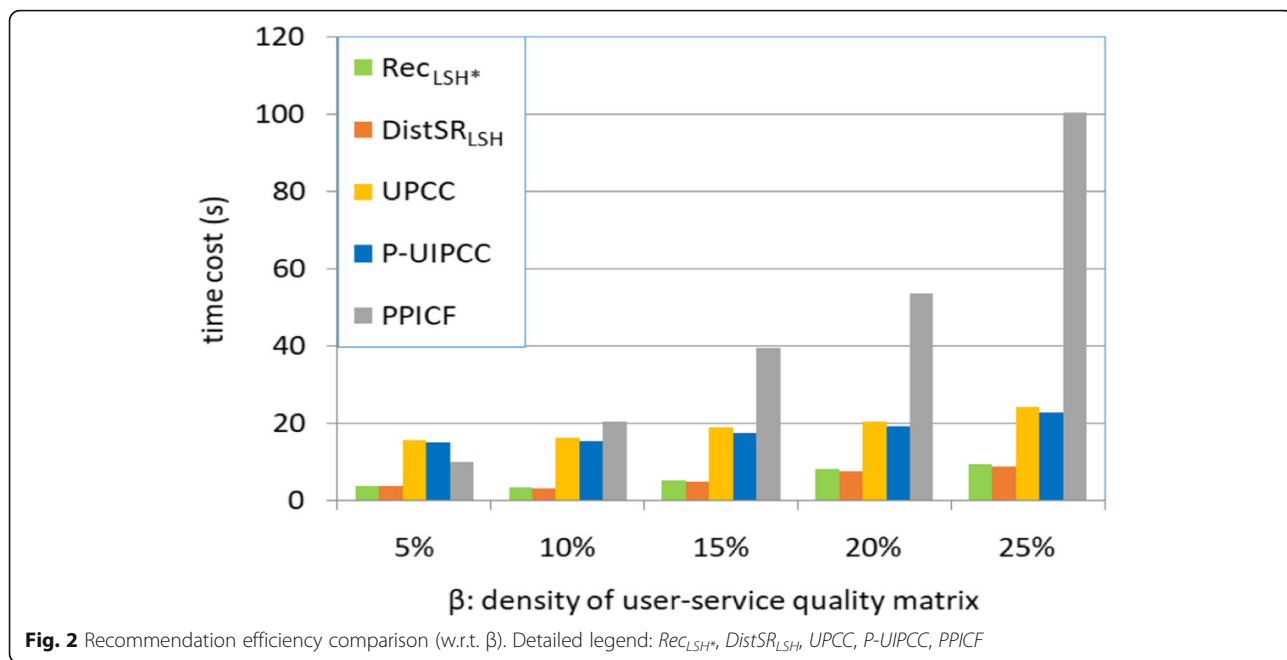
Fig. 2 Recommendation efficiency comparison (w.r.t. β). Detailed legend: *Rec_LSH*, DistSR_LSH, UPCC, P-UIPCC, PPICF*

the accuracy values of LSH-based service recommendation approaches are often correlated with several LSH parameters, such as the number of hash functions (i.e., r) and the number of hash tables (i.e., L). Considering this, in this profile, we test the relationship or correlation between the recommendation accuracy of the Rec_{LSH*} approach and the parameters r and L. The experiment parameters are set as below: $\beta = 25\%$; r is varied from 2 to 12; L is varied from 10 to 20. The concrete experiment execution results are presented in Fig. 3.

As Fig. 3 shows, the recommendation accuracy of the Rec_{LSH*} approach approximately decreases (i.e., MAE increases approximately) with the growth of L. This is because according to the nature of LSH, a larger L value (i.e., more hash tables) often means looser search condition for neighbors; in this situation, some neighbors who are not very similar with the target user would be returned to make service recommendations, and therefore, the recommendation accuracy is decreased to some extent accordingly. Another result that can be observed from Fig. 3 is that the recommendation accuracy of our
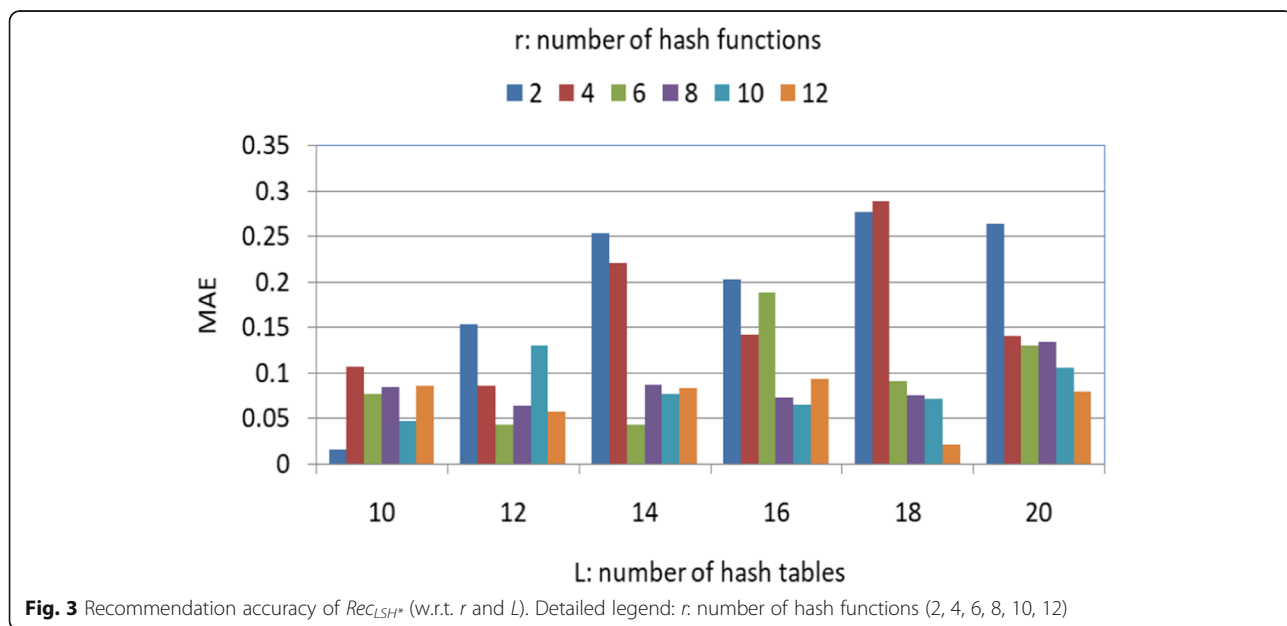

Fig. 3 Recommendation accuracy of Rec_{LSH*} (w.r.t. r and L). Detailed legend: r: number of hash functions (2, 4, 6, 8, 10, 12)

Rec_{LSH*} approach approximately increases (i.e., MAE drops approximately) when r rises. This is because a larger r value (i.e., more hash functions) often means stricter search condition for similar neighbors of a target user. In this situation, only the neighbors who are very similar with the target user would be found and recruited to make recommendation decisions; therefore, the service recommendation accuracy of our Rec_{LSH*} approach is increased accordingly.

4.2.4 Profile 4: Number of neighbors in Rec_{LSH*} w.r.t. r, L.

The number of derived neighbors of a target user is an important reference criterion to evaluate the performance of CF-based service recommendation approaches. In this profile, we measure the number of neighbors of our proposed Rec_{LSH*} approach and investigate its possible relationship with the LSH parameters r and L. The concrete experiment parameters are set as below: $\beta = 25\%$; r is varied from 2 to 12; L is varied from 10 to 20. The experiment results are shown in Fig. 4.

As can be seen from Fig. 4, there is an obvious decline of the number of similar neighbors when the number of hash functions, i.e., r, grows. The reason is the same as that in Fig. 3. More concretely, a larger r value often means stricter search condition for similar neighbors, and therefore, fewer neighbors of a target user would be found according to the stricter search condition. Additionally, when the number of hash tables, i.e., L, grows, the number of returned neighbors is increased approximately. This is because the search condition for neighbors becomes looser when L grows, and in this situation, more users would be regarded as qualified neighbors of a target user according to the looser search condition.

4.3 Discussions

In our approach and experiments, we only consider one quality dimension whose value is in a big range (e.g., *throughput* dimension in *WS-DREAM* dataset), without considering the multiple quality dimensions as well as their complex relationships, e.g., linear correlations [30–33], nonlinear correlations [34–48], and weight significance [49–55]. So in the future, we will further discuss this more complex multi-dimensional service recommendation scenario. Besides, in this paper, we only consider the service quality dimension whose value is a real number, while neglecting the diversity of data styles (e.g., integer number, real number, fuzzy number) [56–60]. Considering this drawback, we will investigate more comprehensive service recommendation approach that integrates diverse data types. Finally, due to the intrinsic shortcoming of LSH technique, it is hard to evaluate the privacy-preservation performance of our suggested Rec_{LSH*} approach. In the future work, we will further investigate this how to solve this key issue.

5 Conclusions

Protecting the private information of educational users hidden in the service quality data distributed across different platforms is a key precondition of successful service recommendations in the distributed environment. LSH has been proven as an effective technique to make privacy-preserving and efficient service recommendations. However, traditional LSH-based service recommendation approaches often fail to return a high-quality recommended list when the sensitive service quality data for recommendations vary in a big range. Considering this drawback, we improve the traditional LSH technique in
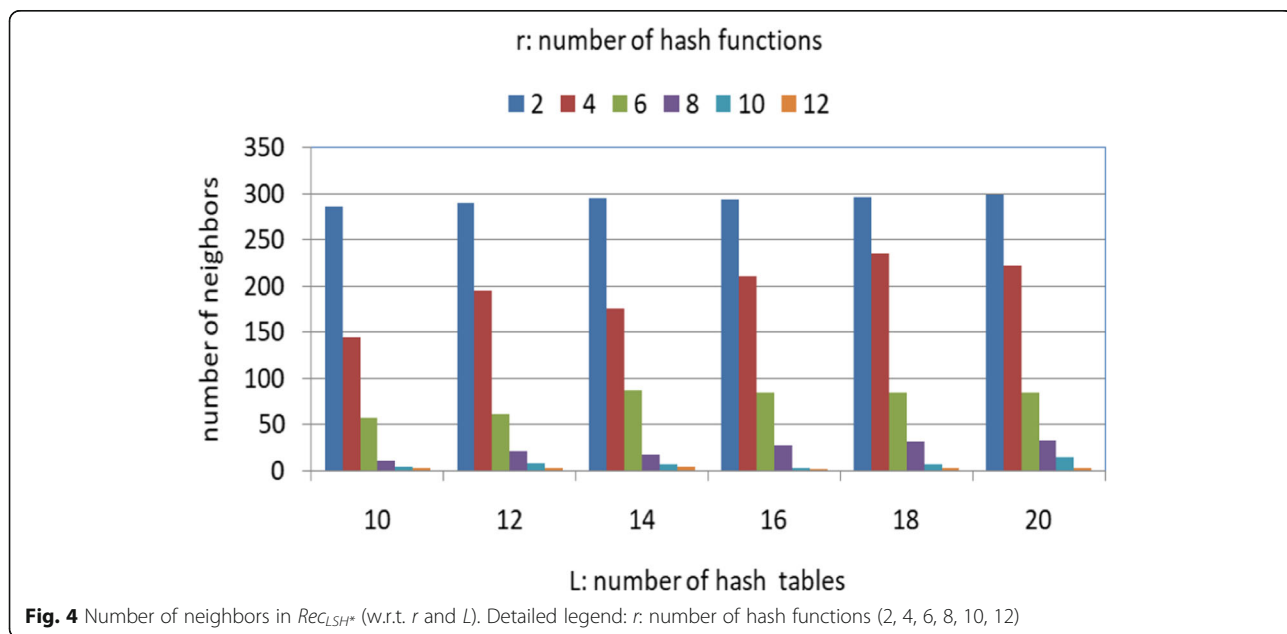

Fig. 4 Number of neighbors in Rec_{LSH*} (w.r.t. r and L). Detailed legend: r: number of hash functions (2, 4, 6, 8, 10, 12)

this paper to make it applicable to the service quality data with a big range. Through the experiments on *WS-DREAM* dataset, we prove the feasibility of the Rec_{LSH^*} approach.

However, we only discuss one quality dimension of services in recommendation decisions for simplicity, without considering the multiple quality dimensions as well as their linear correlations, nonlinear correlations, data styles, and weights. So in the future, we will further refine our work by considering multiple service quality dimensions simultaneously. Additionally, service quality often varies with time; so time-aware recommendation is another research direction in the future.

Abbreviations
CF: Collaborative filtering; DP: Differential privacy; IoT: Internet of Things; LSH: Locality-sensitive hashing

Acknowledgements
Not applicable.

Availability of data and materials
The recruited experiment dataset *WS-DREAM* is available at wsdream.github.io/.

Authors' contributions
CY finished the algorithm and English writing of the paper. XC finished the experiments. QK put forward the idea of this paper. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]School of Information Science and Engineering, Qufu Normal University, Rizhao, China. [2]Student Affairs Office, Qufu Normal University, Rizhao, China. [3]School of Medical Information Engineering, Jining Medical University, Rizhao, China.

References
1. C. Li et al., Predicting short-term electricity demand by combining the advantages of ARMA and XGBoost in fog computing environment. Wirel. Commun. Mob. Comput. **2018**(5018053), 18 (2018)
2. L. Qi et al., Time-aware IoE service recommendation on sparse data. Mob. Inf. Syst. **2016**(4397061), 12 (2016)
3. L. Kuang et al., A personalized QoS prediction approach for CPS service recommendation based on reputation and location-aware collaborative filtering. Sensors **18**(5), 1556, 1–32 (2018)
4. Y. Xu et al., Privacy-preserving and scalable service recommendation based on SimHash in a distributed cloud environment. Complexity **2017**(3437854), 9 (2017)
5. L. Qi et al., A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. IEEE. J. Sel Areas Commun. **35**(11), 2616–2624 (2017)
6. W. Gong et al., Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wirel. Commun.Mob. Comp. **2018**(3075849), 8 (2018)
7. L. Qi et al., Time-aware distributed service recommendation with privacy-preservation. Inf. Sci. **480**, 354–364 (2019)
8. A. Gionis et al., *Similarity search in high dimensions via hashing*. Proceedings of the 25th International Conference on Very Large Databases (VLDB), (Edinburgh, 1999), pp. 518–529
9. L. Qi et al., A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Futur. Gener. Comput. Syst. **88**, 636–643 (2018)
10. C. Yan et al., Privacy-aware data publishing and integration for collaborative service recommendation. IEEE ACCESS **6**, 43021–43028 (2018)
11. L. Qi et al., An exception handling approach for privacy-preserving service recommendation failure in a cloud environment. Sensors **18**(7), 1–11 (2018)
12. Z. Zheng et al., Investigating QoS of real world web services. IEEE Trans. Serv. Comput. **7**(1), 32–39 (2014)
13. I. Memon, Authentication user's privacy: an integrating location privacy protection algorithm for secure moving objects in location based services. Wirel. Pers. Commun. **82**(3), 1585–1600 (2015)
14. S. Sobitha Ahila et al., Role of agent technology in web usage mining: homomorphic encryption based recommendation for E-commerce applications. Wirel. Pers. Commun. **87**(2), 499–512 (2016)
15. J. Zhu et al., *A privacy-preserving QoS prediction framework for web service recommendation*. 2015 IEEE International Conference on Web Services (ICWS), (New York, 2015), pp. 241–248
16. L. Kuang et al., A privacy protection model of data publication based on game theory. Secur. Commun Netw. **2018**(3486529), 13 (2018)
17. D. Li et al., *SES-LSH: shuffle-efficient locality sensitive hashing for distributed similarity search*. 2017 IEEE International Conference on Web Services (ICWS), (Honolulu, 2017), pp. 822–827
18. M. Wang et al., Robust group non-convex estimations for high-dimensional partially linear models. J. Nonparametr. Statistics **28**(1), 49–67 (2016)
19. X. Wang et al., Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure. J. Appl. Statistics **43**(5), 796–809 (2016)
20. P. Wang et al., Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds. Nonlinear Anal. **130**, 1–17 (2016)
21. X. Wang et al., Adaptive group bridge estimation for high-dimensional partially linear models. J. Inequal. Appl. **2017**, 158 (2017). https://doi.org/10.1186/s13660-017-1432-x
22. X. Wang et al., Restricted profile estimation for partially linear models with large-dimensional covariates. Statist. Probab. Lett. **128**, 71–76 (2017)
23. H. Tian et al., Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth Integrable systems. J. Differ Equ. **263**, 7448–7474 (2017)
24. P. Wang et al., The geometric properties of harmonic function on 2-dimensional Riemannian manifolds. Nonlinear Anal. **103**, 2–8 (2014)
25. M. Wang et al., Adaptive lasso estimators for ultrahigh dimensional generalized linear models. Statist. Probab. Lett. **89**, 41–50 (2014)
26. M. Wang et al., A note on the one-step estimator for ultrahigh dimensionality. J. Comput. Appl. Math. **260**, 91–98 (2014)
27. G. Tian et al., Variable selection in the high-dimensional continuous generalized linear model with current status data. J. Appl. Stat. **41**, 467–483 (2014)
28. L. Qi et al., *Privacy-preserving distributed service recommendation based on locality-sensitive hashing*. 2017 IEEE International Conference on Web Services (ICWS), (Honolulu, 2017), pp. 49–56
29. D. Li et al., An algorithm for efficient privacy-preserving item-based collaborative filtering. Futur. Gener. Comput. Syst. **55**, 311–320 (2016)
30. G. Guo et al., Parallel tempering for dynamic generalized linear models. Commun. Statist.-Theory Methods **45**(21), 6299–6310 (2016)
31. L.L. Liu et al., Recurrence relations for linear transformations preserving the strong q-log-convexity. Electron. J. Comb. **23**(3), 1–11 (2016)
32. H. Li et al., Partial condition number for the equality constrained linear least squares problem. Calcolo **54**(4), 1121–1146 (2017)
33. Z. Zhao et al., Existence and uniqueness of positive solutions for some singular boundary value problems with linear functional boundary conditions. Acta Math. Sin. Engl. Ser. **27**(10), 2073–2084 (2011)
34. H. Liu et al., Some new nonlinear integral inequalities with weakly singular kernel and their applications to FDEs. J. Inequal. Appl. **2015**(209), 1–17 (2015)
35. X. Zhang et al., Entire large solutions for a Schrödinger systems with a nonlinear random operator. J. Math. Anal. Appl. **423**(2), 1650–1659 (2015)
36. Z. Zong et al., On Jensen's inequality, Holder's inequality and Minkowski's inequality for dynamically consistent nonlinear evaluations. J. Inequal. Appl. **2015**(152), 1–18 (2015)

37. X. Hao et al., Positive solutions for nonlinear fractional semipositone differential equation with nonlocal boundary conditions. J. Nonl. Sci. Appl. **9**(6), 3992–4002 (2016)
38. X. Hao et al., Iterative solution for nonlinear impulsive advection-reaction-diffusion equations. J. Nonl. Sci. Appl. **9**(6), 4070–4077 (2016)
39. J. Shao et al., Oscillation theorems for second order forced neutral nonlinear differential equations with delayed argument. Int. J. Diff. Equ. **2010**(181784), 1–15 (2010)
40. Y. Bai et al., On a class of Volterra nonlinear equations of parabolic type. Appl. Math. Comput. **2010**(216), 236–240 (2010)
41. Y. Bai, Backward solutions to nonlinear integro-differential systems. Cen. Eur. J. Math. **8**(4), 807–815 (2010)
42. F. Li et al., Uniform energy decay rates for nonlinear viscoelastic wave equation with nonlocal boundary damping. Nonlinear Anal. **74**, 3468–3477 (2011)
43. F. Li et al., Global existence uniqueness and decay estimates for nonlinear viscoelastic wave equation with boundary dissipation. Nonlinear Anal. Real World Appl. **12**, 1770–1784 (2011)
44. A. Qian, Sing-changing solutions for some nonlinear problems with strong resonance. Bound. Value Probl. **18**, 1–9 (2011)
45. Y. Wang et al., Positive solutions for a class of fractional boundary value problem with changing sign nonlinearity. Nonlinear Anal. **74**(17), 6434–6441 (2011)
46. W. Fan et al., Nontrivial solutions of singular fourth-order Sturm-Liouville boundary value problems with a sign-changing nonlinear term. Appl Math. Comput. **217**(15), 6700–6708 (2011)
47. L.L. Liu, Linear transformations preserving log-convexity. Ars Combin. **100**, 473–483 (2011)
48. J. Liu et al., Multiple positive solutions for second-order three-point boundary-value problems with sign changing nonlinearities. *Electronic*. J. Differ. Equ. **2012**(152), 1–7 (2012)
49. S. Yang et al., The weight distributions of two classes of p-ary cyclic codes with few weights. Finite Fields Appl. **44**, 76–91 (2017)
50. Y. Wang et al., Uniform estimate for the tail probabilities of randomly weighted sums. Acta Math. Appl. Sin. Engl. Ser. **30**(4), 1063–1072 (2014)
51. S. Yang et al., A class of three-weight linear codes and their complete weight enumerators. Cryptogr. Commun. **9**, 133–149 (2017)
52. J. Cai, An implicit sigma (3) type condition for heavy cycles in weighted graphs. Ars Combin. **115**, 211–218 (2014)
53. S. Yang et al., Complete weight enumerators of a family of three-weight linear codes. Des Codes Cryptogr **82**, 663–674 (2017)
54. S. Yang et al., Complete weight enumerators of a class of linear codes. Discret. Math. **340**, 729–739 (2017)
55. S. Yang et al., A construction of linear codes and their complete weight enumerators. Finite Fields Appl. **48**, 196–226 (2017)
56. P. Li, Two classes of linear equations of discrete convolution type with harmonic singular operators. Complex Variables Elliptic Equ. **61**(1), 67–75 (2016)
57. Z. Zheng, Invariance of deficiency indices under perturbation for discrete Hamiltonian systems. J. Differ. Equ. Appl. **19**(8), 1243–1250 (2013)
58. M. Han et al., Theory of rotated equations and applications to a population model. Discrete Contin. Dynam. Systems-A **38**(4), 2171–2185 (2018)
59. J. Cai et al., A new sufficient condition for pancyclability of graphs. Discret. Appl. Math. **162**, 142–148 (2014)
60. L.L. Liu et al., Strong q-log-convexity of the Eulerian polynomials of Coxeter groups. Discrete Math. **338**(12), 2332–2340 (2015)