sensors

MDPI

Article

An Exception Handling Approach for Privacy-Preserving Service Recommendation Failure in a Cloud Environment

Lianyong Qi [1,*], Shunmei Meng [2,3], Xuyun Zhang [4], Ruili Wang [5], Xiaolong Xu [6], Zhili Zhou [6] and Wanchun Dou [3]

[1] School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China
[2] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China; mengshunmei@njust.edu.cn
[3] State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China; douwc@nju.edu.cn
[4] Department of Electrical and Computer Engineering, University of Auckland, Auckland 1023, New Zealand; xuyun.zhang@auckland.ac.nz
[5] Institute of Natural and Mathematical Sciences, Massey University, Auckland 0745, New Zealand; Ruili.WANG@MASSEY.AC.NZ
[6] School of Computer and Software, Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China; xlxu@nuist.edu.cn (X.X.); zhou_zhili@163.com (Z.Z.)
* Correspondence: lianyongqi@gmail.com; Tel.: +86-0633-3981060

Received: 23 May 2018; Accepted: 22 June 2018; Published: 26 June 2018

Abstract: Service recommendation has become an effective way to quickly extract insightful information from massive data. However, in the cloud environment, the quality of service (QoS) data used to make recommendation decisions are often monitored by distributed sensors and stored in different cloud platforms. In this situation, integrating these distributed data (monitored by remote sensors) across different platforms while guaranteeing user privacy is an important but challenging task, for the successful service recommendation in the cloud environment. Locality-Sensitive Hashing (LSH) is a promising way to achieve the abovementioned data integration and privacy-preservation goals, while current LSH-based recommendation studies seldom consider the possible recommendation failures and hence reduce the robustness of recommender systems significantly. In view of this challenge, we develop a new LSH variant, named converse LSH, and then suggest an exception handling approach for recommendation failures based on the converse LSH technique. Finally, we conduct several simulated experiments based on the well-known dataset, i.e., Movielens to prove the effectiveness and efficiency of our approach.

Keywords: service recommendation; privacy-preservation; failure; exception handling; converse Locality-Sensitive Hashing

1. Introduction

With the advent of Web of Things (WoT), an increasing number of enterprises or organizations are apt to encapsulate their products (e.g., web API (Application Programming Interface)) into easy-to-access web services and publish them on the web so as to attract potential users and gain more profits. However, the ever-increasing volume and varieties of candidate services place a heavy burden on the service selection decisions of target users [1]. Under the circumstance, Collaborative Filtering (CF)-based recommendation techniques are proposed to minimize such burdens. Typically, for a target user who requires recommend services, the recommender system can look for his/her similar friends by observing the quality of service (QoS) data monitored by various sensors, and then enact appropriate recommendation decisions with the help of derived friends. Nowadays, CF technique has been successfully applied in many recommender systems whose decision-making data for recommendation are organized or stored in a centralized way.

However, in the cloud computing environment, the QoS information that is crucial to recommendation decisions is often not centralized, but rather monitored by distributed sensors and stored in different cloud platforms [2]. In this situation, it is necessary for a recommender system to integrate or fuse these distributed data across different cloud platforms quickly and properly, so as to make comprehensive and accurate recommendation decisions. In particular, to protect the sensitive business information and obey the laws [3–5], preserving user privacy during the abovementioned multi-source data integration process is an important but challenging task [6–10] for the success of subsequent recommendations.

The Locality-Sensitive Hashing (LSH) technique [11] has recently been recruited to make efficient and privacy-preserving service recommendation in the distributed environment. Typically, according to the QoS data, the Locality-Sensitive Hashing technique can be used to search for the similar friends of a target user in an efficient and privacy-preserving manner. Afterwards, recommended results are generated by considering the preferences of obtained similar friends. However, in certain situations, the recommender system cannot generate or produce any satisfying recommended result; in other words, a recommendation failure occurs. While existing LSH-based service recommendation approaches seldom consider this kind of recommendation failure problems as well as the corresponding exception handling solutions; therefore, the robustness of the recommender system is reduced significantly.

An intuitive example is presented in Figure 1, which contains three users and six services. The user ratings are denoted by 1*–5*. According to the traditional LSH technique, the index values of *Tom* and *Alice* are not same as they have no co-invoked services. Therefore, *Tom* is not similar with *Alice*. Likewise, as Figure 1 shows, *Tom* is not similar with *Bob* either. In this situation, no satisfying candidate services can be recommended or returned to *Tom*, i.e., the service recommendation process is failed.

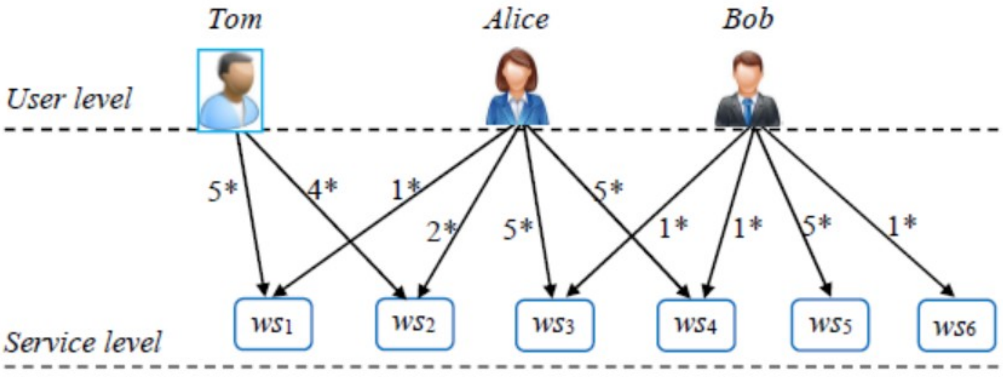

Figure 1. A recommendation failure example (see our previous work [12]).

In view of this challenge, we propose converse LSH technique and utilize it to look for a target user's contrary users (denoted by "enemy" in this paper) whose preferences are totally different from the target user. Afterwards, according to the enemies of the target user, we infer the possible friends of the target user indirectly so as to handle the exception incurred by recommendation failures. Overall, the contributions in this paper are as follows:

(1) A novel LSH variant named converse LSH is developed, which can be utilized to search for the enemy users of a target user, in a time-efficient and privacy-preserving way.

(2) We utilize converse LSH technique to search for the enemies of a target user and then look for the target user's similar friends indirectly based on the "enemy's enemy is a possible friend" inference rule in Social Balance Theory. Afterwards, we generate recommended results by considering the preferences of obtained similar friends, so as to handle the exceptions incurred by recommendation failures.

(3) Comprehensive experiments are simulated based on Movielens dataset, to test the effectiveness of suggested recommendation approach. Experiment results indicate the advantages of our proposal compared to other competitive approaches when a recommendation failure occurs.

The rest of this paper is structured as follows: in Section 2, we introduce the related work. Converse LSH technique is proposed in Section 3. An exception handling approach based on converse LSH is put forward in Section 4, to achieve indirect friend finding and service recommendations. Experiment evaluations are given in Section 5. In Section 6, we summarize the paper and point out the future research directions.

2. Related Work

Many researchers have investigated the privacy concerns in recommendation process and provide their respective resolutions. In [13], the authors suggested that a user can publish partial QoS data to the service community, so as to protect the remaining majority of QoS data. Similarly, in [14], the authors take the amount of published data as a tunable parameter and then transform the privacy-preservation problem into a multi-object optimization problem, so as to achieve a good tradeoff between data availability and data privacy. However, in the above approaches, certain sensitive information about users may be in danger due to the published partial data. Besides, recommendation failures are not considered in these approaches.

Microaggregation idea is adopted in [15] to realize data K-anonymization so that the users' sensitive data (e.g., user location) can be protected. However, there is often a tradeoff between data availability and data privacy; so the recommendation accuracy is often not as high as expected if the anonymous data are employed to make service recommendation decisions. Besides, these approaches do not discuss the possible service recommendation failures. Encryption technique is adopted in work [16] to guarantee the privacy-preservation of sensitive information. However, as a heavy-weight privacy-preservation manner, encryption operations often lead to high computational cost and long delay; therefore, the encryption techniques are often not applicable to the light-weight service recommendation requirements from certain users. Besides, recommendation failures are out of the scope of these encryption-based approaches.

Randomized disturbance idea is adopted in [17] to convert the real QoS data into the disturbed data; afterwards, the latter data are regarded as the recommendation bases to achieve the privacy-preservation goal. However, recommendation failures are not discussed; besides, the applicability of this approach is relatively limited as it can only be applied to the Pearson Correlation Coefficient (PCC)-based collaborative service recommendation scenarios. In [18], the authors utilize the Differential Privacy technique to make noise data injection and confusion, so as to ensure that the real service quality data would not be exposed to the outside. However, the time complexity of Differential Privacy is relatively high; second, when the service quality data are updated frequently, the accumulated noise would be enlarged, which will decrease the service recommendation accuracy accordingly; third, they do not consider the recommendation failures.

As an effective and efficient way to search for similar friends in the big data context, LSH is recently introduced into service recommendation to achieve the distributed data integration and privacy-preservation goals. In our previous work [19–21], LSH is combined with user-based CF to make privacy-preserving service recommendation. Likewise, in [22], LSH is combined with item-based CF to build service index table with little privacy and then make service recommendation based on the service index table. However, these LSH-based recommendation approaches do not consider the recommendation failures as well as the corresponding exception handling resolutions. As to the recommendation failures, the authors in [23] adopt the average idea to predict the missing QoS data. Social Balance Theory is utilized in our previous works [12,24] to look for the possible friends of a target user so as to cope with the recommendation failures. However, privacy concerns are not discussed.

Through the above literature review, a conclusion can be drawn that existing recommendation approaches either fail to protect user privacy or overlook the recommendation failures and exceptions. In view of this drawback, we propose converse LSH technique and utilize it to handle the exceptions incurred by recommendation failures; this way, the robustness of recommender systems can be improved significantly.

3. Converse Locality-Sensitive Hashing

Traditional LSH is an effective similar neighbor search technique. Therefore, in the service recommendation domain, LSH is often integrated with user-based Collaborative Filtering (CF) technique to search for the similar friends of a target user, in an efficient and privacy-preserving way. In this section, we modify the traditional LSH technique and transform it into converse LSH which can be used to search for the enemies of a target user efficiently while guaranteeing user privacy. Next, we introduce the rationale of converse LSH.

Let's consider two n-dimensional vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ whose similarity can be depicted by the PCC distance. Next, according to LSH theory [11], we can transform vectors X and Y containing private information into corresponding hash values with little privacy, i.e., $h(X)$ and $h(Y)$, respectively. Concretely, $h(X)$ can be calculated by (1), where V is an n-dimensional vector $(v_1, \ldots, v_n)$ and v_j ($j = 1, 2, \ldots n$) is randomly selected from $[-1, 1]$; "$\circ$" represents the inner product of different vectors. The physical meaning of equation in (1) is: vector V is a hyper plane which divides the n-dimensional space into two parts; if point X is above hyper plane V (i.e., $X \bigcirc V > 0$), then $h(X) = 1$ with high probability; otherwise, $h(X) = 0$:

$$h(X) = \begin{cases} 1 & \text{if } X \circ V > 0 \\ 0 & \text{if } X \circ V \leq 0 \end{cases} \tag{1}$$

Thus, through the hash map in (1), n-dimensional vectors X and Y are transformed into two Boolean values, i.e., $h(X)$ and $h(Y)$, respectively. However, LSH is essentially a probability-based similar friend search technique; therefore, a single hash value $h(X)$ or $h(Y)$ cannot precisely represents the original n-dimensional vector X or Y. Considering this, more hash functions, i.e., a hash function family $H(.) = \{h_1(.), \ldots, h_r(.)\}$ ($r << n$) are adopted here. Through the hash function family $H(.)$, we can transform the n-dimensional vectors X and Y into r-dimensional 0–1 vectors, i.e., $H(X) = \{h_1(X), \ldots, h_r(X)\}$ and $H(Y) = \{h_1(Y), \ldots, h_r(Y)\}$, respectively. The vectors X and Y, as well as their respective hash values $H(X)$ and $H(Y)$, form a hash table. We repeat the above hash table building process until L hash tables, i.e., $Tb_1, \ldots, Tb_L$ are obtained. Next, according to LSH theory, vectors X and Y are contrary with large probability iff the condition in (2) holds. In (2), $H_z(X)$ and $H_z(Y)$ denote the hash values of vectors X and Y in z-th hash table, respectively; symbol "$\oplus$" represents the XOR operation:

$$\exists\, z, \text{ s.t. } H_z(X) \oplus H_z(Y) = (1, 1, \ldots, 1) \ (z \in \{1, \ldots, L\}) \tag{2}$$

The physical meaning of equation in (2) is clarified as below: if points X and Y are always located on the different sides of hyper plane V (i.e., $h_i(X) \neq h_i(Y)$ holds for all $i \in \{1, \ldots, r\}$), then X and Y are far away from each other with large probability (i.e., $H(X) \oplus H(Y) = (1, 1, \ldots, 1)$). Furthermore, if $H(X) \oplus H(Y) = (1, 1, \ldots, 1)$ in any of $Tb_1, \ldots, Tb_L$, points X and Y can be regarded as two contrary points (i.e., enemies). This is the main idea of our proposed converse LSH technique. Through converse LSH, we can search for the users (denoted by "enemies") whose preferences are totally different from the target user, in an efficient and privacy-preserving manner, as elaborated in the next section.

4. An Exception Handling Approach Based on Converse LSH

Next, we introduce an approach for handling the exceptions incurred by service recommendation failures, named SerRec$_{\text{converse-LSH}}$, based on the converse LSH technique introduced in Section 3. Concretely, our approach consists of three steps.

- Step-1: Build user indices offline through traditional LSH technique.

Let's consider a user u whose single hash value (denoted by $h(u)$) is based on the hash map in Equation (1) and user u's historical service quality data (assume fixed and real values) monitored by sensors. Furthermore, according to the hash function family $H(.) = \{h_1(.), \ldots, h_r(.)\}$ ($r << n$) in Section 3, we can obtain user u's compound hash value $H(u) = \{h_1(u), \ldots, h_r(u)\}$. Then $H(u)$ is treated as user u's index. Moreover, all users as well as their respective indices form a hash table. Repeat the above process until L hash tables, i.e., $Tb_1, \ldots, Tb_L$ are obtained. The above user indices building process can be executed offline before a service recommendation requirement is raised; therefore, its time complexity is O(1), which indicates that the recommendation speed can be accelerated greatly.

- Step-2: Determine the indirect friends of the target user u^* based on user indices and converse LSH technique.

We have obtained the user indices $H(u)$ (including the index $H(u^*)$ for the target user (denoted by u^*)), and form L hash tables $Tb_1, \ldots, Tb_L$. Next, if $H(u) \oplus H(u^*) = (1, 1, \ldots, 1)$ holds in any $Tb_1, \ldots, Tb_L$, then user u can be regarded as a qualified enemy of the target user u^* based on the converse LSH theory introduced in Section 3. Likewise, for each user ω ($\omega \neq u$ and $\omega \neq u^*$), if $H(\omega) \oplus H(u) = (1, 1, \ldots, 1)$ holds in any hash table, then user ω can be regarded as a qualified enemy of user u. Thus, ω can be considered as an indirect friend of u^* based on the "enemy's enemy is a possible friend" rule in Social Balance Theory; afterwards, we put ω into a new user set Possible_Friend_set(u*). Repeat the above process until all the indirect friends of u^* are found. This way, we can derive the friends of u^* in an indirect manner, if u^* does not have similar friends due to the data sparsity according to the traditional LSH technique.

Next, we turn to the example in Figure 2 where three users $\{u_1, u_2, u_3\}$ and two hash tables $\{Tb_1, Tb_2\}$ are present. The index values of the three users are also shown in Figure 2. Then according to the judgement condition in Equation (2), u_2 is an enemy of u_1 as $(1, 0, 1, 0) \oplus (0, 1, 0, 1) = (1, 1, 1, 1)$ holds in hash table Tb_1. Similarly, u_3 is an enemy of u_2 as $(1, 1, 0, 1) \oplus (0, 0, 1, 0) = (1, 1, 1, 1)$ holds in hash table Tb_2. With the above analyses, we can infer that u_3 is a possible friend of u_1 based on the "enemy's enemy is a possible friend" rule. So u_3 is put into the friend set of u_1, i.e., Possible_Friend_set(u$_1$).

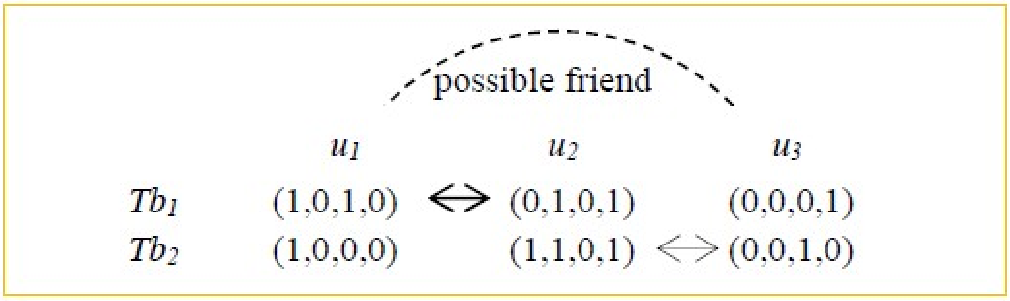

Figure 2. Indirect friend finding based on converse LSH: an example.

- Step-3: Recommend services to u^* based on the possible friends of u^*.

According to target user u^*'s possible friends (Possible_Friend_set(u*)) derived in Step-2, the users in, we can recommend appropriate services to u^*. First of all, we predict the missing QoS data of ws

by u* by (3) where q is a QoS criterion, $q(u$*, $ws)$ represents ws's QoS over criterion q invoked by u*; Φ denotes the set of users who are possible friends of u* and have invoked service ws before, which can be obtained by (4). Thus we can rank services ws based on $q(u$*, $ws)$ in (3). At last, optimal services are recommended to u*; this way, the recommendation failures are overcome:

$$q(u^*, ws) = \frac{1}{|\varphi|}^{*} \sum_{u_i \in \varphi} q(u_i, ws) \tag{3}$$

$$\Phi = \{u_i \mid u_i \in \text{Possible\_Friend\_set}(\text{u}^*) \text{ and } u_i \text{ has ever invoked } ws\} \tag{4}$$

5. Experiments

5.1. Experiment Configurations

To prove the effectiveness of our suggested exception handling approach named SerRec$_{\text{converse-LSH}}$, we design and deploy several experiments based on popular dataset Movielens [25]. Movielens reports the rating data of 3900 movies rated by 6040 users all over the world. Different from other applications where multiple dimensions are present [26–34], we consider only one quality dimension (i.e., user rating) in the experiments and take it as the unique recommendation basis, because the Movielens dataset only provides one dimension, i.e., user-service rating. The complex multi-dimensional service recommendation scenarios are out of the scope of this work. (In the multi-dimensional service recommendation scenarios, the multiple quality dimensions as well as their mutual correlations, such as the linear correlations and non-linear correlations should all be taken into consideration. In this situation, the problem becomes more complex and cannot be directly extended from this work, so we will investigate the complex multi-dimensional service recommendation problems in the future; see Section 5.3 "Shortcoming analyses & future work"). We randomly remove partial entries of the dataset to simulate recommendation failure scenarios. To evaluate the performance of exception handling approaches, we test the time cost and Mean Absolute Error (MAE), respectively (Note that LSH can protect data inherently, therefore, the privacy protection effect of SerRec$_{\text{converse-LSH}}$ is not tested in the experiment). Moreover, to validate the feasibility of our proposed SerRec$_{\text{converse-LSH}}$ approach, we compare our proposal with the following three competitive handling approaches.

(1) Random: this benchmark approach predicts the missing service quality data based on the quality of a randomly selected service, and returns the service with the optimal predicted quality.

(2) WSRec [23]: it predicts the missing service quality data by two pieces of average quality, i.e., average quality of the service rated by all users and average quality of all services rated by the user. Finally, the optimal service is returned to the target user.

(3) SBT-SR [12]: this approach first looks for the indirect friends of a target user based on Collaborative Filtering and Social Balance Theory, and then recommends appropriate services based on the derived indirect friends.

The experiment running environment is as follows: (1) hardware: 2.80 GHz CPU + 2.0 GB RAM; (2) software: Windows XP + JAVA 1.5. Experiments are executed ten times and their average values are reported.

5.2. Experiment Results

Four experiments are designed and deployed, respectively. Four parameters are present in the experiments: m, n denote the sizes of user set and service set; L, r represent the sizes of hash table set and hash function set.

- Profile 1: Accuracy comparison of four approaches

The accuracy values of outputted results of four exception handling approaches are compared. Here, m = 6000, n = 3900, L = 10, r = 8. Experiments are repeated ten times. The concrete experiment results of the ten iterations and the average result are demonstrated in Figure 3, which shows that the accuracy value of Random approach is the smallest, as a random strategy is adopted to predict the missing quality data. A simple and naïve "average" strategy is recruited in WSRec approach to predict the missing service quality, while the average service quality cannot reflect the real running quality of services very well; therefore, the recommendation accuracy of WSRec approach is also low. Both $\text{SerRec}_{\text{converse-LSH}}$ and SBT-SR approaches utilize the Social Balance Theory to improve the recommendation robustness; however, the accuracy value of our suggested $\text{SerRec}_{\text{converse-LSH}}$ approach outperforms those of the other three competitive approaches including SBT-SR, as only the similar friends (obtained in an indirect manner) of a target user are taken into consideration in missing QoS prediction in $\text{SerRec}_{\text{converse-LSH}}$. Another observation from Figure 3 is that the recommendation accuracy value of $\text{SerRec}_{\text{converse-LSH}}$ approach does not vary significantly and regularly, which means that our proposal can make relatively stable service recommendations.

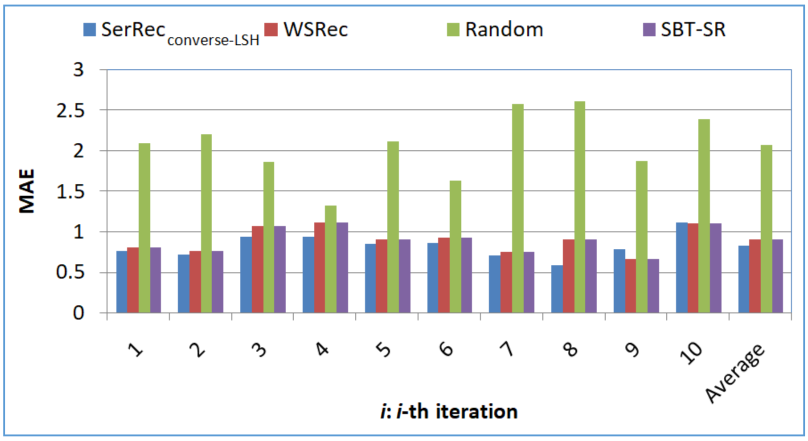

Figure 3. Accuracy of recommended results.

- Profile 2: Efficiency comparison of four approaches

We measure the time cost for generating recommended results in our suggested $\text{SerRec}_{\text{converse-LSH}}$ approach and compare it with the rest three approaches. The parameters are as follows: $m = 6000$, $n = 3900$, $L = 10$, $r = 8$. Experiments are repeated ten times. The concrete experiment results of the ten iterations and the average result are demonstrated in Figure 4.

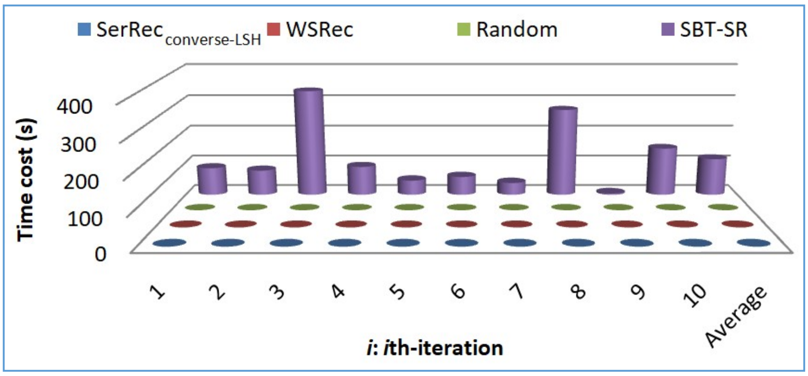

Figure 4. Recommendation efficiency comparison.

Figure 4 shows that the recommendation efficiency of SBT-SR is low as it is based on Collaborative Filtering and hence the time cost is rather high. The service recommendation efficiencies of the other three approaches, i.e., $\text{SerRec}_{\text{converse-LSH}}$, WSRec and Random are rather high and approximately the same. This is because most tasks in our SerRec$_{\text{converse-LSH}}$ approach can be done offline, and both WSRec and Random approaches have a polynomial time complexity.

- Profile 3: Accuracy of SerRec$_{\text{converse-LSH}}$ with respect to L and r

Next, we test the variation tendency of accuracy of the proposed SerRec$_{\text{converse-LSH}}$ approach with respect to the parameters L and r. Here, m = 6000, n = 3900, L and r are both varied from 6 to 10. Experiments are repeated ten times. The average values are demonstrated in Figure 5.

According to LSH theory, a larger r value or a smaller L value implies tighter condition for neighbor search and higher recommendation accuracy (i.e., lower MAE value). However, as Figure 5 indicates, the recommendation accuracy of SerRec$_{\text{converse-LSH}}$ does not render an obvious fluctuation tendency with L and r. This is due to the following reason: in our proposal, the traditional LSH technique is modified to be the converse LSH technique; and the converse LSH technique is recruited twice in order to search for the friends of a target user indirectly. So the influence of parameters L and r over the recommendation accuracy is not so obvious any more.

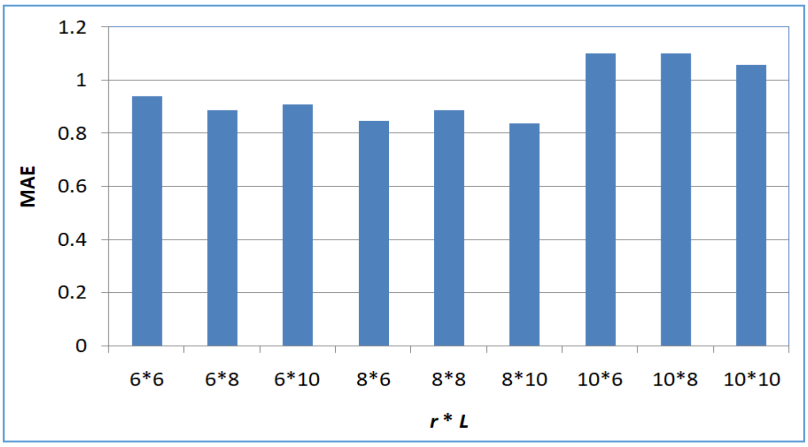

Figure 5. Accuracy of SerRec$_{\text{converse-LSH}}$.

- Profile 4: Efficiency of SerRec$_{\text{converse-LSH}}$ with respect to L and r

Next, we evaluate the efficiency of SerRec$_{\text{converse-LSH}}$ with respect to L and r. Here, m = 6000; n = 3900; L = 6, 8, 10; r = 6, 8, 10. Experiments are repeated ten times. The average results are demonstrated in Figure 6.

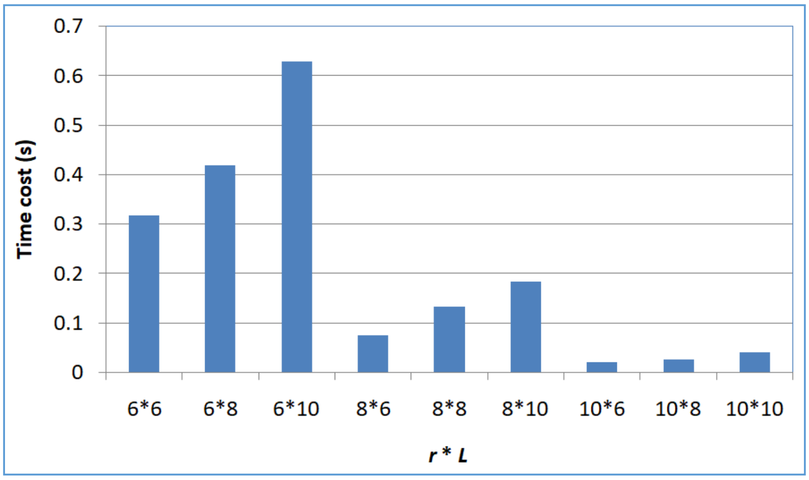

Figure 6. Efficiency of SerRec$_{\text{converse-LSH}}$.

From the figure, we can see that our efficiency decreases when the number of hash tables, i.e., L rises. This is because when L grows, the search condition for dissimilar enemy becomes looser and correspondingly, more qualified enemies are returned to take part in the service recommendation decision-makings; in this situation, more time cost is needed. Another result that Figure 6 indicates is that our efficiency decreases when the number of hash functions, i.e., r drops. This is because when r drops, the search condition for dissimilar enemy becomes stricter and correspondingly, fewer qualified enemies of a target user are returned to take part in the service recommendation decision-makings; therefore, less computational time is needed.

With the above analyses, a conclusion can be drawn that $\text{SerRec}_{\text{converse-LSH}}$ approach achieves a good tradeoff between service recommendation accuracy and efficiency. Besides, $\text{SerRec}_{\text{converse-LSH}}$ outperforms the other approaches in terms of privacy-preservation due to the inherent characteristic of LSH.

5.3. Shortcoming Analyses & Future Work

There are still several shortcomings in our approach. First of all, we only consider the recommendation scenario where one quality dimension is monitored by sensors, while multi-dimensional and weighted applications are more common in practice [35–37], so in the future, we will further refine our work by considering the multiple service quality dimensions as well as their respective weights. Besides, for simplicity, we only discuss the service quality dimensions with real and continuous monitored values, without considering the diversity of the quality values (e.g., discrete [38,39], binary [40], fuzzy [41] and correlated [42–44]). Considering this drawback, we will further improve our proposed recommendation approach by integrating the diverse forms (or formats) of different service quality dimensions, for the purpose of getting more comprehensive and reasonable recommended services.

6. Conclusions

The multi-source property of service usage data (monitored by distributed sensors) used to make service recommendations in the cloud environment requires that a recommender system to quickly integrate the distributed monitored data so as to make comprehensive and accurate recommendation decisions. In this situation, protecting the private information of users from leakage during the above data integration process is an important but challenging task for the successful service recommendation. Although the LSH technique can be recruited to achieve the abovementioned data integration and privacy-preservation goals, existing LSH-based service recommendation approaches seldom consider the possible recommendation failures as well as the resulted exceptions. In view of this drawback, we put forward a new LSH variant, i.e., converse LSH, and integrate it with the Social Balance Theory so as to look for the possible friends of a target user indirectly and then recommend appropriate services based on the obtained possible friends. The experiments conducted on Movielens dataset prove the effectiveness of our approach in terms of service recommendation accuracy and time cost while guaranteeing privacy-preservation of quality data monitored by sensors.

However, in $\text{SerRec}_{\text{converse-LSH}}$, only one quality dimension of services is considered. In our future work, we will continue to refine $\text{SerRec}_{\text{converse-LSH}}$ by considering multiple quality dimensions as well as their weight information. Besides, QoS data often vary with concrete service execution context (e.g., service invocation time and user location); therefore, we will further improve our approach by taking context into consideration.

Author Contributions: Data curation, W.D.; Formal analysis, S.M., X.Z. and Z.Z.; Methodology, R.W.; Validation, X.X.; Writing—original draft, L.Q.

Funding: This paper is partially supported by the Natural Science Foundation of China (No. 61672276, No. 61702277, No. 61702299), the Open Research Project of State Key Laboratory of Novel Software Technology (Nanjing University) under Grant No. KFKT2017B07, the New Zealand Marsden Funds, the UoA Faculty Research

Development Fund (No. 3714668) and the Open Project of State Key Laboratory for Novel Software Technology (No. KFKT2016B22).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Naim, H.; Aznag, M.; Quafafou, M.; Durand, N. Probabilistic approach for diversifying web services discovery and composition. In Proceedings of the International Conference on Web Services (ICWS), San Francisco, CA, USA, 27 June–2 July 2016; pp. 73–80.
2. Qi, L.; Zhang, X.; Dou, W.; Ni, Q. A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2616–2624. [CrossRef]
3. Cui, J.; Zhang, Y.; Cai, Z.; Liu, A.; Li, Y. Securing display path for security-sensitive applications on mobile devices. *Comput. Mater. Contin.* **2018**, *55*, 17–35.
4. Cao, Y.; Zhou, Z.; Sun, X.; Gao, C. Coverless information hiding based on the molecular structure images of material. *Comput. Mater. Contin.* **2018**, *54*, 197–207.
5. Liu, Y.; Peng, H.; Wang, J. Verifiable diversity ranking search over encrypted outsourced data. *Comput. Mater. Contin.* **2018**, *55*, 37–57.
6. Li, T.; Li, J.; Liu, Z.; Li, P.; Jia, C. Differentially Private Naive Bayes Learning over Multiple Data Sources. *Inf. Sci.* **2018**, *444*, 89–104. [CrossRef]
7. Meng, W.; Tischhauser, E.; Wang, Q.; Wang, Y.; Han, J. When Intrusion Detection Meets Blockchain Technology: A Review. *IEEE Access* **2018**, *6*, 10179–10188. [CrossRef]
8. Zhang, Y.; Chen, X.; Li, J.; Wong, D.S.; Li, H.; You, I. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Inf. Sci.* **2017**, *379*, 42–61. [CrossRef]
9. Cai, Z.; Yan, H.; Li, P.; Huang, Z.; Gao, C. Towards secure and flexible EHR sharing in mobile health cloud under static assumptions. *Cluster Comput.* **2017**, *20*, 2415–2422. [CrossRef]
10. Li, P.; Li, T.; Ye, H.; Li, J.; Chen, X.; Xiang, Y. Privacy-preserving machine learning with multiple data providers. *Future Gener. Comput. Syst.* **2018**, *87*, 341–350. [CrossRef]
11. Gionis, A.; Indyk, P.; Motwani, R. Similarity search in high dimensions via hashing. *VLDB J.* **1999**, *99*, 518–529.
12. Qi, L.; Zhang, X.; Wen, Y.; Zhou, Y. A social balance theory-based service recommendation approach. In Proceedings of the International Conference on Asia-Pacific Services Computing (APSCC), Bangkok, Thailand, 7–9 December 2015; pp. 48–60.
13. Qi, L.; Zhou, Z.; Yu, J.; Liu, Q. Data-sparsity tolerant web service recommendation approach based on improved collaborative filtering. *IEICE T. Inf. Syst.* **2017**, *E100D*, 2092–2099. [CrossRef]
14. Zheng, X.; Cai, Z.; Li, J.; Gao, H. Location-privacy-aware review publication mechanism for local business service systems. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
15. Fran, C.; Josep, D.F.; Constantinos, P.; Domènec, P.; Agusti, S. A k-anonymous approach to privacy preserving collaborative filtering. *J. Comput. Syst. Sci.* **2015**, *81*, 1000–1011.
16. Ahila, S.S.; Shunmuganathan, K.L. Role of agent technology in web usage mining: homomorphic encryption based recommendation for e-commerce applications. *Wirel. Pers. Commun.* **2016**, *87*, 499–512. [CrossRef]
17. Zhu, J.; He, P.; Zheng, Z.; Lyu, M.R. A privacy-preserving qos prediction framework for web service recommendation. In Proceedings of the International Conference on Web Services (ICWS), New York, NY, USA, 27 June–2 July 2015; pp. 241–248.
18. Dou, K.; Guo, B.; Kuang, L. A privacy-preserving multimedia recommendation in the context of social network based on weighted noise injection. *Multimed. Tools Appl.* **2017**, 1–20. [CrossRef]
19. Xu, Y.; Qi, L.; Dou, W.; Yu, J. Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment. *Complexity* **2017**, *2017*, 3437854. [CrossRef]
20. Qi, L.; Zhang, X.; Dou, W.; Hu, C.; Yang, C.; Chen, J. A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. *Future Gener. Comput. Syst.* **2018**. [CrossRef]
21. Gong, W.; Qi, L.; Xu, Y. Privacy-aware multi-dimensional mobile service quality prediction and recommendation in distributed fog environment. *Wirel. Commun. Mob. Commun.* **2018**, *2018*, 3075849.

22. Zhang, K.; Fan, S.; Wang, H.J. An efficient recommender system using locality sensitive hashing. In Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS), Hawaii, HI, USA, 3–6 January 2018.
23. Zheng, Z.; Ma, H.; Lyu, M.R.; King, I. QoS-aware web service recommendation by collaborative filtering. *IEEE Trans. Serv. Comput.* **2011**, *4*, 140–152. [CrossRef]
24. Qi, L.; Dou, W.; Zhang, X. An inverse collaborative filtering approach for cold-start problem in web service recommendation. In Proceedings of the Australasian Computer Science Week (ACSW), Geelong, Australia, 31 January–3 February 2017; pp. 46–54.
25. Movielens. Available online: https://grouplens.org/datasets/movielens/ (accessed on 11 March 2018).
26. Tian, G.; Wang, M.; Song, L. Variable selection in the high-dimensional continuous generalized linear model with current status data. *J. Appl. Stat.* **2014**, *41*, 467–483. [CrossRef]
27. Wang, M.; Tian, G. Robust group non-convex estimations for high-dimensional partially linear models. *J. Nonparametr. Stat.* **2016**, *28*, 49–67. [CrossRef]
28. Wang, X.; Wang, M. Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure. *J. Appl. Stat.* **2016**, *43*, 796–809. [CrossRef]
29. Wang, P.; Zhao, L. Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds. *Nonlinear Anal.* **2016**, *130*, 1–17. [CrossRef]
30. Wang, P.; Wang, X. The geometric properties of harmonic function on 2-dimensional Riemannian manifolds. *Nonlinear Anal.* **2014**, *103*, 2–8. [CrossRef]
31. Wang, M.; Song, L.; Tian, G. SCAD-penalized least absolute deviation regression in high dimensional models. *Commun. Stat.-Theory Methods* **2015**, *44*, 2452–2472. [CrossRef]
32. Xu, F.; Zhang, X.; Wu, Y.; Liu, L. Global existence and the optimal decay rates for the three dimensional compressible nematic liquid crystal flow. *Acta Appl. Math.* **2017**, *150*, 67–80. [CrossRef]
33. Wang, X.; Zhao, S.; Wang, M. Restricted profile estimation for partially linear models with large-dimensional covariates. *Stat. Probabil. Lett.* **2017**, *128*, 71–76. [CrossRef]
34. Tian, H.; Han, M. Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems. *J. Differ. Equ.* **2017**, *263*, 7448–7474. [CrossRef]
35. Yang, S.; Yao, Z.; Zhao, C. The weight distributions of two classes of p-ary cyclic codes with few weights. *Finite Fields Their Appl.* **2017**, *44*, 76–91. [CrossRef]
36. Wang, Y.; Yin, C.; Zhang, X. Uniform estimate for the tail probabilities of randomly weighted sums. *Acta Math. Appl. Sin. E* **2014**, *30*, 1063–1072. [CrossRef]
37. Cai, J. An implicit sigma(3) type condition for heavy cycles in weighted graphs. *Ars Combin.* **2014**, *115*, 211–218.
38. Liu, H.; Meng, F. Some new generalized volterra-fredholm type discrete fractional sum inequalities and their applications. *J. Inequal. Appl.* **2016**, *2016*, 213. [CrossRef]
39. Li, P.R.; Ren, G.B. Some classes of equations of discrete type with harmonic singular operator and convolution. *Appl. Math. Comput.* **2016**, *284*, 185–194. [CrossRef]
40. Zhang, B. Remarks on the maximum gap in binary cyclotomic polynomials. *Bull. Math. Soc. Sci. Math.* **2016**, *59*, 109–115.
41. Wang, L. The fixed point method for intuitionistic fuzzy stability of a quadratic functional equation. *Fixed Point Theory A* **2010**, 107182. [CrossRef]
42. Liu, L.L.; Ma, D. Some polynomials related to dowling lattices and x-stieltjes moment sequences. *Linear Algebra Appl.* **2017**, *533*, 195–209. [CrossRef]
43. Li, L.; Meng, F.; Zheng, Z. Oscillation results related to integral average technique for linear hamiltonian systems. *Dyn. Syst. Appl.* **2009**, *18*, 725–736.
44. Xu, A.; Ding, N. Semidualizing bimodules and related gorenstein homological dimensions. *J. Algebra Appl.* **2016**, *15*, 1650193. [CrossRef]