A Locality Sensitive Hashing Based Approach for Federated Recommender System

Hongsheng Hu*
School of Electrical and Computer Engineering
The University of Auckland
Auckland, NZ, 1023.
Email: hhu603@aucklanduni.ac.nz

Gillian Dobbie
School of Computer Science
The University of Auckland
Auckland, NZ, 1023.
Email: g.dobbie@auckland.ac.nz

Zoran Salcic
School of Electrical and Computer Engineering
The University of Auckland
Auckland, NZ, 1023.
Email: z.salcic@auckland.ac.nz

Meng Liu
School of Mechanical, Electrical and Information Engineering
Shandong University, Weihai
Shandong, CN, 264209.
Email: liumeng@sdu.edu.cn

Jianbing Zhang
School of Computer Science and Technology
Nanjing University
Nanjing, China, 210093.
Email: zjb@nju.edu.cn

Xuyun Zhang*
School of Science and Engineering
Macquarie University
Sydney, AU, 2109.
Email: xuyun.zhang@mq.edu.au

***Abstract*—The recommender system is an important application in big data analytics because accurate recommendation items or high-valued suggestions can bring high profit to both commercial companies and customers. To make precise recommendations, a recommender system often needs large and fine-grained data for training. In the current big data era, data often exist in the form of isolated islands, and it is difficult to integrate the data scattered due to privacy security concerns. Moreover, privacy laws and regulations make it harder to share data. Therefore, designing a privacy-preserving recommender system is of paramount importance. Existing privacy-preserving recommender system models mainly adapt cryptography approaches to achieve privacy preservation. However, cryptography approaches have heavy overhead when performing encryption and decryption operations and they lack a good level of flexibility. In this paper, we propose a Locality Sensitive Hashing (LSH) based approach for federated recommender system. Our proposed efficient and scalable federated recommender system can make full use of multiple source data from different data owners while guaranteeing preservation of privacy of contributing parties. Extensive experiments on real-world benchmark datasets show that our approach can achieve both high time efficiency and accuracy under small privacy budgets.**

***Keywords*-recommender system; locality sensitive hashing; differential privacy**

I. Introduction

Recommender systems aim to provide suggestions for users on how to select items among a large number of options. It is an important and practical data mining technique which is widely studied by both academia and industry, e.g., *Netflix* holds an open competition for best predicting films for its users [1]. A wide range of models has been proposed based on different methods of generating recommendation items [2]. Among these methods, filtering technique plays an important role. According to the characteristics of filtering, filtering algorithms can be divided into collaborative filtering, demographic filtering, content-based filtering, and hybrid filtering. One can refer to [2] for more details.

In the current big data era, data sets with rich and fine-grained information exist in the form of isolated islands and pose a considerable challenge on traditional data mining and analytics domains including recommender systems [3]. To achieve accurate recommendations, a recommendation algorithm often involves multiple source data for training. However, it is usually the case that multiple source data collected by different owners exist at different locations. For example, in a product recommendation service, the electric product on-line dealer Jingdong company has the data of users' purchase of electric product but the data of users' purchase of daily supplies can exist in other daily supplies on-line dealers such as Taobao and Tmall. However, due to privacy security concerns and complicated administrative procedures, it is difficult to integrate the data scattered at different companies. Especially, data security and privacy raise more and more concerns with the increasing incidents and scandals of compromising user privacy. For instance, personal data of millions of Facebook users were harvested from their profiles by Cambridge Analytica in 2018 without the users' consent and were further exploited for a political advertising purpose [4]. Hence, how to design privacy-preserving recommender system algorithms for utilizing different sources of data is of paramount importance. Besides, the volumes of the training data often become increasingly massive with continuous updates over time. As such, scalability and efficiency are important for the success of the recommender system in big data applications.

Given these challenges, the collaborative filtering based privacy-preserving recommender system is believed to be

978-1-7281-6095-5/20/$31.00 ©2020 IEEE
DOI 10.1109/CCGrid49817.2020.000-1

a promising way as they can recommend efficiently and accurately while achieving privacy preservation. The core idea of collaborative filtering based methods is to make recommendations to each user based on information provided by those users who have most in common with target user. However, traditional collaborative filtering methods such as those using user Pearson correlation coefficient need to train the recommender models at the raw data level, which poses privacy threats to users' data. Existing collaborative filtering based privacy-preserving recommender systems mainly focus on combining cryptography approaches with collaborative filtering to achieve privacy preservation. For example, a privacy-preserving item-based collaborative filtering method that uses an unsynchronized secure multiparty computation protocol is proposed to protect user privacy while achieving high recommendation accuracy [5]. Besides the heavy overheads of performing encryption and decryption operations, these methods, however, fail to offer a good level of flexibility for privacy-preserving data publishing and mining applications, because a recipient will know everything (the key is known) or nothing (the key is unknown) about the encrypted data. A better alternative is to make use of differential privacy to control the trade-off between data utility and privacy, since differential privacy has a solid theoretical foundation and strong privacy protection capability [6]. A differentially private recommender system framework is proposed by McSherry et al. in [7]. However, this framework does not show how to handle multiple source data. Qi et al. [8] proposed an LSH based recommender system to achieve fast and accurate privacy-preserving recommendation service while making use of multiple source data, but they did not investigate the privacy risks of their method and their method lack formal privacy guarantee.

In this paper, we propose a differentially private LSH based approach which can make use of data from different sources for fast and accurate recommendation while providing differential privacy guarantee for users. The main contributions of our proposed approach are threefold. First, we propose a generic collaborative filtering framework with differentially private LSH for scalable, efficient, and private recommender system. Compared to traditional LSH based privacy-preserving recommender system, our proposed method retains high recommendation accuracy while providing formal privacy guarantee. Second, our proposed method can make full use of different sources of data while eliminating data exposure risks. Finally, comprehensive experiments validate the effectiveness and efficiency of our approach.

The remainder of this paper is organized as follows. Section II reviews the related research work. Then, we show a motivating example and state the research problem in Section III. Section IV introduces the LSH user-based privacy-preserving recommender system and point out the potential privacy risks. Our generic framework is formulated in Section V, followed by theoretical analysis in Section VI. Extensive empirical evaluation on benchmark datasets is described in Section VII. We conclude this paper and discuss future work in the end.

II. Related Work

Privacy incidents and scandals give rise to the high demand for privacy preservation when performing analytics and mining on privacy-sensitive data, and an increasing number of laws and regulations have been established to avoid them. For example, the European Union has recently published the General Data Protection Regulation (GDPR) to give control to individuals over their personal data [9]. As a classical application of machine learning, recommender system collects and uses as much user data as possible to build accurate recommendation. However, this clearly has a negative impact on the privacy of users since they may feel that the system knows too much about their true preferences and the data collected by the systems may be revealed. Thus, designing privacy-preserving recommender systems is of paramount importance.

Kaur et al. propose a privacy-preserving collaborative filtering scheme on arbitrary distributed data based on multiparty random masking and polynomial aggregation technique [10]. In their scheme, privacy preservation is achieved by adapting protocols and paillier homomorphic encryption. Similarly, Badsha et al. propose a privacy-preserving user-based collaborative filtering technique based on homomorphic encryption [11]. The technique can calculate similarities among users and generate recommendations without revealing any private information. Polatidis et al. propose a multi-level privacy-preserving method for collaborative filtering systems by perturbing each rating before it is submitted to the server [12]. The perturbation method is based on multiple levels and different ranges of random values for each level. However, the random perturbations method lacks formulated privacy definition and it is hard to quantify the privacy level.

Compared to encryption privacy models, differential privacy has light computation overhead and has a solid theoretical foundation of privacy preservation. Dwork et al. firstly introduce and define the differential privacy [6] and since then it has been used in many domains. During the last decade, differential privacy attracts interest and its central notion spans a range of research areas, from the privacy community to areas of data science such as machine learning, data mining, statistics, and learning theory [13], [14], [15]. Based on differential privacy, McSherry et al. propose a general recommender system scheme which retains high recommendation performance while providing differential privacy guarantee for users [7]. The scheme uses data pre-processing after which data can be used to train many recommender systems algorithms.

As an effective and scalable similarity computation scheme, Locality Sensitive Hashing (LSH) has been successfully used in many applications such as near-duplicate detection, hierarchical clustering, genome-wide association study, and image similarity identification [16]. This is because LSH has a salient feature that it has linear time complexity for similarity computation, while traditional pair-wise computation has quadratic complexity. Based on these good properties, Liang et al. [17] propose real-time collaborative filtering recommender systems. They used locality-sensitive hashing to construct user or item blocks, which facilitate real-time neighborhood formation and recommendation making. Qi et al. [8] propose a distributed LSH based recommender system to achieve fast, accurate and high-quality recommendation service while preserving user privacy of different contributed parties. They state that through projection the original form of a user's data is represented by its index which hides the raw data. However, they do not discuss the potential issues of reconstructing original data and lack of theoretical analysis.

III. Motivating example and problem statement

Consider the example federated recommender system in Fig. 1.

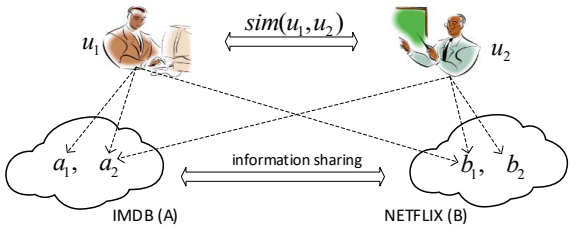


Figure 1: Motivation of federated recommender system.

Let A denote the IMDB company and B denote the NETFLIX company, where both companies recommend movies to users. a_1 and a_2 are the two movies in IMDB and b_1 and b_2 are the two movies in NETFLIX. Two users u_1 and u_2 are interested in movies $\{a_1, a_2, b_1\}$ and $\{b_1, b_2, a_2\}$ respectively. Now, driven by financial motives, IMDB and NETFLIX agree to share their respective user's movie interests information with each other to attract more users. According to the traditional user-based collaborative filtering recommendation method, if IMDB wants to recommend other movies to u_2, the first step is to calculate the similarity between u_1 and u_2, i.e., $sim(u_1, u_2)$. The calculation of $sim(u_1, u_2)$ is based on the movie preferences of u_1 and u_2, i.e., $\{a_1, a_2, b_1\}$ and $\{b_1, b_2, a_2\}$. However, due to privacy concerns, laws and regulations, NETFLIX can not share the raw data with IMDB, which makes the collaboration process infeasible and leads to inaccurate recommended results.

The example above can be extended to multiple companies. The goal of federated recommender systems is to provide high quality recommendations based on information from many companies without sharing the raw data while providing formal privacy guarantee for users. To achieve such a goal, we consider an LSH approach to implement data anonymization and combine differential privacy to provide formal privacy guarantees. We introduce the LSH user-based collaborative filtering recommender system (Rec_{LSH}) [8] which can achieve data anonymization while achieving the collaboration of multiple companies in the next section. Then, we describe our proposed federated LSH user-based collaborative filtering recommender system ($FRec_{LSH}$), which can not only provide data anonymization but also differential privacy guarantees.

IV. The fundamentals of Rec_{LSH}

The core idea of Rec_{LSH} is to use LSH as an index technique to achieve fast and efficient recommendations while providing data anonymization. We use Fig. 2 to show how the Rec_{LSH} provides data anonymization.

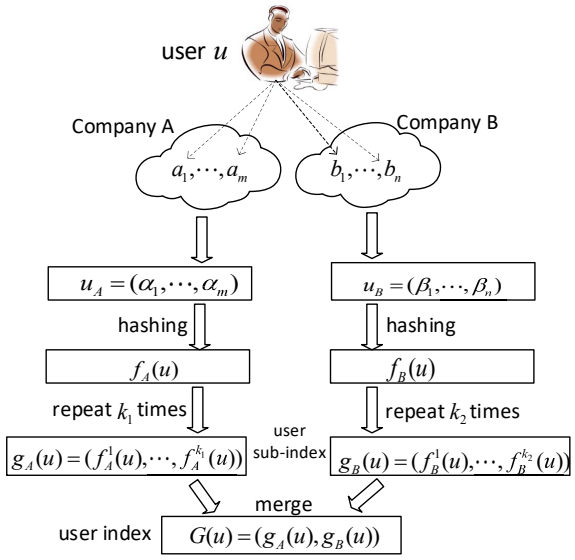


Figure 2: Building user index with collaboration.

Let u be a user who are interested in products $\{a_1, \cdots, a_m\}$ provided by company A and $\{b_1, \cdots, b_n\}$ provided by company B. The preference degree is quantified as $u_A = \{\alpha_1, \cdots, \alpha_m\}$ collected by A and $u_B = \{\beta_1, \cdots, \beta_n\}$ collected by B. The process of building a user index is as follows. Firstly, a LSH function is used by both company A and B to hash u_A and u_B to get $f_A(u)$ and $f_B(u)$. Then, A repeats the hash process k_1 times by using k_1 different LSH functions and B repeats the hash process k_2 times by using k_2 different LSH functions. A

gets the sub-index $g_A(u) = (f_A^1(u), \cdots, f_A^{k_1}(u))$ of u and B gets the sub-index $g_B(u) = (f_B^1(u), \cdots, f_B^{k_2}(u))$ of u. Both $g_A(u)$ and $g_B(u)$ can be shared with each other, thus, A and B can merge the sub-index of u to get the u's user index. Rec_{LSH} is based on such a users' index. A detailed description and implementation of Rec_{LSH} can be found in [8].

Data anonymization refers to hiding the identity and/or masking privacy-preserving data so that the privacy of an individual is preserved. Since u_A and u_B are represented by their index, company A and B can not know the original value of u_A and u_B, Thus, Rec_{LSH} provides data anonymization for both company A and company B. However, data anonymization techniques usually suffer from the de-anonymization issue. For example, Narayanan and Shmatikov successfully re-identified the individuals from an anonymized dataset provided by Netflix with the use of matching the data with film ratings in the Internet Movie database [18]. As a data anonymization technique, Rec_{LSH} also has privacy risks. An intuitive potential privacy risk is that a malicious attacker can reconstruct u_A or u_B by using the shared sub-index $g_A(u)$ and $g_B(u)$. Sensitive information about the user u can be exposed through the reconstructed results. Moreover, Rec_{LSH} does not provide formal privacy guarantees and it is hard to quantify the privacy level. Given the challenges above, we propose $FRec_{LSH}$ to retain data anonymization while providing differential privacy guarantees for users.

V. The fundamentals of $FRec_{LSH}$

$FRec_{LSH}$ has Rec_{LSH} at its foundation. However, the challenge of $FRec_{LSH}$ is to prove that $FRec_{LSH}$ guarantees particular levels of privacy, i.e., differential privacy. The aim of $FRec_{LSH}$ is to build a differential private LSH index. We propose a differentially private LSH (DPLSH) approach to build the differential private LSH index. The core idea of DPLSH is to apply perturbations in the process of building the hash index. We introduce what is local differential privacy and then show how DPLSH works.

Definition 1 (ϵ-local differential privacy): [19] A randomized function $f(\cdot)$ gives ϵ-local differential privacy if and only if for any two inputs t and t' in the domain of $f(\cdot)$, for any output t^* of $f(\cdot)$, we have

$$\Pr[f(t) = t^*] \leq \exp(\epsilon) \cdot \Pr[f(t') = t^*] \tag{1}$$

Where $Pr[\cdot]$ means probability. According to the above definition, an attacker, who receives the perturbed output t^*, can not distinguish whether the true value is t or another value t' with high confidence, where the confidence is controlled by the parameter ϵ, regardless of the background information of the attacker. Plausible deniability is provided to the users by local differential privacy [20].

Given a user's data $\boldsymbol{u}$, the company constructs the differentially private LSH index of $\boldsymbol{u}$ and shares it with other companies. The differentially private LSH index is a "noisy" representation of the user's data $\boldsymbol{u}$. The noisy LSH index of $\boldsymbol{u}$ is constructed in such a way so as to reveal a *controlled* amount of information about $\boldsymbol{u}$, limiting the malicious attacker's ability to learn with confidence what $\boldsymbol{u}$ was. To provide such a strong privacy guarantee, the process of constructing differentially private LSH index is implemented on a defense mechanism based on the idea of randomized response and and the level of privacy protection is controllable.

The DPLSH algorithm takes in the user's data $\boldsymbol{u}$ and parameters k, p, and executes locally by the company performing the following three steps:

Step 1**: Hashing.** Calculate the hash value of $\boldsymbol{u}$ using k locality sensitive hashing functions to get the hash signature $\mathcal{S}$.

Step 2**: Perturbation.** For each user's data $\boldsymbol{u}$ and bits $i, 0 \leq i < k$ in $\mathcal{S}$, create a differentially private (DP) hash signature $\mathcal{S}'$ whose corresponding value $\mathcal{S}_i$ follows the following probability condition

$$\begin{cases} \Pr(S_i' = 1 | S_i = 1) = p \\ \Pr(S_i' = 1 | S_i = 0) = 1 - p \end{cases} \tag{2}$$

where p is a tunable parameter controlling the level of privacy guarantee.

Step 3**: Report.** Send the generated hash signature $\mathcal{S}'$ to other companies.

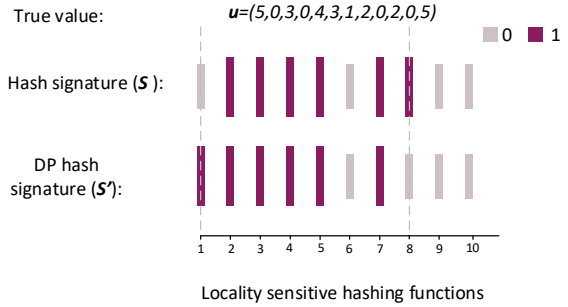


Figure 3: Process of constructing DPLSH signature: The user's data $\boldsymbol{u}$ is hashed to get its hash signature $\mathcal{S}$ using k (here $k = 10$) locality sensitive hashing functions. Then, a DP hash signature $\mathcal{S}'$ is generated locally by the company and can be shared with other companies.

The process of constructing the DPLSH index is intuitive and simple to understand. The Perturbation (step 2) replaces

the true locality sensitive hashing signature $\mathcal{S}$ with a randomized noisy version $\mathcal{S}'$. The reported hash signature $\mathcal{S}'$ may or may not contain the real locality sensitive hashing values of the user's data $\boldsymbol{u}$ depending on whether hashing values from $\mathcal{S}$ are replaced by random 1 or 0 with probability $1-p$. Privacy is guaranteed by the Perturbation (step 2) because the malicious attacker's ability to differentiate the true hash signature and the fake hash signature is limited. Therefore, a malicious attacker can not differentiate whether the inferred private information is true or not from the fake hash signature.

Figure 3 shows how the DPLSH algorithm works. Here, a user's data $\boldsymbol{u} = (5, 0, 3, 0, 4, 3, 1, 2, 0, 2, 0, 5)$ from a movie recommendation dataset indicates the ratings of 12 different movies. The ratings range from $[0, 5]$ and higher ratings mean higher preferences. The number of hash functions is 10 and the tunable parameter $p = 0.75$. The user's data $\boldsymbol{u}$ are hashed by 10 LSH functions to get the hash signature $\mathcal{S}$. Then, the hashing signature $\mathcal{S}$ of the first and eighth bits are flipped by Perturbation (Step 2). The DP hash signature that can be shared is shown at the bottom of the figure.

The difference between $FRec_{LSH}$ and Rec_{LSH} is the process of building hash index. $FRec_{LSH}$ provides the differential privacy guarantee that even if a malicious attacker can infer some private information of a user according to his/her locality sensitive hashing signature, the malicious attacker can not differentiate whether the inferred private information is true or not.

VI. Differential privacy of DPLSH

In this section, we give the theoretical analysis that DPLSH satisfies ϵ-differential privacy and show how to select parameters of DPLSH.

A. Proof of differential privacy

We show the DPLSH satisfies differential privacy with the help of observation 1.

Observation 1: [19] For $a, b \geq 0$ and $c, d > 0$, we have $\frac{a+b}{c+d} \leq \max\left(\frac{a}{c}, \frac{b}{d}\right)$.

Proof: Assume wlog that $\frac{a}{c} \geq \frac{b}{d}$, and suppose that the statement is false, i.e., $\frac{a+b}{c+d} > \frac{a}{c}$. Then $ac + bc > ac + ad$ or $bc > ad$, a contradiction with assumption that $\frac{a}{c} \geq \frac{b}{d}$. ■

Theorem 1: The Perturbation (Step 2 of DPLSH) satisfies ϵ-differential privacy where $\epsilon = k \ln\left(\frac{p}{1-p}\right)$.

Proof: Let $\mathcal{S}' = s'_1, \cdots, s'_m$ be randomized hash signatures generated by the company. The probability of observing any shared hash signature s' given the true user's value $\boldsymbol{u}$ is

$$\begin{aligned}\Pr\left(S' = s'|U = u\right) &= \Pr\left(S' = s'|s, u\right) \cdot \Pr(s|u) \\ &= \Pr\left(S' = s'|s\right)\end{aligned} \tag{3}$$

Without loss of generality, let the $1, \cdots, i$ hash values be set, i.e. $s = \{h_1 = 1, \cdots, h_i = 1\ , h_{i+1} = 0, \cdots, h_k = 0\}$. Then,

$$\begin{aligned}\Pr\left(s'|s\right) =& (1-p)^{1-h'_1}(p)^{h'_1} \times \cdots \times (1-p)^{1-h'_i}(p)^{h'_i} \times \cdots \\ & \times (p)^{1-h'_{i+1}}(1-p)^{h'_{i+1}} \times \cdots (p)^{1-h'_k}(1-p)^{h'_k}\end{aligned} \tag{4}$$

Let $\mathcal{S}^*$ be the set of all possible differentially private hash signatures. Let ϕ be the ratio of two such conditional probabilities with distinct values of $\mathcal{S}$, s_1, and s_2, i.e., $\phi = \frac{\Pr\left(S' \in S^*|S=s_1\right)}{\Pr\left(S' \in S^*|S=s_2\right)}$. To satisfy the differential privacy condition, ϕ should be bounded by $exp(\epsilon)$.

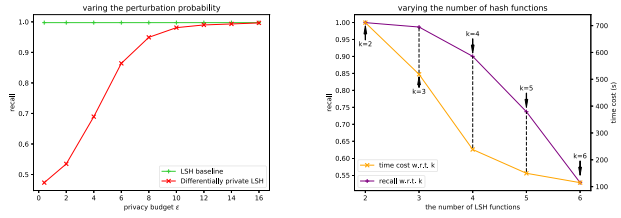
$$\begin{aligned}\phi =& \frac{\Pr\left(S' \in S^*|S = S_1\right)}{\Pr\left(S' \in S^*|S = S_2\right)} \\ =& \frac{\sum_{s' \in S^*} \Pr\left(S' = s'_j|S = s_1\right)}{\sum_{s' \in S^*} \Pr\left(S' = s'_j|S = s_2\right)} \\ \leq& \max_{s' \in S^*} \frac{\Pr\left(S' = s'_j|S = s_1\right)}{\Pr\left(S' = s'_j|S = s_2\right)} \text{ (by Observation 1)} \\ =& (p)^{k-2i+2\left(h'_1+h'_2+\cdots+h'_i-h'_{i+1}-h'_{i+2}-\cdots-h'_k\right)} \times \\ & (1-p)^{2i-k+2\left(h'_{i+1}+h'_{i+2}+\cdots+h'_k-h'_1-h'_2\cdots-h'_i\right)}\end{aligned} \tag{5}$$

Sensitivity is maximized when $h'_1 = h'_2 = \cdots = h'_i = 1$ and $h'_{i+1} = h'_{i+2} = \cdots = h'_k = 0$. Then, $\phi = \left(\frac{p}{1-p}\right)^k$ and $\epsilon = k \ln\left(\frac{p}{1-p}\right)$.

■

B. Parameter selection

The success of using LSH to implement similarity search is its linear time complexity, while traditional pair-wise computation requires quadratic complexity. In the original LSH, the number of hash functions k controls the trade-off between similarity search time and accuracy. Small k can ensure accuracy while the searching time is long. Larger k will decrese time, however, the searching accuracy will also decrease. A proper k should be fixed to balance searching time and accuracy. In our proposed DPLSH scheme, k also controls the trade-off between similarity search time and accuracy. Moreover, p affects accuracy since p controls the Pertubation (step 2) process. To implement the DPLSH algorithm, the parameters require to be tuned. We conduct a number of experiments (averaged over 10 replicates) to study how parameters affects similarity search. We ran two types of experiments to evaluate how p and k affect the similarity search capability of the DPLSH and compare the similarity search capability with the traditional LSH. We use the recall to measure the similarity search capability. The recall is the fraction of the M objects retrieved by a particular similarity search method among the actual M-nearest objects. We set $M = 100$ in our experiments. We use the *MNIST* dataset [21] to do our experiments for its convenience and spending minimal efforts on preprocessing and formatting. The *MNIST* dataset has a training set of



(a) recall w.r.t. ϵ (b) recall w.r.t. k

Figure 4: Recall versus privacy budget and the number of hash functions.

60000 handwritten digits and a test set of 10000 examples. Each image is converted to a 28×28 dimensional vector.

The first type of experiments is conducted to show how to choose the number of hash functions. The first type of experiments evaluate how the number of hash functions k affects recalls and running time. The first type of experiments evaluate how the number of hash functions k affects recalls and running time. In Fig. 4a, the left vertical axis stands for the recall of similarity search and the right vertical axis stands for the running time of similarity search. We can see from Fig. 4a when $k = 4$, the LSH has a recall of 0.90 (much higher than $k = 5$) with the running time 230 seconds (much smaller than $k = 3$). Thus, we recommend choosing $k = 4$ to achieve fast similarity search while guaranteeing a certain level of similarity search accuracy.

The second type of experiments evaluat how the privacy budget ϵ affects recall given the fixed number of hash functions k. According to the first type of experiments, we fixed k to 4 and set the value of p to be in the range $[0, 1)$. Fig. 4b shows the recall with regard to different ϵ values. We can see from Fig. 4b when the privacy budget ϵ is larger than 10, the similarity search capability of the DPLSH is nearly the same as that of LSH. When $\epsilon = 10$ and $k = 4$, the corresponding p is around 0.90. Thus, we recommend choosing $k = 4$ and $p = 0.90$ to achieve a good trade-off between privacy level and similarity search capability.

VII. Experimental Evaluation

A. Experiment Settings

To evaluate the effectiveness of $FRec_{LSH}$, we compare $FRec_{LSH}$ to Rec_{LSH} and one classic non-private approach, $UPCC$ [22]. The following three evaluation measures are examined and compared.

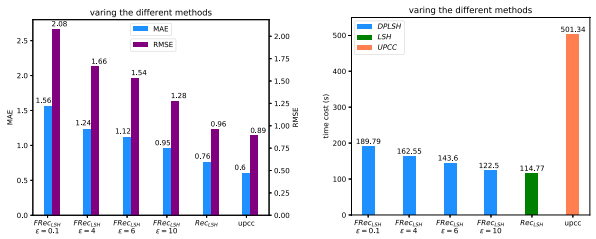
- ***Recommendation time:*** time consumed for generating recommendation results, through which we can test the recommendation efficiency and scalability.
- ***Mean Absolute Error (MAE):*** average difference between predicted quality and real quality of recommendations, through which we can test the recommendation accuracy.
- ***Root Mean Squared Error (RMSE):*** the square root of the average of the squared difference between predicted quality and real quality of recommendations, through which we can test the recommendation accuracy.

Rec_{LSH} provides data anonymization gurantee for each user, while $FRec_{LSH}$ provides not only data anonymization but also differential privacy guarantee. $UPCC$ is a non-private collabrative filtering recommender approach which based on access to the raw data. The MAE and RMSE of $UPCC$ are gold standard compared to Rec_{LSH} and $FRec_{LSH}$ since they oprate on private data. The MAE and RMS of Rec_{LSH} and $FRec_{LSH}$ should be larger than that of $UPCC$. We use one classic recommendation datasets which is *Jester* dataset from https://grouplens.org/datasets/jester/ to conduct the recommender system experiments. The Jester dataset contains 4.1 million continuous ratings $(-10.00, 10.00)$ of 100 jokes from $73,496$ users. We partition the Jester dataset into a test dataset and a training dataset according to the percentage of examples. We set the percentage of the test dataset to 0.1 and the percentage of the training dataset to 0.9. The experiments are conducted on a DELL OptiPlex 7050 computer with 3.60 GHz processors and 16.0 GB of RAM, running on Windows 10, Python 3.7.

B. Results and Analysis

Accuracy Comparison of the three Methods: Fig. 5a shows MAE and RMSE of $FRec_{LSH}$, Rec_{LSH}, and $UPCC$ methods respectively. In Fig. 5a, the left vertical axis is MAE value and the right vertical axis is the RMSE value. As shown in Fig. 5a, $UPCC$ method has the smallest MAE and RMSE compared to $FRec_{LSH}$ and Rec_{LSH}. This is because $UPCC$ method is a non-privacy-preserving method and it implements on the raw data. $FRec_{LSH}$ methods have the largest MAE and RMSE. However, the MAE and RMSE of $FRec_{LSH}$ method are getting smaller and smaller when the privacy budget ϵ increases and they are slightly larger than that of traditional Rec_{LSH} method when $\epsilon = 10$. This indicates that the recommendation accuracy of the $FRec_{LSH}$ method is comparable with the Rec_{LSH} method while providing differential privacy guarantees.

Efficiency Comparison of the three Methods: Fig. 5b shows the recommendation time of $FRec_{LSH}$, Rec_{LSH}, and $UPCC$ methods respectively. As shown in Fig. 5b, the time costs of both $FRec_{LSH}$ and Rec_{LSH} are far smaller the time cost of $UPCC$. This is because LSH based method has linear time complexity for similarity computation while $UPCC$ has quadratic complexity. The time cost of Rec_{LSH} is around $\frac{1}{5}$ of the time cost of $UPCC$. Moreover, the time cost of $FRec_{LSH}$ decreases when the privacy budget increases. The Perturbation (step 2) of the $DPLSH$ only marginally increases processing time compared to traditional LSH.



(a) MAE & RMSE w.r.t. different methods (b) time cost w.r.t. different methods

Figure 5: Recommendation accuracy and efficiency comparison.

VIII. Conclusions and Future Work

In this paper, we propose a Locality Sensitive Hashing (LSH) based method for federated recommender system. Specifically, we combine differential privacy and LSH and propose differentially private LSH to eliminate potential data exposure issues. Experiments have demonstrated that our differentially private LSH can achieve very close recommendation quality as traditional LSH while under a small differential privacy budget. Moreover, our proposed approach retains efficiency with marginally increased processing time compared to traditional LSH. Designing efficient and effective differential privacy algorithms for a given computational task is a challenging problem. Based on the contributions, we plan to integrate different LSH types with the aim of achieving scalable privacy-preserving recommender system.

References

[1] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, NY, USA., 2007, p. 35.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *KBS*, vol. 46, pp. 109–132, 2013.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *TIST*, vol. 10, no. 2, p. 12, 2019.

[4] C. Cadwalladr and E. Graham-Harrison, "The cambridge analytica files," *The Guardian*, vol. 21, pp. 6–7, 2018.

[5] D. Li, C. Chen, Q. Lv, L. Shang, Y. Zhao, T. Lu, and N. Gu, "An algorithm for efficient privacy-preserving item-based collaborative filtering," *FGCS*, vol. 55, pp. 311–320, 2016.

[6] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EuroCrypt*. Springer, 2006, pp. 486–503.

[7] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *SIGKDD*. ACM, 2009, pp. 627–636.

[8] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *JSAC*, vol. 35, no. 11, pp. 2616–2624, 2017.

[9] I. File, "Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation)," *General Data Protection Regulation*, 2015.

[10] H. Kaur, N. Kumar, and S. Batra, "An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system," *FGCS*, vol. 86, pp. 297–307, 2018.

[11] S. Badsha, X. Yi, I. Khalil, and E. Bertino, "Privacy preserving user-based recommender system," in *ICDCS*. IEEE, 2017, pp. 1074–1083.

[12] N. Polatidis, C. K. Georgiadis, E. Pimenidis, and H. Mouratidis, "Privacy-preserving collaborative recommendations based on random perturbations," *Expert Systems with Applications*, vol. 71, pp. 18–25, 2017.

[13] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, "Differentially private data publishing and analysis: A survey," *TKDE*, vol. 29, no. 8, pp. 1619–1638, 2017.

[14] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS*. ACM, 2016, pp. 308–318.

[15] A. Friedman and A. Schuster, "Data mining with differential privacy," in *SIGKDD*. ACM, 2010, pp. 493–502.

[16] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *TPAMI*, vol. 40, no. 4, pp. 769–790, 2017.

[17] H. Liang, H. Du, Q. Wang *et al.*, "Real-time collaborative filtering recommender systems." in *AusDM*, 2014, pp. 227–231.

[18] A. Narayanan and V. Shmatikov, "How to break anonymity of the netflix prize dataset," *arXiv preprint cs/0610105*, 2006.

[19] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *CCS*. ACM, 2014, pp. 1054–1067.

[20] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *ICDM*. ACM, 2018, pp. 1655–1658.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *UAI*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.