PrivateRec: Differentially Private Training and Serving for Federated News Recommendation

Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Yang Cao, Lingjuan Lyu, Weike Pan, Yun Chen, Hong Chen, Xing Xie

1 Renmin University of China, 2 Microsoft Research Asia, 3 Tsinghua University, 4 Kyoto University, 5 Sony, 6 Shenzhen University, 7 Shanghai University of Finance and Economics

{ruixuan.liu, chong}@ruc.edu.cn, {wufangzhao, wuchuhan15}@gmail.com, {yanlinwang, xingx}@microsoft.com, yang@t.kyoto-u.ac.jp, lingjuan.lv@sony.com, panweike@szu.edu.cn, yunchen@sufe.edu.cn

ABSTRACT

Privacy protection is an essential issue in personalized news recommendation, and federated learning can potentially mitigate the privacy concern by training personalized news recommendation models over decentralized user data. For a theoretical privacy guarantee, differential privacy is necessary. However, applying differential privacy to federated recommendation training and serving conventionally suffers from the unsatisfactory trade-off between privacy and utility due to the high-dimensional characteristics of model gradients and hidden representations. In addition, there is no formal privacy guarantee for both training and serving in federated recommendation. In this paper, we propose a unified federated news recommendation model training method for effective and privacy-preserving model training and online serving with differential privacy guarantees. We first clarify the notion of differential privacy over users’ behavior data for both model training and online serving in the federated recommendation scenario. Next, we propose a privacy-preserving online serving mechanism under this definition with differentially private user interest decomposition. More specifically, it decomposes the high-dimensional and privacy-sensitive user embedding into a combination of public basic vectors and adds noise to the combination coefficients. In this way, it can avoid the dimension curse and improve the utility by reducing the required noise intensity for differential privacy. Besides, we design a federated recommendation model training method with differential privacy, which can avoid the dimension-dependent noise for large models via label permutation and differentially private attention modules. Experiments on real-world news recommendation datasets validate the effectiveness of our method in achieving a good trade-off between privacy protection and utility for federated news recommendations.

KEYWORDS

Federated Learning, Privacy Protection, Recommender System

ACM Reference Format:
Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Yang Cao, Lingjuan Lyu, Weike Pan, Yun Chen, Hong Chen, Xing Xie. 2022. PrivateRec: Differentially Private Training and Serving for Federated News Recommendation. In Proceedings of ACM Conference (Conference’17). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnn.nnnnnn

1 INTRODUCTION

Nowadays, news recommendation systems are indispensable for users to filter a large amount of news articles and relieve the information overload. For an accurate performance, most news recommendation methods [4, 7, 15, 30, 35, 42] depend on training models over user behavior data such as historical clicks to learn users’ personalized interests. During recommendation services, personal behavior data are also necessary for the model to rank candidate news articles that fit user interests. Therefore, existing news recommendation systems entail a frequent collection of behavior data for training and serving into a centralized storage. However, the behavior data contains abundant personal identifiable information which can be manipulated by the recommendation service providers. Thus, an ever-increasing privacy concern arises in the whole society for the current recommendation system. Many law regulations (e.g., GDPR) are adopted to limit the transportation and exploitation of personal data. As a result, collecting behavior data for training and serving in recommendation systems may be forbidden in the near future, making an effective recommendation service challenging.

Federated learning (FL) [19] is proposed as a new paradigm to train models on scattered data, which mitigates the privacy concern because personal data are only kept on users’ devices. Based on FL, a federated news recommendation framework [27] enables to train a personalized recommendation model without collecting local behavior data. The global recommendation model is updated over multiple rounds of communications between the central server and multiple users until convergence. In each round, the server distributes the current global model to users. Then each user uploads model gradients to the server after local training on personal behavior data. Finally, the server aggregates local gradients and updates the global model. Thus, multiple users and a server collaboratively train news recommendation models by exchanging model gradients instead of accessing local data. However, two critical challenges hinder the implementation of federated news recommendations.

First, the existing federated news recommendation [27] does not guarantee a theoretical privacy, which causes legal and regulatory barriers in practical deployment. Existing privacy attacks indicate the possibility of inferring private information in general federated training by observing local model updates or even manipulating global model parameters [23, 43]. Similarly, potential adversaries

https://gdpr-info.eu
in federated recommendations can infer users’ clicking histories and track their personal intents [5]. Furthermore, any form of interactions with local behavior data in recommendation serving carries a certain amount of private information. Thus, bounding the theoretical privacy level in federated news recommendation training and serving is essential.

Second, a utility hurdle arises in the attempt of providing a theoretical privacy guarantee for federated news recommendation. As a golden standard of privacy criterion, differential privacy (DP) [10] quantifies the impact of any individual data record on the algorithm’s output from a probabilistic perspective. A direct way to protect user privacy in federated training is perturbing the local model update with the local DP noise [8, 10] before sending to the server. Since the noise magnitude is dimension-dependent, the utility drop is significant for large deep models with an enormous amount of parameters, which are normal for news recommendations. For the federated model serving with local behavior data, a simple strategy is local serving [6, 13], where the server sends all candidate items to each user’s device for a local ranking and display. However, news recommendation involves millions of candidate news. Thus, the local serving is impractical due to enormous communication and memory costs. Thus, we focus on the more practical way of online serving [27], where each user encodes historical clicks as a user embedding and sends it to the server for a ranked news list. To preserve private information encoded in the user embedding, a natural choice is to send a user embedding perturbed with DP noise. Typically, a larger-dimensional user embedding is more potent for describing user interests, but the utility is ruined with the dimension-dependent noise. Therefore, it is challenging to provide a satisfactory utility under reasonable privacy for training and serving in federated news recommendations.

To solve above challenges, we propose a federated news recommendation framework PrivateRec for a theoretical privacy guarantee and a decent performance. First, the theoretical privacy level in PrivateRec is quantified with the standard differential privacy [10]. The noise is applied when representing a user into an embedding during the federated training and serving for protecting historical clicks. Additionally, clicked labels in the training data are preserved by a proposed DP label permutation module, which results in an overall privacy guarantee of PrivateRec. Second, the severe utility drop caused by dimension-dependent noise in naive DP solutions can be mitigated by decomposing a user embedding into low-dimensional attention scores before applying DP noise in both training and serving. We further reduce the noise magnitude and improve the utility with the privacy amplification effect of the random padding in the user encoder.

To summarize, the main contributions of this paper are summarized as follows.

- We propose a novel and unified framework PrivateRec for federated news recommendations. The private information that users expose to the server throughout training and serving is bounded by the differential privacy.
- We design privacy-preserving mechanisms for federated training and serving with a decent recommendation performance. We decompose the user embedding into a lower-dimensional attention vector to avoid the utility drop by the dimension curse for both training and inference. Then we design a random padding mechanism to further reduce injected noise and improve the performance.
- Extensive experiments and analysis conducted on real-world datasets validate the significant utility improvement of training and serving in PrivateRec compared to FedRec [27] with conventional DP solutions under the same privacy level.

2 RELATED WORK

2.1 Personalized News Recommendation

Personalized news recommendation is important for intelligent online news service. Many deep learning-based recommendation models have been proposed for this task [25, 34, 35, 37]. Generally, their frameworks include three core components, i.e., news model, user model, and click prediction module. The news model aims to learn news representations from news content. For example, Wu et al. [34] use CNN to learn local contextual word representations and attention network to select informative words. Wu et al. [38] apply the pre-trained language models as news encoders to empower news recommendation. The user model aims to model user interest from their clicked news. For example, Okura et al. [25] apply GRU network, Wu et al. [35] utilize a personalized attention, and Wu et al. [37] use the combination of multi-head self-attention with the additive attention networks. Last, the clicking prediction module estimates the matching score between a news embedding and a user embedding, which can be implemented with the dot product [3], the outer product [12] and the dense network [32]. However, these methods rely on centralized user data for model training and online serving, which may have privacy risks and concerns [27].

2.2 Federated Recommendation

Many federated recommendation systems [2, 13, 17, 27, 36] follow FedAvg [19], where users first pull a global model from the server and conduct local optimization with local data. Then the server aggregates gradients from a group of users and updates the global model. For instance, Ammad-Ud-Din et al. [2] propose a federated collaborative filtering system with users’ local implicit feedback by uploading the item matrices gradient to the server. For faster convergence, Muhammad et al. [22] propose to cluster users before sampling participants in each round. For efficient communication cost, Yi et al. [41] propose to maintain the news model on the server and avoid exchanging gradients of the news model.

On one hand, exchanging gradients still poses a threat to user privacy [5, 43]. Cryptography can be applied to defend against an untrusted server. Chai et al. [5] propose to protect user privacy by encrypting the local gradient with Paillier encryption. Yi et al. [41] apply the secure aggregation to avoid the server knowing a specific local gradient. Anonymity also helps mitigate privacy risks. Lin et al. [17] propose to protect users’ historical clicks by locally injecting dummy items with fake ratings. Ultimately, perturbing the local gradients with noise, [27, 28] can reduce privacy leakage. We note that the crypto-based methods [5, 41] do not provide protection for publishing the updated global model. The anonymity strategy [17] and noise perturbation method [27, 28] do not hold for a well-defined theoretical privacy guarantee. Moreover, previous
definitions for recommendations are various. They are inapplicable for protecting each behavior, such as the existence of a local App [13], the meta feature for an item [6], or the indistinguishment for any two users [24, 31].

On the other hand, only a few works discussed the recommendation model serving with local data. Qi et al. [28] proposed to train a unified framework for training and serving with both recall and ranking models. Even noise is injected on gradients for training and on user interests for serving, the theoretical privacy definition is not discussed. In other works [6, 13], the server sends all items to a user to perform local serving with the local data on multiple platforms, which may be impractical due to the high communication and memory cost. In this paper, we propose a privacy-preserving solution for federated training and online serving method with a theoretical privacy guarantee. Also, this protection is effective for further procedures after distributing the global model.

3 METHODOLOGY
3.1 System Overview
3.1.1 Threat Model. We first introduce the threat model as follows.

Parties. There are three parties in PrivateRec: 1) \( S \), the recommendation service provider (the server) who organizes the model training and returns the recommended results in serving. Typically, \( S \) is an honest-but-curious server that follows the standard workflow but is curious about user’s behavior data for more commercial interests. 2) \( \mathcal{U} = \mathcal{U}_t \cup \mathcal{U}_q \), the set of users \( \mathcal{U}_t \) who participate in federated training and the set of users \( \mathcal{U}_q \) who query recommendation services. 3) \( \mathcal{A} \), any potential third-party adversary who may spy the communication. The information that \( \mathcal{A} \) can access is no more than \( S \), so we focus on the privacy protection against \( S \) and the guarantee naturally holds against any \( \mathcal{A} \).

Adversary’s objectives. Given a victim user \( u \in \mathcal{U} \) whose behavior log is \( D \), we denote the local procedure as \( M \). The adversary’s objective for either training or serving is inferring whether an item \( i \in \mathcal{I} \) exists in \( D \) by observing the output \( M(D) \).

3.1.2 Privacy Definition. Based on the standard differential privacy definition [10], various privacy levels are defined for federated learning: McMahan et al. [20] define the privacy in a central view as the existence of a user cannot be inferred after publishing the global model, which requires a trusted server to conduct the perturbation. Nguyêñ et al. [24] and Shin et al. [31] assume the server is untrusted and define the privacy as the information sent from any two users are indistinguishable to the server, which erases the penalization by its definition. These definitions did not consider the privacy cost in federated model serving. In PrivateRec, we define the differential privacy as the plausibility of whether an item is clicked by a user when sending messages to the server. Thus, we can defend against the untrusted server while providing a fine-grained privacy notion for a reasonable personalization. Also, our privacy notion is unified across training and serving stages, which enables users to quantify the overall privacy cost in a federated recommendation service. Formally, the information leakage in either training or serving should be bounded as follows.

**Definition 1.** For two adjacent user behavior logs \( D, D' \in \mathbb{D} \) with only one clicked behavior different and any output \( z \in \text{Range}(M) \), a mechanism \( M : \mathbb{D} \rightarrow \text{Range}(M) \) is \((\epsilon, \delta)\)-differentially private if and only if:

\[
\Pr[M(D) = z] \leq e^\epsilon \cdot \Pr[M(D') = z] + \delta.
\]

3.1.3 PrivateRec Framework. We first introduce the global model in PrivateRec with the following trainable components. Then we present modular steps for serving and training.

- The news model \( \Theta_N \), which is composed of a word embedding layer \( \Theta_e \) and a feature extractor layer \( \Theta_f \). Given the token sequence of a news, \( \Theta_e \) first outputs a sequence of word embeddings \( \mathbf{r} = \{e_i \in \mathbb{R}^d, i \in [L]\} \) and then \( \Theta_f \) encodes the news embedding into \( \mathbf{r} \in \mathbb{R}^d \).
- The user model \( \Theta_U \), which takes a sequence of \( H \) historical clicked news embeddings \( \mathbf{r} = \{r_i \in \mathbb{R}^d, i \in [H]\} \) and outputs the user embedding \( u = \Theta_U(\mathbf{r}) \) with \( u \in \mathbb{R}^d \).
- The basic interest vectors \( \mathbf{b} = \{b_i \in \mathbb{R}^d, i \in [B]\} \), which are a set of bas to decompose each unique user embedding into a \( B \)-dimensional vector.

With above components, each user in PrivateRec model serving locally conducts the following modules, as shown in Fig. 1:

![Figure 1: Privacy-preserving online serving in PrivateRec.](image-url)
• The User Encoder (UE): The server randomly masks some items in a sequence of historical clicks and inputs them into the UE module. Then, $\Theta_N$ outputs a sequence of news embeddings and feeds it into $\Theta_U$ for a user embedding $u$.

• The Interest Decomposer (ID): By querying each basic vector in $b$ with the user embedding $u$, ID module decomposes $u$ into an attention vector $\alpha$.

• Differentially Private Attention (DPA): For privacy protection on the true attention score, DPA module perturbs $\alpha$ after clipping, and outputs a normalized attention vector $\tilde{\alpha}$.

Then, the server conducts the following modules to get the news recommendation results:

• Interest Reconstructor (IR): After receiving each perturbed attention vector $\tilde{\alpha}$, IR module recovers the user embedding as $\hat{u}$ by summing basic vectors $b$ linearly with weights $\tilde{\alpha}$.

• The Clicking Predictor (CP): CP module calculates the matching score $\tilde{y}$ between $\hat{u}$ and $C$ candidate news embeddings $\tilde{y}^c$. Then the server can rank all candidate news and return the recommendation results.

Similarly, the federated training in PrivateRec also includes the UE, ID, DPA, and IR modules, as shown in Fig. 2. For each training participant, the user behavior dataset $D$ includes the historical clicks $N_h$ and the candidate items $N_c$. The privacy protection for $N_h$ is similar as the privacy protection in the model serving, which goes through the UE, ID, DPA and IR module sequentially to derive the reconstructed user embedding $\hat{u}$. Then the training process entails two extra modules:

• Differentially Private Label Permutation (DPL): For protecting the clicking labels in $N_h$, DPL permutes the actual clicks in a differentially private way and outputs the news embeddings of candidate news. Then the loss is calculated with the reconstructed user embedding and candidate embeddings. Finally, the user sends the local gradients $\nabla \Theta$ to the server.

### Algorithm 1 Online Serving in PrivateRec

| Line | Description |
|------|-------------|
| 1. | $u \in \mathcal{U}_g$: $N_h, \theta, \delta, \epsilon, p, \Theta = \Theta_U \circ \Theta_N$ |
| 2. | $\tilde{\alpha} = \text{GetPrivAttn}(N_h, \theta, \epsilon, \delta, p, \hat{u}, \Theta_U, \Theta_N)$ |
| 3. | send $\tilde{\alpha}$ to $\mathcal{S}$ |
| 4. | // IR modules |
| 5. | reconstruct user embedding $\hat{u} = \sum_{i}^{C} \tilde{\alpha}_i b_i$ |
| 6. | // CP modules |
| 7. | get news embeddings for $n_i \in N_h$ with $r_{\epsilon|C} = \Theta_N(n_i)$ |
| 8. | get matching scores over $C$ news $\tilde{y} = \{y_i = \hat{u}^T \cdot r_i, \ i \in [C]\}$ |
| 9. | return the ranked news list |

• Aggregate and Update (AU): In the last step, the server runs the AU module to aggregate all local model gradients and updates the global model for distributing in the next round.

## 3.2 Privacy-Preserving Online Serving

In this section, we introduce detailed modules in PrivateRec model serving, which lay the foundation for our federated training design. When the last round global model is deployed on local devices, the more practical strategy for recommendation serving without downloading a large volume of news candidates is online serving. In non-privacy-preserving online serving, each user sends a user embedding $u$ to $\mathcal{S}$ for describing personal interests. We denote the local dataset as the set of all historical clicked news $D = N_h$ and the set of $C$ candidate news items stored on the server as $N_c$. With the local mechanism $M$, we are supposed to ensure $M$ satisfies $(\epsilon, \delta)$-DP in Definition 1 when a user sends $M(D)$ to the server for a recommendation results from $N_c$.

### 3.2.1 Vanilla DP User Embedding (VDP)

A conventional paradigm [9] for this goal is locally perturbing the user embedding. The user embedding is first clipped to $\theta$ with $\hat{u} = \frac{u}{\max(1, \|u\|_2)}$. 
Algorithm 2 GetPrivAttn()

Input: \( N_t, \theta, \epsilon, \nu, \theta_U, \Theta_N = \Theta_U \circ \theta_U \circ \theta_U \circ \theta_U \circ \epsilon_0 \)

Output: \( \tilde{\alpha} \)

1: \( r_t = \Theta_F(\hat{e}_0) \), sensitivity \( S = \theta \)
2: for each clicked item \( n_i \in N_t \) do
3: \( r_i = \Theta_N(n_i) \)
4: \( r_i = \begin{cases} r_i & \text{w.p. } 1-p, \\ r_0 & \text{w.p. } p. \end{cases} \)
5: end for
6: user embedding \( u = \Theta_U(\tilde{r}) \)
7: attention score \( \alpha = \text{Softmax}(\frac{QK^T}{\sqrt{d}}) \), where \( Q = u, K = \tilde{b} \)
8: clipping \( \tilde{\alpha} = \frac{\alpha}{\max(1, |\tilde{r}|)} \)
9: sample noise vector \( \tilde{n} \sim \mathcal{N}(0, \sigma^2I_{d \times d}) \), where \( \sigma = \frac{S}{\log \frac{e^{\frac{d}{2}}}{\epsilon} - \epsilon} \)
10: return private attention vector \( \tilde{\alpha} = \{ \tilde{\alpha}_i = \frac{\tilde{n} \cdot \tilde{e}_i}{\max(1, |\tilde{r}|)} \}_{i \in [B]} \)

Figure 3: Details of the news model and the padding vector.

Then a noise vector \( \tilde{n} \) is drawn from the Gaussian distribution \( \mathcal{N}(0, \sigma^2I_{d \times d}) \), where \( \sigma = \sqrt{2 \log \frac{e^{\frac{d}{2}}}{\epsilon} - \epsilon} \). The sensitivity \( S = \max_{D',D} ||M(D) - M(D')||_2 \) bounds the maximum change that one clicked item causes to the user embedding. Finally, the user sends \( \tilde{u} = u + \tilde{n} \) to the server. If the Laplace mechanism is applied when \( \delta = 0 \), \( \tilde{n} \) is drawn from \( \text{Lap}(S/\epsilon_0) \). Usually, the user embedding with a larger dimension encodes more information. However, it is obvious that the intensity of noise \( \mathbb{E}[||\tilde{n}||^2] \) scales with the dimension \( d \) of the user embedding. In other words, the information in a higher-dimensional user embedding would be submerged by the noise. Hence, VDP cannot provide a decent trade-off between utility and privacy, as we validate in experiments.

3.2.2 DP User Embedding with Interest Decomposition. Thus, we are motivated to reduce the intensity of noise by decomposing \( u \) into a lower-dimensional vector before the perturbation. As shown in Algorithm 1, each user in PrivateRec sends the perturbed lower dimensional vector to the server for a ranked result.

The key step of the interest decomposition (ID) is Line 7 in Algorithm 2. We decompose the user embedding \( u \) into a low-dimensional vector \( \tilde{\alpha} \) by querying \( u \) to \( b \) with a scaled dot product, where \( b \) is a set of basic vectors trained for capturing \( B \) abstract user interests. Since variant user interests can be generalized into several basic vectors [39], we have \( B \ll d \). Thus, the intensity of the differentially private noise only scales with \( B \), which avoids the dimension curse on the user embedding.

Then, we provide the DP guarantee by perturbing the vector \( \tilde{\alpha} \) with DPA module, as shown from Line 9 to Line 10 of Algorithm 2. It should be noted that the sensitivity is \( S = \theta \) because attention scores are always positive for any two adjacent datasets. We use SoftPlus [11] for positive attention weights and apply a normalization for keeping the summation as 1. After receiving a \( B \) dimensional attention vector \( \tilde{\alpha} \), \( S \) can reconstruct the user embedding with a weighted summation over \( \tilde{b} \) in Line 5 of Algorithm 1. So the communication cost that a user spends for a recommendation service query is reduced from \( O(d) \) to \( O(B) \).

3.2.3 Privacy Amplification by Behavior Padding. Based on the interest decomposition, we further reduce the noise magnitude with the privacy amplification effect from a random padding, as shown from Line 1 to Line 5 in Algorithm 2. In other words, by randomly masking some items with public information, we can apply a smaller \( \sigma \) for a given privacy. Previous works [18, 33] that utilize this effect usually mask values into null, which inevitably incur the information loss. Instead, we pad the masked news to an anonymous news embedding \( r_0 \) with the probability of \( p \). It is generated from the feature extractor \( \Theta_B \) by inputting a sequence of padding token embedding \( e_0 \) as shown in Fig. 3. Since the padding token embedding \( e_0 \) is used to pad empty token in training, the generated \( r_0 \) is more informative than null by encoding some general interest information.

Since the \( r_0 \) is public, this does not incur any extra privacy concerns than masking items to null values. It can improve the model serving utility with less noise under the same level of privacy while reducing the information loss caused by masking actual items to null. We will elaborate on how to train the padding token embedding \( e_0 \) and basic vectors \( \overline{b} \) in the next section. Ultimately, with the post-processing property [10], we can derive the privacy guarantee as follows.

Theorem 1. With GetPrivAttn() as the local mechanism \( \mathcal{M} \), sending \( \tilde{\alpha} = \mathcal{M}(D) \) to \( S \) in Algorithm 1 is \((\epsilon_x, \delta_x)\) -DP for any \( D \in \mathcal{D} \).

3.3 Privacy-Preserving Model Training

As shown in Fig. 2, the global model in PrivateRec is \( \Theta = \Theta_N \circ \Theta_U \circ \Theta_U \circ \Theta_U \circ \Theta_U \circ \epsilon_0 \). Compared to the conventional federated news recommendation framework [27], we introduce a set of basic vectors \( \overline{b} \) for getting the basics to decompose the user embedding. The local behavior dataset \( D \) comprises the set of \( H \) historical clicks \( N_h \) and the candidate set \( N_c \) with \( C \) candidates with a clicked item.

For each training round as shown in Algorithm 3, the server \( S \) first samples \( r \) percent of users \( U_t \) and distributes \( \Theta \) to them. Then, the sampled user \( u_r \) updates the local copy of \( \Theta \) with the mechanism \( \mathcal{M} \) and sends \( \forall \Theta_i \) to the server. Finally, the server aggregates local gradients and updates \( \Theta \) with FedAdam [29]. Since only local gradients generated from \( M \) are exposed, we aim to guarantee the \( M \) satisfies \((\epsilon_x, \delta_x)\) -DP over \( D \).

The DPA module provides \((\epsilon_x, \delta_x)\) -DP for \( N_b \). Recall that we require an anonymous news embedding \( r_0 \) which can encode general context information for all items. We illustrate two kinds of
We conduct experiments on two real-world datasets: MIND and NewsFeeds. MIND\(^1\) [40] is a public dataset for news recommendation, which was collected from anonymous user behavior logs over 6 weeks (Oct. 12 to Nov. 22, 2019) from MSN News\(^2\) website. NewsFeeds contains behavior logs of 10,000 users from the news feeds app over two weeks (Aug. 1st to Sep. 1st, 2020). We utilize the impression logs in the first three weeks as the training data. User logs in the latter two days are used for validation and the rest in the last week are used for testing. The statistical details of datasets are shown in Table 1. We show the generality of the proposed solution by evaluating over three news recommendation models: NRMS [37], NAML [34], and PLM-NR [38].

In our experiments, we use the Glove embeddings [26] pre-trained over 840 billion tokens to initialize the word embeddings. By default, the dimension of news embedding and user embedding is 400. The dimension of query vectors in the attention module is 200. The number of attention heads is 20. We apply a dropout ratio of 0.2 to the word embeddings and the multi-head attention outputs. In each training sample, the number of negative samples is 4. For each round in the federated model training, we sample 5% and 1% users for NewsFeeds and MIND to conduct local optimization and update the global model. We use FedAdam [29] as the optimization algorithm for all federated baselines and we use Adam [14] for optimizing centralized baselines. We apply the Laplace noise in all privacy-preserving federated training with \(\delta_r = 0\) and set \(\delta_s = 0.001\) for Gaussian noise injected in serving. Considering the limitation of GPU memory, the maximum number of user behaviors for modeling user embedding is 50 for all experiments, and the length of news titles is clipped to 30. Following previous works [27], we utilize averaged AUC, MRR, nDCG@5, and nDCG@10 overall impressions to measure the recommendation utility. We independently repeat each experiment 5 times and report the averaged results.

Baselines in our experiments include: 1) Centralized recommendation, where the server trains the model over all collected personal data. 2) DP-FedRec, where a global recommendation model is trained over local data. Local gradient and user embedding is perturbed for training and serving in DP-RedRec for a comparison under the same privacy level. Under the privacy definition 1, the noise is applied to each local gradient in training with gradient clipping norm \(\theta = 0.005\) and to the user embedding in serving with clipping threshold \(\theta = 0.001\). It should be noted that \(\epsilon = \infty\) indicates the non-private FedRec without any noise.

### 4 EXPERIMENTS

#### 4.1 Dataset and Experimental Settings

We conduct experiments on two real-world datasets: MIND and NewsFeeds. MIND\(^1\) [40] is a public dataset for news recommendation, which was collected from anonymous user behavior logs

---

\(^{1}\)MIND-small from https://msnnews.github.io/

\(^{2}\)https://www.msn.com/en-us/news
user privacy. \textit{FedRec} with $\epsilon_t = \infty$ can mitigate the direct privacy leakage without data collection and achieve the suboptimal performance, but no theoretical privacy is ensured during training. For privacy-preserving baselines, we observe that PrivateRec outperforms \textit{DP-FedRec} over both datasets with higher metric scores. Also, we observe a larger utility drop from \textit{FedRec} with $\epsilon_t = \infty$ to \textit{DP-FedRec} on the PLM-NR model, because the noise amount on gradients scales with the model size.

Then, we visualize the privacy-utility trade-off in federated training with various privacy budget $\epsilon_t \in \{1, 5, 10, 20, \infty\}$ in Fig. 4. First, the model performance of PrivateRec is better than PrivateRec for all budgets $\epsilon_t$. Additionally, the performance of PrivateRec for a small privacy budget $\epsilon_t = 5$ is still acceptable with AUC 61.35. Second, the decomposition of user interest has a negligible effect on the privacy budget $\epsilon_t$. When $\epsilon_t = \infty$, the performance of PrivateRec is comparable to the non-private baseline with $\epsilon_t = \infty$. This observation validates our claim that thousands of user interests can be summarized into several basic interests.

### 4.3 Privacy-preserving Serving Performance

Then, we evaluate the privacy-preserving model serving for \textit{PrivateRec} and \textit{DP-FedRec} in Table 3 under the same serving privacy budget $\epsilon_s$. First, we can observe that for two versions of baseline, i.e., \textit{DP-FedRec} with or without privacy-preserving federated training, \textit{PrivateRec} can achieve a significantly better performance across all models and datasets.

Then we investigate the privacy-utility trade-off in Fig. 5. First, we can observe that the model serving performance of \textit{PrivateRec} is consistently better than \textit{DP-FedRec} under various serving privacy budgets $\epsilon_s$. Second, we find the proper amount of noise injected in \textit{PrivateRec} training can help the privacy-preserving model serving. From Fig. 5, the performance of \textit{PrivateRec} with $\epsilon = \infty$ in the dashed line cannot outperform privacy-preserving training \textit{PrivateRec} for $\epsilon_s$ from 0.001 to 10. This fact emphasizes our claim that the design of privacy-preserving federated training should co-design with the privacy-preserving model serving as an end-to-end solution for the federated recommendation system. If $\epsilon_t$ is too small (e.g., 1), the performance upper bound under ($\epsilon_s = \infty$) is limited. If $\epsilon_t$ is too large (e.g., 20), the performance upper bound is higher. And under a smaller $\epsilon_s$, it is not as robust as the performance with a moderate $\epsilon_t = 10$. Thus, we conclude that the $\epsilon_s$ can be the knob to tune the privacy-utility trade-off in privacy-preserving model serving. More importantly, we find that the utility gain of \textit{PrivateRec} in federated training can maintain for privacy-preserving model serving, whereas \textit{DP-FedRec} cannot. As shown in Fig. 5, the utility of \textit{FedRec} with $\epsilon_s = \infty$ is better than PrivateRec. However, the model serving utility is catastrophically ruined by the DP noise.

### 4.4 Hyperparameter Analysis

#### 4.4.1 Effect of the Number of Basic Vectors
For a more comprehensive comparison, we conduct experiments to investigate if the user embedding dimension can mitigate the dimension curse problem in model serving instead of the proposed \textit{PrivateRec}. As shown in Fig. 7, we compare \textit{PrivateRec} with variant dimension $d$ of the user embedding in \textit{DP-FedRec}. To ignore the utility difference in training and only measure the utility loss from the noise perturbation, we define a metric $\tau$ as the skew level. Suppose the user embedding without noise is $\mathbf{u}$, the ranked results over the candidate set are denoted as $R$. With the privatized user embedding $\hat{u}$ in \textit{DP-FedRec} or the reconstructed user embedding $\tilde{u}$ in PrivateRec,
Privacy Budget \( s \)

Noise Scale \( \times10^3 \)

Noise Reduction by Padding

Privacy Budget \( s \)

We validate the claim of our motivation for interest decomposition module (ID) that the method of vanilla DP user embedding faces a dimension curse on utility. It should be noted that with \( e_f = \infty \) and \( e_t = \infty \), a larger user embedding usually brings a more informative representation and has a higher AUC. However, the utility advantage will disappear with privacy-preserving model serving. We show a similar results for Gaussian noise in the Appendix. Then, we observe that even if the FedRec model trained with a smaller \( d \) can slightly improve the private serving utility. It is still inferior to our proposed PrivateRec. Lastly, the skew level \( \tau \) of DP-FedRec is

| Model | \( e_t \) | \( e_s \) | Baseline | NewsFeeds | MIND | NewsFeeds | MIND |
|---|---|---|---|---|---|---|---|
| | | | AUC | MRR | nDCG5 | nDCG10 | AUC | MRR | nDCG5 | nDCG10 |
| NAML | \( \infty \) | 10 | DP-FedRec | 50.59±0.08 | 22.44±0.06 | 22.88±0.04 | 30.19±0.03 | 50.32±0.03 | 22.75±0.05 | 23.41±0.04 | 28.94±0.03 |
| | 10 | | DP-FedRec | 50.25±0.05 | 22.21±0.06 | 21.96±0.07 | 29.92±0.03 | 50.11±0.09 | 22.69±0.07 | 23.31±0.09 | 28.83±0.08 |
| | 10 | | PrivateRec | 61.63±0.19 | 29.10±0.21 | 30.81±0.26 | 38.81±0.22 | 57.52±0.13 | 26.39±0.22 | 27.88±0.26 | 33.45±0.22 |
| NRMS | \( \infty \) | 10 | DP-FedRec | 50.32±0.14 | 22.37±0.11 | 22.18±0.13 | 30.07±0.13 | 50.23±0.06 | 22.79±0.05 | 23.44±0.06 | 28.98±0.05 |
| | 10 | | DP-FedRec | 50.18±0.11 | 22.11±0.06 | 21.86±0.10 | 29.74±0.06 | 50.19±0.10 | 22.71±0.06 | 23.34±0.07 | 28.87±0.07 |
| | 10 | | PrivateRec | 61.37±0.22 | 28.77±0.16 | 30.42±0.21 | 38.47±0.20 | 57.00±0.04 | 26.15±0.14 | 27.49±0.18 | 32.74±0.11 |
| PLM-NR | \( \infty \) | 10 | DP-FedRec | 50.44±0.03 | 21.75±0.06 | 21.40±0.06 | 29.45±0.03 | 50.59±0.01 | 22.61±0.06 | 23.18±0.05 | 28.78±0.06 |
| | 10 | | DP-FedRec | 50.16±0.09 | 21.85±0.07 | 21.57±0.09 | 29.48±0.08 | 50.03±0.04 | 22.17±0.06 | 22.70±0.11 | 28.26±0.10 |
| | 10 | | PrivateRec | 62.62±0.14 | 29.74±0.10 | 31.57±0.11 | 39.66±0.11 | 58.69±0.29 | 26.37±0.11 | 27.75±0.18 | 33.57±0.08 |

Figure 5: Performance of privacy-preserving model serving. \((e_f = \infty \text{ for DP-FedRec and } e_f = 10, B = 5 \text{ for PrivateRec})\)

Figure 6: Effect of privacy amplification by behavior padding.

we get the results \( \hat{R} \). We define \( \tau = 1 - \text{accuracy}(R, \hat{R}) \), of which a smaller value indicates a less utility loss by the noise perturbation.

First, we find in Fig. 7 that reducing \( d \) can improve the privacy-preserving model serving utility for DP-FedRec. This observation validates the claim of our motivation for interest decomposition module (ID) that the method of vanilla DP user embedding faces the dimension curse on utility. It should be noted that with \( e_f = \infty \) and \( e_t = \infty \), a larger user embedding usually brings a more informative representation and has a higher AUC. However, the utility advantage will disappear with privacy-preserving model serving. We show a similar results for Gaussian noise in the Appendix. Then, we observe that even if the FedRec model trained with a smaller \( d \) can slightly improve the private serving utility. It is still inferior to our proposed PrivateRec. Lastly, the skew level \( \tau \) of DP-FedRec is
higher than PrivateRec for various $\epsilon_s$, which indicates the attention $\alpha$ is more robust to the noise perturbation than the user embedding.

Next, we evaluate the influence of the number of basic vectors in Fig. 8 for PrivateRec federated training. We observe that the best number of basic vectors for NewsFeeds and MIND datasets are $B = 3$. If $B$ is too small, the basic vectors might be too coarse-grained and cannot express the personal user interest. If $B$ is too large, the dimension of an attention vector $\alpha$ is large, which results in an increased amount of noise.

4.4.2 Effect of Behavior Padding. Last, we evaluate the effect of privacy amplification by behavior padding in Fig. 6. The motivation for padding the sequence of users’ historical behavior representations is to reduce the amount of noise with the privacy amplification effect by nullification. Thus, in the first sub-figure, we show the required noise perturbation scales for the same serving privacy budget $\epsilon_s$ w.r.t different padding ratios. We can see that the required noise is reduced with a larger padding ratio. Additionally, this effect is more significant when the privacy requirement for serving is stricter (i.e., with a smaller $\epsilon_s$). Generally, $p$ controls the trade-off between the personalized interest and the generalized interest as well as the strength of noise reduction. From the right three sub-figures, we can observe that if $p$ is too small, the noise reduction by the privacy amplification is negligible. If $p$ is too large, the personalized information is erased, which reduces the utility of model serving. Concretely, the best $p$ for PrivateRec is 0.5 when $B = 5$ and 0.2 when $B = 3$. This is reasonable because a smaller $B$ indicates more coarse-grained interest summarization, and the information for each basic vector is more abstract. Thus, for PrivateRec with a smaller $B$, the personalized interest information is more important, and a smaller $p$ is preferred.

5 CONCLUSIONS

In this paper, we propose a differentially private federated news recommendation framework PrivateRec, which can achieve a better utility for training and serving under a formal privacy definition. We first present a universal and formal privacy definition over user’s behavior data for both model training and online serving in the federated recommendation scenario and then propose mechanisms under this definition. To avoid the dimension-dependent noise and improve utility in privacy-preserving online serving, we decompose the high-dimensional and privacy-sensitive user embedding into a combination of public basic vectors and add noise to the combination coefficients. Then the server can reconstruct the user embedding with public basic vectors and perturbed coefficients for recommendation services. In addition, to further reduce noise, we utilize the amplification effect by randomly padding user historical behavior representations. For improving utility in privacy-preserving model training, we avoid the dimension curse for large models via label permutation and differentially private attention modules. Experiments on two real-world news recommendation datasets validate our method’s effectiveness and utility improvement on the model training and serving stage.

SUPPLEMENTARY MATERIALS

Hyperparameter Settings

The hyperparameter settings of our PrivateRec are as follows:

| Hyperparamters | centralized | FL |
|----------------|-------------|----|
| learning rate  | 0.0001      | 0.00005 |
| batch size     | 64          | 128 |
| negative sampling ratio | 4      | 4 |
| Adam beta1     | 0.9         | 0.9 |
| Adam beta2     | 0.99        | 0.99 |

Experimental Environment

Our experiments are conducted on a Linux server with Ubuntu 16.04 operation system, 4 Tesla V100 GPUs with 32GB memory, and Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz with 128GB total memory. We use Python 3.6 and PyTorch DataParallel framework for model training on the 4 GPUs.
