PrivRec: User-Centric Differentially Private Collaborative Filtering Using LSH and KD

Yifei Zhang[1,2], Neng Gao[1(✉)], Junsha Chen[1,2], Chenyang Tu[1], and Jiong Wang[1,2]

[1] SKLOIS, Institute of Information Engineering, CAS, Beijing, China
{zhangyifei,gaoneng,chenjunsha,tuchenyang,wangjiong}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. The collaborative filtering (CF)-based recommender systems provide recommendations by collecting users' historical ratings and predicting their preferences on new items. However, this inevitably brings privacy concerns since the collected data might reveal sensitive information of users, when training a recommendation model and applying the trained model (i.e., testing the model). Existing differential privacy (DP)-based approaches generally have non-negligible trade-offs in recommendation utility, and often serve as centralized server-side approaches that overlook the privacy during testing when applying the trained models in practice. In this paper, we propose PrivRec, a user-centric differential private collaborative filtering approach, that provides privacy guarantees both intuitively and theoretically while preserving recommendation utility. PrivRec is based on the locality sensitive hashing (LSH) and the teacher-student knowledge distillation (KD) techniques. A teacher model is trained on the original user data without privacy constraints, and a student model learns from the hidden layers of the teacher model. The published student model is trained without access to the original user data and takes the locally processed data as input for privacy. The experimental results on real-world datasets show that our approach provides promising utility with privacy guarantees compared to the commonly used approaches.

Keywords: Information security · Neural collaborative filtering · Differential privacy · Locality sensitive hashing · Knowledge distillation

1 Introduction

Collaborative filtering (CF) leverages the historical interactions between users and items to predict their preferences on a new set of items [12]. It provides recommendations by modeling users' preferences on items based on their historical user-item interactions (e.g., explicit five-star rating and implicit 0–1 feedback

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 113–121, 2020.
https://doi.org/10.1007/978-3-030-63820-7_13

on items) [5]. However, the direct collecting and modeling on the original data might reveal personally sensitive data and brings privacy concerns, since a neural model generally overfits on specific training examples in the sense that some of these examples are implicitly memorized.

Recently, many researches focus on DP-based recommendation approaches. Differential privacy (DP) has emerged as a strong privacy notation with a provable privacy guarantee. McSherry et al. [8] applied differential privacy theory to recommender systems for the first time, and Nguyên et al. [9] applied local differential privacy (LDP) to help users to hide their own information even from first-party services during the data collection process. However, existing DP-based models generally focus on the training data privacy but overlook the data privacy in practice while applying trained models (namely during testing). A DP-based CF model needs to be retrained when applying it on new users that are not in the training data, which is computationally expensive. Intuitively, an LDP-based method can protect user privacy during testing by applying random response from the client side. However, LDP can only guarantee the accuracy of the statistical result while avoiding individual record disclosure, leading to the fact that a single input from the client is often flooded with too much random noises, which reduces the recommendation utility.

In this work, we propose PrivRec, a user-centric differentially private collaborative filtering approach, that preserves recommendation utility. PrivRec leverages a user-centric privacy enhancing algorithm to privately model users from the client sides, which protects the data privacy of both training data and the data during testing. Specifically, an LSH-based user data modeling process is applied to generate user representations with intuitive privacy, and the Laplace mechanism is leveraged to provide theoretical privacy. Moreover, a knowledge distillation architecture is applied for further privacy guarantee, as the released model does not have the access of the original sensitive user information in the training data. Our contributions can be summarized as follows:

- We propose PrivRec, a user-centric differentially private collaborative filtering approach that protects user privacy, while retaining recommendation utility.
- We address the challenge of the privacy-enhanced client-side utility-preserving user modeling with a locality sensitive hashing-based user representation algorithm that applies Laplace mechanism.
- We prevent the privacy disclosure from the potential overfitting of the models by introducing the knowledge distillation architecture. The released student model is trained without any access to the original sensitive data.
- Experimental results on two real-world datasets demonstrate that PrivRec outperforms other neural collaborative filtering-based methods on retaining recommendation utility with privacy guarantees.

2 Preliminaries

2.1 Differential Privacy

Differential privacy (DP) [3] has become the *de facto* standard for privacy preserving problems. Local differential privacy (LDP) is a special case of DP where the random perturbation is performed by the users on the client side.

Definition 1. *A randomized mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (ϵ-DP) if for any adjacent sets d, d' differing by only one record for any subset S of outputs $S \subseteq R$,*

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(d') \in S], \tag{1}$$

where Pr *denotes the probability and ϵ is positive. Lower values of ϵ indicates higher degree of privacy.*

Laplace Mechanism is the most commonly used tool in differential privacy and has been applied in a number of works on differential privacy analysis [8,15].

Definition 2. *For a real-valued query function $q : \mathcal{D}^n \rightarrow R$ with sensitivity Δ, the output of Laplacian mechanism will be,*

$$\mathcal{M}(d) = q(d) + Lap(\frac{\Delta}{\epsilon}), \tag{2}$$

where $Lap(\Delta)$ is a random variable drawn from the probability density function

$$Lap(x) = \frac{1}{2\Delta} e^{-\frac{|x|}{\Delta}}, \forall x \in R. \tag{3}$$

2.2 Locality Sensitive Hashing

Locality Sensitive Hashing (LSH) [4] is an effective approach for approximate nearest neighbor search and has been applied in many privacy-preserving tasks [14]. The main idea of LSH is to find a hashing function or a family of hashing functions such that a hash collision occurs on similar data with higher probability than others, i.e., it is likely that, (i) two neighboring points are still neighbors after hashing, and (ii) two non-neighboring point are still not neighbors after hashing. For data in domain S with distance measure D, an LSH family is defined as:

Definition 3. *A family of hashing functions $\mathcal{H} = \{h : S \rightarrow U\}$ is called (d_1, d_2, p_1, p_2)-sensitive, if for any $x, y \in S$,*

$$\textit{If } D(x, y) \leq d_1, \textit{then } \Pr[h(x) = h(y)] \geq p_1, \tag{4}$$

$$\textit{If } D(x, y) \geq d_2, \textit{then } \Pr[h(x) = h(y)] \leq p_2. \tag{5}$$

MinHash. Minwise hashing [2] is the LSH for resemblance similarity, which is usually leveraged to measure text similarity. MinHash applies a random permutation (i.e., hashing function) $\pi : \Omega \rightarrow \Omega$ on the given set S, and stores only the minimum value of the results of the permutation mapping. Specifically, a permutation mapping randomly shuffles all possible items of the input, and returns the corresponding indices of input items. Formally, the result of MinHash (namely *signature*) is defined as,

$$h_{\pi}^{min}(S) = \min(\pi(S)). \tag{6}$$

Given sets S_1 and S_2, the probability of that the two sets have the same signature is shown as,

$$\Pr(h_{\pi}^{min}(S_1) = h_{\pi}^{min}(S_2)) = \frac{|S_1 \bigcap S_2|}{|S_1 \bigcup S_2|}, \tag{7}$$

By applying multiple independent MinHash permutations, the probability of the two sets having the same MinHash signature is an unbiased estimate of their Jaccard similarity [13].

2.3 Knowledge Distillation

Knowledge Distillation (KD) is introduced by Hinton et al. [6], to transfer "knowledge" from one machine learning model (the *teacher*) to another (the *student*). The main idea of KD is that the student model can learn the distilled information directly from the teacher model, in order to reduce the amount of parameters while retain the performance. Specifically, the teacher model is trained on the original training data with ground-truth labels, and then the student model is fed with the same input but set the outputs of hidden layers in the teacher model as targets.

3 Threat Model

We are addressing two major privacy threats, *training data privacy* and *privacy in practice*. The training data privacy is the common problem since the identity-revealing user representations are stored in the first-party service providers or memorized by the potential overfitting of machine learning models. The potential data breach brings privacy threats on the training data. When applying a trained machine learning model, the privacy in practice is often overlooked. Consider a client-server scenario, a client-side user is faced with one of the following problems: (i) if the user is an existing user whose historical data are in the training set, it is expected to send its identifier (e.g., user ID, cookies) so that the server can retrieve the corresponding representations, which reveals its identity; (ii) if the user is a new user, it cannot get personalized recommendations since they are not modeled, or it has to send its historical data for the server to infer its preferences, which brings threats of privacy disclosure.

4 Design of PrivRec

4.1 Framework

Our approach follows the teacher-student knowledge distillation (KD) structure. We illustrate the visualized framework of PrivRec in Fig. 1. It consists of two parts: the teacher model and the student model. We leverage a multi-layer perceptron (MLP) to perform neural collaborative filtering (NCF) by predicting the interaction between a user and an item. It is the prototype of many recent researches on CF-based neural recommender systems [1,5,7,11]. Next, we will present the specifics of the PrivRec.

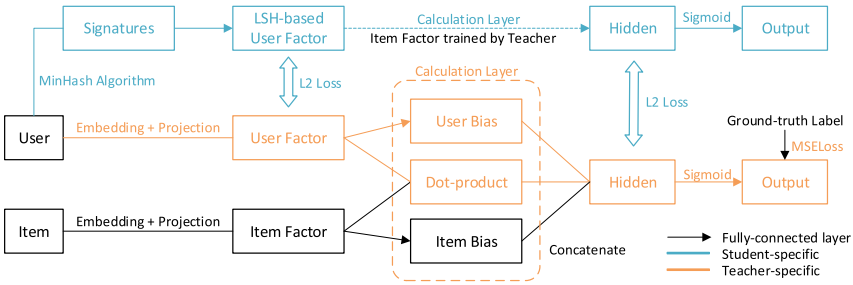


Fig. 1. The general framework of our approach. The teacher-only modules are replaced with the student-only modules in the student model.

4.2 Components

Teacher. The teacher model is a naive NCF model trained on the training data without any privacy constraints. Consider a user u and an item v, the network structure is described as follows. The model takes the identifiers u and v as input and embeds them to the d_u and d_v-dimensional latent spaces with embedding layers $emb_u : u \to p_u$ and $emb_v : v \to p_v$. Then, p_u and p_v are projected to matrix factorization spaces with the same dimension d_m, denoted as $m_u : p_u \to f_u$ and $m_v : p_v \to f_v$. The calculation layer concatenates the summed dot-product of the factors $dot(f_u, f_v)$ and the results of two bias layers $bias : f \to b$ for the two factors. Finally, the result of the calculation layer is passed to a regression layer composed by a fully-connected layer with the sigmoid activation for the final prediction $\sigma : l_{u,v} \to \hat{r}_{u,v}$. Overall, the teacher learns the latent representation of users and items for the prediction. The teacher model is not published for privacy of the training data. Formally, we have:

$$f_u = m_u(emb_u(u)), \quad f_v = m_v(emb_v(v)), \tag{8}$$

$$\hat{r}_{u,v} = \sigma(calc(f_u, f_v)) = \sigma(cat(dot(f_u, f_v), bias(f_u), bias(f_v))). \tag{9}$$

Student. The structure of the student model is similar with the teacher's, but takes different inputs and targets. It is this model that be published to the deployment environment in practice. We reuse the item embeddings p_v in the teacher model, since they do not reveal user-specific information. The user embedding layer is replaced by a differentially private locality sensitive hashing(LSH)-based representation procedure described in Algorithm 1. The MinHash functions are *publicly available* and identical for all user, so that the permutations stay unchanged both in server-side training process and in client-side practice. The hashing signatures are arranged into a vector, where each entry represents a perspective of the user data. Accordingly, the user representation can be computed completely on the client side with the publicly available LSH permutations and the user's private positive item set.

Algorithm 1: Differentially private LSH-based user representation.

Input: User set U, item set V, expected dimention of vector representation K, global privacy sensitivity Δ, overall privacy budget ϵ.
Output: Representation vectors of users.
Initialize k independent MinHash random permutations $\pi = \{\pi_1, \pi_2, \cdots, \pi_K\}$;
foreach $u \in U$ **do**
  Select the subset of items that u rated over its average $S_u = \{u | r_{u,v} > \bar{r}_u\}$;
  foreach $\pi_i \in \pi$ **do**
    Generate a MinHash signature $h_i^{min}(u) = \min(\pi_i(S_u))$;
    $h_i^{min}(u) = h_i^{min}(u) \mod 2$; // *A hashing result itself has no numerical significance for additive noises. A "mod 2" operation is leveraged to binarize the signatures, so that applying Laplace mechanism on MinHash signatures will not completely invalidate the user representation.*
    $h_i^{min}(u) = h_i^{min}(u) + Lap(\frac{\Delta}{\epsilon})$;
  Concatenate the k signatures $h_u = [h_1^{min}(u), h_2^{min}(u), \cdots, h_k^{min}(u)]$;
  A fully-connected layer FC is applied on h_u;
  Append h_u as u's representation vector to result;
Return the user representation vectors.

The remaining difference between the teacher model and the student model is that the student is not trained on the ground-truth labels. Instead, it learns from the output of the teacher's user factorization layer f_u and the last hidden output l_u, v. The LSH-based user representation sig_u obtained from the above algorithm is projected to a fully-connected layer as the student's user factor $fc : sig_u \rightarrow f'_u$, and the output of the calculation layer is $l'_{u,v}$.

We summarize the working process of the student model as follows. (i) It projects user u's identifier into a latent space with the LSH-based algorithm for the MinHash signatures sig_u as Algorithm 1; (ii) The signatures are fed into a fully-connected layer to get the student's user factor $fc : sig_u \rightarrow f'_u$; (iii) the item v's factor f_v is obtained from the teacher model; (iv) the factors f'_u and f_v are passed to the calculation layer; (v) finally, the regression layer is applied to produce the final prediction.

Detailed privacy analysis of PrivRec is omitted due to page limit. Please refer to the extended version on arXiv.

5 Experiments

5.1 Experimental Settings

Datasets. We conduct our experiments on the commonly used MovieLens 1M dataset of movie ratings and the Jester dataset of joke ratings. We filter out the invalid data, including the users and items with no ratings in the dataset, and the users that rate equally on all items.

Metric. To measure the effectiveness of knowledge distillation . The recommendation utility is measured as the accuracy of rating prediction. We adopt mean absolute error $\text{mae}(y_{predict}, y_{truth}) = E|y_{predict} - y_{truth}|$ as the metric of the recommendation accuracy. A lower mae indicates a more precise rating prediction on collaborative filtering.

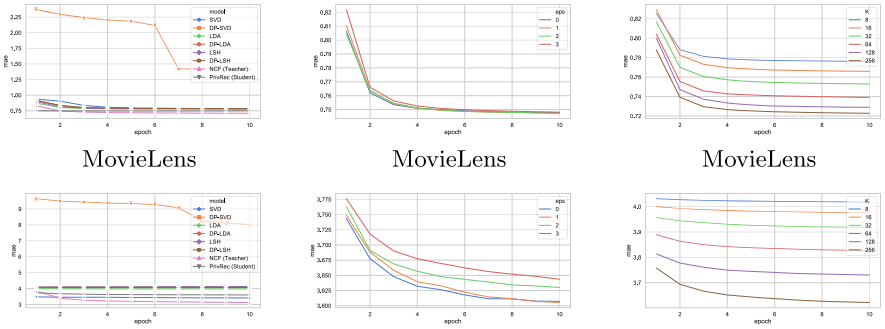
Baselines. We evaluate our model by comparing with several widely used baseline methods. Specifically, neural collaborative filtering models with following pretrained user representations are considered:

- NCF: The original Neural Collaborative Filtering method that models users and items into representation vectors. It is the prototype of our teacher model without privacy constraints.
- SVD: Singular Value Decomposition, a matrix factorization method, is one of the most popular collaborative filtering methods. It takes user-item interaction matrix as input, and returns the low-dimensional representations of users.
- LDA: Latent Dirichlet Allocation is an unsupervised learning algorithm for natural language processing initially, and discover topics based on contents. We employ LDA to learn a user's latent preference as its representation.
- LSH: Locality Sensitive Hashing is the prototype of our LSH-based user representation. It applies multiple MinHash signatures on a user's historical data as the user's representation vector.
- DP-SVD, DP-LDA and DP-LSH: We apply the differentially private Laplace mechanism on the baselines above to explore the utility preservation of our LSH-based differentially private user representation.

5.2 Results

We compare our PrivRec with the mentioned baselines for the neural collaborative filtering recommender systems. As shown in Fig. 2(a), by comparing the first six methods in the legend ([DP-]SVD, [DP-]LDA, [DP-]LSH), we observe that Laplace mechanism significantly degrades the recommendation service utility

of SVD and LDA, while LSH-based method is having less trade-off in applying differential privacy. This demonstrates that the LSH-based user representation implies more user preferences than traditional methods after introducing the same amount of noises. According to the last three methods in the legend (DP-LSH, NCF, PrivRec), the knowledge distillation (KD) architecture of our PrivRec substantially improves the recommendation utility of the DP-LSH method. PrivRec shows comparable averaged recommendation utility with the baselines, with the same privacy budget of differential privacy and intuitively stronger privacy within its knowledge distillation training process.



(a) Comparisons on mean absolute error of CF precision.

(b) The influence of privacy budget ϵ (eps $= -\log \epsilon$ for simplicity).

(c) The influence of different amount of MinHash signatures (K).

Fig. 2. Experimental results.

Further experimental results on the influence of the noise scale and the amount of MinHash signatures are shown in Figs. 2(b) and (c). The influence on ϵ is measured with the averaged result on different K, and vice versa. A greater eps indicates smaller ϵ, which introduces more noises into MinHash signatures and provides stronger guarantee on user privacy, while slightly downgrading prediction accuracy. A larger K implies more facets of hashing and more information of user data, and further reduces rating prediction error.

6 Conclusion

In this work, we focus on both training data privacy and the privacy in practice during testing in neural collaborative filtering. The proposed PrivRec is an early effort to protect user privacy in practice from the client side. It manages to model a user locally and privately with a LSH-based algorithm and the DP principle. PrivRec shows promising results with privacy guarantees on NCF. In the future, we will extend it to more variety of recommender systems.

Acknowledgments. This work is supported by the National Key Research and Development Program of China.

References

1. Bai, T., Wen, J.R., Zhang, J., Zhao, W.X.: A neural collaborative filtering model with interaction-based neighborhood. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1979–1982 (2017)
2. Broder, A.Z., Charikar, M., Frieze, A.M., Mitzenmacher, M.: Min-wise independent permutations. J. Comput. Syst. Sci. **60**(3), 630–659 (2000)
3. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
4. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. VLDB **99**, 518–529 (1999)
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
7. Liu, Y., Wang, S., Khan, M.S., He, J.: A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. Big Data Min. Anal. **1**(3), 211–221 (2018)
8. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the NetFlix prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 627–636. ACM (2009)
9. Nguyên, T.T., Xiao, X., Yang, Y., Hui, S.C., Shin, H., Shin, J.: Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053 (2016)
10. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K.: Semi-supervised knowledge transfer for deep learning from private training data. arXiv preprint arXiv:1610.05755 (2016)
11. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009)
12. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
13. Shaked, S., Rokach, L.: Publishing differentially private medical events data. In: Buccafurri, F., Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-ARES 2016. LNCS, vol. 9817, pp. 219–235. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45507-5_15
14. Vatsalan, D., Sehili, Z., Christen, P., Rahm, E.: Privacy-preserving record linkage for big data: current approaches and research challenges. In: Zomaya, A.Y., Sakr, S. (eds.) Handbook of Big Data Technologies, pp. 851–895. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49340-4_25
15. Yin, C., Ju, X., Yin, Z., Wang, J.: Location recommendation privacy protection method based on location sensitivity division. EURASIP J. Wirel. Commun. Netw. **2019**(1), 266 (2019)