Differentially Private Matrix Factorization with Secure Aggregation and Random Projection

Anonymous Authors[1]

Abstract

Federated learning offers a promising approach to developing privacy-preserving matrix factorization algorithms. By combining secure aggregation (SecAgg) with differential privacy (DP), also referred to as Distributed Differential Privacy (Distributed DP), it is possible to achieve formal privacy guarantees while maintaining a satisfactory level of accuracy. Recently, the Poisson Binomial Mechanism (PBM) has emerged as a state-of-the-art Distributed DP mechanism, which provides an unbiased estimator. However, despite its effectiveness, PBM suffers from increased communication overhead caused by SecAgg. To address this issue, we propose a novel framework called Differentially Private Matrix Factorization with Random Projection (DPMF-RP). This framework integrates PBM with sparse random projection, breaking the dependency between communication costs and parameter sizes. In our approach, users apply sparse random projections to the gradient matrix, reducing its dimensionality before transmission, thereby significantly decreasing communication overhead. Our work is the first to design a differentially private matrix factorization framework that leverages the combination of PBM and random projection. We rigorously analyze how these techniques can be effectively integrated to achieve privacy and efficiency. Empirical studies in two MovieLens datasets demonstrate that our approach has little loss in accuracy with $\epsilon \geq 1$ and $m/p \geq 2$, while reducing the communication overhead by at least 50%.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

In recent years, recommender system has been extensively used in on-line services. Collaborative filtering (CF) is one of the dominant approaches in recommendation systems (Su & Khoshgoftaar, 2009), and matrix factorization (MF) (Koren et al., 2009) is among the most effective techniques in CF algorithms. The success of MF algorithm largely relies on access to users' interaction history. However, directly using these interaction data raises serious privacy concerns.

Federated learning (FL) (McMahan et al., 2016) offers a promising solution to address these privacy issues. In the FL paradigm, user data never leaves their devices, and the model is trained by sharing model parameter updates with the server. To prevent information leakage from transmitted gradients (Chai et al., 2020), differential privacy (DP) (Nguyên et al., 2016; Berlioz et al., 2015) has been incorporated to provide formal privacy guarantees. Despite its effectiveness in preserving privacy, a primary challenge with DP is the trade-off between privacy and utility, often resulting in utility loss in the trained embedding matrix.

Secure aggregation (SecAgg) allows the server to obtain the sum of gradients without inspecting individual user updates (Bonawitz et al., 2017; Bell et al., 2020). The combination of SecAgg and DP, also known as Distributed DP (Goryczka & Xiong, 2015; Kairouz et al., 2021; Stevens et al., 2022), reduces the magnitude of noise added by each user compared with pure local DP. However, most Distributed DP mechanisms adopt additive noises with unbounded support, which requires modular clipping and leads to estimator bias. Recently, (Chen et al., 2022) proposed Poisson Binomial mechanism (PBM), a state-of-the-art mechanism that returns an unbiased estimator in the high privacy setting.

Despite improved privacy-utility trade-offs, SecAgg usually suffers drastically increased communication overheads (Bonawitz et al., 2019). This challenge is exacerbated in MF-based recommendation systems, where the payload increases linearly with item size. Striking a balance between efficiency, privacy, and utility remains a non-trivial task. This leads to a natural question: *can we leverage SecAgg to enhance the utility of MF-based recommendation systems under DP guarantee, while optimizing communication cost?*

Despite its significance, existing research has not adequately addressed these challenges in federated learning (FL)-based recommender systems. To fill in the gap, we propose differentially private matrix factorization with random projection (DPMF-RP), a novel privacy-preserving federated matrix factorization framework. Our framework employs PBM as the distributed DP mechanism to ensure unbiased aggregation and incorporates sparse random projection (William & Lindenstrauss, 1984; Kane & Nelson, 2014) into PBM to reduce communication overhead.

Our rigorous theoretical analysis demonstrates the dual advantages of sparse random projection. Firstly, the technique breaks the dependency between communication cost and item size, resulting in a significantly smaller training payload. More importantly, it enhances the privacy protection level as less information is transmitted to the server. In other words, the privatization error is reduced under the same privacy budget by projecting the matrix into a lower dimension.

Our contributions are as follows:

- We present a differentially private matrix factorization framework enhanced by secure aggregation and sparse random projection. Though these techniques have been studied in a separate manner, little work has rigorously explored their combination to develop a privacy-preserving matrix factorization algorithm.
- We employ PBM, a state-of-the-art mechanism, for federated matrix factorization with Distributed DP guarantee, and perform comprehensive theoretical analysis w.r.t. the privacy guarantee, utility loss, communication, and computation costs when integrating PBM with sparse random projection. To the best of our knowledge, this is the first work to combine PBM with random projection in the context of federated learning.
- We rigorously demonstrate the dual benefits of sparse random projection in the PBM-based Distributed DP mechanism. Besides significantly reducing communication costs, this technique diminishes privatization error, enabling the optimal selection of a projected dimension to balance privatization and projection errors effectively.

2. Literature Review

Differentially private MF MF can be combined with differential privacy in two manners: (i) perturb the model at the server side to ensure that the released item or user profile and recommendation results satisfy differential privacy (Liu et al., 2015; Zhang et al., 2019); (ii) perturb clients' gradients locally so that the server only sees the noisy updates (Minto et al., 2021; Berlioz et al., 2015). The former assumes a trusted server and is referred to as central DP. The latter protects users' information against an untrusted server and relates more to the research scope of this paper.

Multi-party computation in recommender system Secure multi-party computation (MPC) is employed to compute the summation of private gradients without revealing any individual values. Existing work has employed secret sharing and fake marks to protect users' data at the cost of higher communication overhead (Lin et al., 2021). However, SecAgg alone does not provide provable privacy guarantees, as the aggregated gradient still leaks information about individual users (Pasquini et al., 2022; Song & Shmatikov, 2019). (Balu & Furon, 2016) combined DP with third-party-based MPC to provide a privacy guarantee, requiring that the third-party couldn't collude with the recommender. In (Wang et al., 2020), two servers are introduced to store clients' data as additive secrets, and model training was performed between the two servers.

Communication efficient recommender system in FL setting The transmitted model size depends on the number of items, rendering large-scale federated recommendation impractical. Little research has been done to address this challenge. Authors in (Shin et al., 2018) made the first attempt to reduce the communication cost using random projection. (Khan et al., 2021) proposed to transmit the gradients with the use of a multi-armed bandit approach.

There is a dearth of research addressing the trade-off between privacy, utility and communication complexity in federated MF algorithm. Our work fills this gap by integrating PBM with random projection to preserve utility level and optimize communication cost under formal privacy bound.

3. Preliminaries

3.1. Federated Matrix Factorization

Consider a system with n users and m items. The sparse rating matrix $R \in \mathbb{R}^{n \times m}$ is a user-item interaction matrix. Each r_{ij} is an explicit feedback of user i for items j, with $1 \leq i \leq n$ and $1 \leq j \leq m$. MF factorizes R into two low-dimensional latent factor matrices, $U \in \mathbb{R}^{n \times k}$ for users and $V \in \mathbb{R}^{m \times k}$ for items. The linear combination of MF is expressed as

$$R \sim UV^T, \tag{1}$$

such that

$$\widehat{r}_{ij} = u_i v_j^T, \tag{2}$$

with $u_i \in \mathbb{R}^k$ and $v_i \in \mathbb{R}^k$. In sparse matrix R, $r_{ij} = 0$ can be clarified by situations like the item j possibly not being of interest to the user i or the user i being unaware of its existence. To reduce the influence of zero ratings, the

uncertain factor (Hu et al., 2008) is introduced as:

$$c_{ij} = \begin{cases} 1+\kappa & r_{ij} \geq 0 \\ 1 & r_{ij} = 0 \end{cases}, \tag{3}$$

where κ is a positive parameter. The loss function with uncertain factor expresses as:

$$\mathcal{J} = \sum_{(i,j)} c_{ij}(r_{ij} - u_i v_j^T)^2 + \lambda \sum_i \|u_i\|^2 + \mu \sum_j \|v_j\|^2, \tag{4}$$

with regulation factors λ and μ. Alternating Least Squares (ALS) and Stochastic Gradient Descent (SGD) are two common ways involved in minimizing the loss function. We follow (Ammad-Ud-Din et al., 2019) to integrate the two algorithms for federated gradient update.

Users' latent factor matrix U is computed locally using closed form formula under fixed V (Ammad-Ud-Din et al., 2019):

$$\widehat{u}_i = (V^T C^i V + \lambda I)^{-1} R_{u_i}^T C^i V^T, \tag{5}$$

where $C^i \in \mathbb{R}^{m \times m}$ is a diagonal matrix with $C^i_{jj} = c_{ij}$.

Items' latent factor V is updated using SGD approach:

$$v_j = v_j - \gamma_t \frac{\partial J}{\partial v_j}, \tag{6}$$

where γ_t is the learning rate at iteration t, and:

$$\frac{\partial J}{\partial v_i} = -2 \sum_i f(i,j) + 2\mu v_i, \tag{7}$$

where $f(i,j) = c_{ij}(r_{i,j} - u_i v_j^T)u_i$ is computed on each user locally. We adopt the Adaptive Moment Estimation (Adam) method (Kingma & Ba, 2014) for gradient descent.

3.2. Differential Privacy

We introduce the definition of differential privacy (Cormode et al., 2018) and Rényi differential privacy (RDP) (Mironov, 2017).

Definition 3.1 (Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if it satisfies:

$$P(\mathcal{M}(D) \in S) \leq e^{\epsilon} P(\mathcal{M}(D') \in S) + \delta, \tag{8}$$

for all neighboring datasets D and D' and $S \subseteq \text{Range}(\mathcal{M})$.

Definition 3.2 (Rényi Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ is (α, ϵ)-RDP if it satisfies $D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D')) \leq \epsilon$ for all neighboring datasets D and D' and $S \subseteq \text{Range}(\mathcal{M})$, where $D_\alpha(\mathcal{M}(D) \| \mathcal{M}(D'))$ is the Rényi divergence defined as:

$$D_\alpha(P \| Q) \triangleq \frac{1}{\alpha} \log \left(\mathbb{E}_Q \left[\left(\frac{P(X)}{Q(x)} \right)^\alpha \right] \right). \tag{9}$$

Note that if M is an $(\alpha, \epsilon(\alpha))$-RDP mechanism, it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-differential privacy for any $0 < \delta < 1$.

Another related concept is l_p-sensitivity, capturing the magnitude by which an entry can change the output of function f in the worst case (Dwork et al., 2014).

Definition 3.3. Let the mechanism $f(R) : \text{Range}(R) \to \mathbb{R}^d$ operates on a dataset R. The l_p-sensitivity of f is:

$$\Delta_p(f) = \max_{R^A, R^B} \|f(R^A) - f(R^B)\|_p, \tag{10}$$

where R^A and R^B are any pair of neighboring datasets that differ in at most one entry, and $\| \cdot \|_p$ denotes the l_p-norm of the inner vector.

Remark 3.4. For mechanism $f(R) : \text{Range}(R) \to \mathbb{R}^{n \times k}$, its l_p-sensitivity is defined by:

$$\Delta_p(f) = \max_{R^A, R^B} \|f(R^A) - f(R^B)\|_{p,p}, \tag{11}$$

where $\| \cdot \|_{p,p}$ denotes the $l_{p,p}$-norm (Ding et al., 2006) of inner matrix:

$$\|A\|_{p,p} = \left(\sum_i \sum_j |a_{ij}|^p \right)^{1/p}. \tag{12}$$

3.3. Sparse Random Projection

We state the definition of sparse random projection matrix S.

Definition 3.5. For $p \ll m$, let $S \in \mathbb{R}^{p \times m}$ be a sparse random projection matrix with each entry given by

$$S_{ij} = \sigma(j) \cdot 1_{\{h(j)=i\}}, \tag{13}$$

where $\sigma : [m] \to \{-1, +1\}$ and $h : [m] \to [p]$ are two independent hash functions.

The matrix is sparse in the sense that each column in S contains exactly one non-zero element.

For a vector $v \in \mathbb{R}^m$, the random projection matrix allows to reduce the dimension from n to p via the transformation $w = Sv$. The original vector can be approximately reconstructed using $\hat{v} = S^T w$.

3.4. The Poisson Binomial Mechanism

The Poisson binomial mechanism (PBM) (Chen et al., 2022) is a discrete differential privacy mechanism that achieves unbiased estimator with the same utility trade-offs as the continuous Gaussian mechanism. The protocol mainly consists of three steps: (a) transform the l_2 geometry of the vector into l_∞ via Kashin's representation, (b) each client applies scalar Poisson binomial mechanism (Algorithm 1) on each coordinate of the vector, and (c) server aggregates and reconstructs the sum.

Algorithm 1 The Scalar Poisson Binomial Mechanism (ScalarPBM)

Input: clipping threshold $c > 0$; private number $x \in [-c, +c]$; scaling factor $\theta \in [0, \frac{1}{4}]$; quantization count $q \in \mathbb{N}$
Output: privatized number z
Re-scaling x: $p = \frac{\theta}{c}x + \frac{1}{2}$
Privatization by sampling from binomial distribution $z = \text{Binom}(q, p)$.
Return z

3.5. Secure Aggregation

To prevent the server from accessing individual user gradients, our framework employs the secure aggregation (SecAgg) protocol proposed by Bonawitz et al. (Bonawitz et al., 2017). The main idea of the protocol is that the pairwise masks established between clients will be canceled out once the masked gradients are summed up at the server. Specifically, each client i: (1) negotiates shared randomness seed s_{ij} with every other clients $j \in [n/i]$; (2) derives pairwise masks $\mathbf{m}_{ij} = F(s_{ij})$, which are added to their local gradients $\mathbf{x}_i$:

$$\mathbf{y}_i = \mathbf{x}_i - \sum_{j<i} \mathbf{m}_{ij} + \sum_{j>i} \mathbf{m}_{ij}. \tag{14}$$

The masked gradient $\mathbf{y}_i$ is sent to server for aggregation:

$$\sum_i \mathbf{y}_i = \sum_i \left(\mathbf{x}_i - \sum_{j<i} \mathbf{m}_{ij} + \sum_{j>i} \mathbf{m}_{ij} \right) = \sum_i \mathbf{x}_i. \tag{15}$$

4. Proposed Method

We assume an honest-but-curious (Yang et al., 2019) recommender and users are unwilling to share their interaction data with the recommender. Our system operates in a federated learning setting where users compute their local parameters and transmit their privatized updates to the recommender. The recommender then updates the shared parameters using the aggregated updates. The overall framework of our method is visualized in Figure 1.

4.1. Differentially Private Matrix Factorization

Our proposed algorithm, DPMF-RP, builds on the federated implementation introduced in Section 3.1. We slightly modify the computation of $f(i, j)$ in Equation 7 to preserve l_1 sensitivity. We clip the rating error within $[-e, e]$.

$$\begin{aligned} e_{ij} &= (r_{ij} - u_i v_j^T), \\ \hat{e}_{ij} &= \max\{-e, \min\{e_{ij}, e\}\}, \\ f(i, j) &= c_{ij} \hat{e}_{ij} u_i. \end{aligned} \tag{16}$$

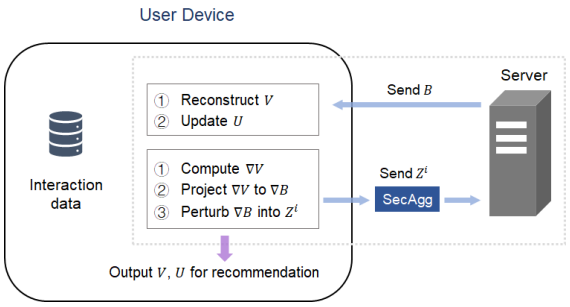

Figure 1. Overall framework of DPMF-RP

User i's latent factor is clipped to a maximum l_1-norm of B_1:

$$\hat{u}_i = u_i \cdot \min\{1, \frac{B_1}{\|u_i\|_1}\}. \tag{17}$$

The resulting l_1 sensitivity of the gradient matrix ∇V is bounded by $\Delta_1 = 4e(1 + \kappa)B_1/\beta$ (see Lemma **??**).

Algorithm 2 outlines the federated learning implementation for our protocol, which consists of three main components:

- Sparse random projection: each participating user i computes a compressed version of item embedding gradients ∇B^i by sparse random projection $\nabla B^i = S\nabla V^i$. The server updates and maintains the compressed item embedding matrix B.
- Poisson binomial mechanism: the projected gradient matrix ∇B^i is privatized by Poisson binomial mechanism. Initially, we transform the l_1 geometry of ∇B^i into l_∞ via Algorithm 3. Following this transformation, we apply scaler PBM on each coordinate to obtain a noisy form of discrete representation Z^i.
- SecAgg: we utilize the SecAgg outlined in Section 3.5 to compute the modular sum of Z^i. The communication cost of the SecAgg protocol scales linearly with the number of users and items.

4.2. Security and Privacy Analysis

4.2.1. Rényi Differential Privacy (RDP) Guarantee

We summarize the privacy guarantee in the following theorem.

Theorem 4.1. *For any $\alpha > 1$, Algorithm 2 satisfies (α, ϵ)-RDP with:*

$$\epsilon = C_0 \frac{\theta^2}{(1 - 2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1} \right) \frac{qpk}{\beta n}, \tag{18}$$

for some large enough C_0.

Algorithm 2 Differentially Private MF

Input: rating matrix $R \in \mathbb{R}^{n \times m}$
Output: user latent factor $U \in \mathbb{R}^{n \times k}$; item latent factor $V \in \mathbb{R}^{m \times k}$
Initialize S, U and V
$B = SV$
for $t = 1, 2, ..., T$ **do**
  Server randomly selects β of users as sample $\mathcal{A}_t(u)$
  for user $i \in \mathcal{A}_t(u)$ **do**
    Receive B from server
    Compute $V = S^T B$
    Update u_i using equation 5
    Clip $u_i = u_i \cdot \min\{1, \frac{B_1}{\|u_i\|_1}\}$
    Compute item latent factor gradients matrix ∇V^i, with j^{th} row be $f(i, j)$ computed from equation 16
    Compute $\nabla B^i = S \nabla V^i$
    Flatten the matrix $\nabla \tilde{B}^i \leftarrow \text{FlatMat}(\nabla B^i)$.
    for each element of $\nabla \tilde{B}^i$ **do**
      $Z^i_{kl} \leftarrow \text{ScalarPBM}\ (\frac{\Delta_1}{\sqrt{p'}}, \nabla \tilde{B}^i_{kl}, \theta, q)$
    end for
    Send Z^i to the server via SecAgg.
  end for
  Server computes via SecAgg the aggregation $\nabla \tilde{B} = \frac{\Delta_1}{\sqrt{p'} q \theta |\mathcal{A}_t(u)|} \left(\sum_{i \in \mathcal{A}_t(u)} Z^i - \frac{q|\mathcal{A}_t(u)|}{2} \right)$
  Server unflattens the matrix $\nabla B \leftarrow \text{RevMat}\ (\nabla \tilde{B}, p)$
  Server updates B using ADAM
end for
Return $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{m \times k}$

Proof. We initially leverage the sensitivity bound of ∇B to guarantee that each $|Z^i_{jl}|$ is upper bounded by $\Delta_1/\sqrt{p}$. Let ∇V and $\nabla V'$ be any two $m \times k$ matrices with l_1 sensitivity bound $\|\nabla V - \nabla V'\|_1 \leq \Delta_1$. Lemma 4.2 states that applying the sparse random projection preserves the l_1 sensitivity.

Lemma 4.2. *For sparse matrix S defined in Definition 3.5, we have $\|S\nabla V - S\nabla V'\|_{1,1} \leq \Delta_1$.*

The privacy guarantee for the scalar version of PBM with one dimension is established by Lemma 4.3.

Lemma 4.3. *Algorithm 1 with SecAgg satisfies (α, ϵ)-RDP for any $\alpha > 1$ and:*

$$\epsilon = C_0 \frac{\theta^2}{(1-2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1} \right) \frac{q}{\beta n}, \tag{19}$$

where C_0 is some large enough universal constant.

To achieve the privacy bound for the multi-dimensional PBM, we apply the composition theorem of RDP (Mironov, 2017) and obtain the privacy guarantee:

$$\epsilon = \sum_k \sum_l \epsilon_{kl} = C_0 \frac{\theta^2}{(1-2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1} \right) \frac{qpk}{\beta n}. \tag{20}$$

□

4.2.2. Stronger privacy protection

An important observation in Theorem 4.1 is that sparse random projection enhances the RDP privacy guarantee, especially when the gradient matrix is projected to smaller size. Theorem 4.1 reveals that a smaller value of p leads to a higher level of privacy protection, given the same set of privatization parameters. Furthermore, projecting the item gradients to a $p \times k$ matrix effectively decreases the privacy budget ϵ compared to the case where random projection is not applied (see Appendix C.3). From the noise perspective, we will show in Section 5.1 that the privatization error remains unchanged regardless of the projected dimension. In other words, random projection can reduce the privatization noise required to achieve the same privacy guarantee, as fewer information is transmitted to the server.

4.2.3. Security against server colluding with clients

We begin with the scenario where both the server and clients are curious-but-honest. The SecAgg protocol ensures that the joint view of the server and $t \leq \lceil \beta n/3 \rceil - 1$ users reveals no information about individual gradients, except the aggregation results (Bonawitz et al., 2017). In other words, the SecAgg protocol allows the server to collude with up to $\lceil \beta n/3 \rceil - 1$ clients without compromising privacy.

The SecAgg protocol also preserves privacy against malicious server that conducts active inference attack by sending falsified message to the clients. The signature scheme enables clients to verify message integrity, ensuring that messages are not tampered by the server (Bonawitz et al., 2017). The collusion between a malicious server and up to $\lceil \beta n/3 \rceil - 1$ clients does not reveal any information about individual updates.

4.3. Communication and Computation Analysis

In this section, we analyze the communication and computation overheads. Table 1 presents a comparison of the computation cost among DPMF-RP, the federated MF in non-private and pure SecAgg setting. For the non-private setting, each user directly uploads their plaintext gradients to the server. The pure SecAgg setting employs SecAgg to protect individual gradients while excluding PBM and random projection. b is the number of bits required to represent a single numerical value.

4.3.1. Choice of θ and q

To minimize the communication cost, we would like to pick q as small as possible. The following steps describe how to determine q and θ under given ϵ and α for optimal communication efficiency:

- Compute θ using expression 18 using $q = 1$, and clip θ to the maximum value of $1/4$.
- If $\theta = 1/4$, calculate q according to expression 18. Otherwise $q = 1$.

The mechanism ensures that q is upper bounded by:

$$q \leq \left(\frac{\alpha - 1}{\alpha^2}\right) \frac{\beta n \epsilon}{C_0 p k}. \tag{21}$$

4.3.2. Communication Cost

We analyze the communication overhead in terms of message size each user uploads to the server. The per iteration communication cost is $\mathcal{O}(pk \log q + \beta n)$ for each user. Plugging in the upper bound of q, the communication cost can be expressed as $\mathcal{O}(pk \log \frac{\beta n \epsilon}{\alpha p k} + \beta n b)$. Then the server communication cost is given by $\mathcal{O}(\beta n p k \log \frac{\beta n \epsilon}{\alpha p k} + (\beta n b)^2)$.

Table 1 shows that the communication overheads are optimized compared with the pure SecAgg setting, given that $p < m$ and $q < b$. While the SecAgg protocol introduces an additional communication cost of $\beta n b$ compared to the non-private setting, DPMF-RP maintains an advantage when p and q are sufficiently small.

4.3.3. Computation Cost

The user's computation cost can be broken as: (1) reconstruct V from B ($O(mk)$ complexity); (2) update user latent factor u_i ($O(mk^2 + k^3)$ complexity); (3) obtain the raw reduced gradient matrix ∇B^i ($O(mk)$ complexity); (4) flatten the reduced gradient matrix ($O(p^2 k)$ complexity); (5) compute the privatized output Z^i ($O(pk)$ complexity); (6) operations related to SecAgg ($O(pk + \beta^2 n^2)$ complexity) Therefore, the user's computation cost adds up to $O(mk^2 + k^3 + p^2 k + \beta^2 n^2)$.

The server's computation cost can be broken as: (1) aggregate gradients from participating users ($O(\beta n p k)$ complexity); (2) unflatten the aggregation matrix ($O(p^2 k)$ complexity); (3) operations related to SecAgg ($O(pk\beta^2 n^2)$ complexity). Hence, the server computation overhead is $O(pk\beta^2 n^2 + p^2 k)$.

The user's extra overhead $O(p^2 k)$ in our algorithm comes primarily from PBM related operation-matrix flatten. On the other hand, we allow the server to improve the computation efficiency over the pure SecAgg setting since it operates on a reduced matrix.

5. Utility Analysis

5.1. Error Bound on Gradients

We examine the deviation between $S^T \nabla (B^{(t)})'$ and $\nabla V^{(t)}$ measured by MSE, which is decomposed as:

$$\mathbb{E}\left[\|S^T \nabla B' - \nabla V\|_F^2\right] \leq \underbrace{\mathbb{E}\left[\|S^T S \nabla V - \nabla V\|_F^2\right]}_{\text{projection error}} + \underbrace{\mathbb{E}\left[\|S^T (\nabla B' - \nabla B)\|_F^2\right]}_{\text{privatization error}}, \tag{22}$$

where $\mathbb{E}[\cdot]$ takes the expectation over Φ and L, and $\|\cdot\|_F$ denotes the Frobenius norm of the inner matrix.

We start by bounding the projection error.

Lemma 5.1. *Given any $\nabla V \in \mathbb{R}^{m \times k}$ and $S \in \{-1, 0, 1\}^{p \times m}$ defined in Definition 3.5, we have:*

$$\mathbb{E}\left[\|S^T S \nabla V - \nabla V\|_F^2\right] \leq \frac{2m}{p} \|\nabla V\|_F^2. \tag{23}$$

Next, we establish a bound for the privatization error as follows:

Lemma 5.2. *Given any $\nabla B'$ computed via Algorithm 2, and $S \in \{-1, 0, 1\}^{p \times m}$ defined in Definition 3.5, it holds that:*

$$\mathbb{E}\left[\|S^T (\nabla B' - \nabla B)\|_F^2\right] \leq \frac{mk\Delta_1^2}{4\beta n q \theta^2}, \tag{24}$$

where ∇B represents the gradient in non-private setting.

Finally, we obtain the total error bound by summing up the projection and privatization errors:

Theorem 5.3. *Given any $\nabla B'$ computed via Algorithm 2, and $S \in \{-1, 0, 1\}^{p \times m}$ defined in Definition 3.5, it holds that:*

$$\mathbb{E}\left[\|S^T \nabla B' - \nabla V\|_F^2\right] \leq \frac{2m}{p} \|\nabla V\|_F^2 + \frac{mk\Delta_1^2}{4\beta n q \theta^2}. \tag{25}$$

5.2. Optimal Choice of Projected Dimension

In this section, we examine how to determine the projected dimension p to optimize the error bound on gradients. By referring to equation 18 and 25, we observe that a decrease in the value of p leads to an increase in the error bound measured by mean square error (MSE), along with a reduction in the privacy budget ϵ. We establish theorem 5.4 to balance the trade-off between MSE and privacy budget:

Theorem 5.4. *For any $\alpha > 1$ and $\epsilon > 1$, by picking $p = \Theta\left(\frac{n\beta\sqrt{\beta mn\epsilon}}{k\sqrt{\alpha}}\right)$, the MSE in equation 25 is minimized as $\Theta\left(mk\Delta_1^2\sqrt{\alpha m}/\sqrt{\beta n\epsilon}\right)$.*

The optimal selection of p depends on the balance between projection and privatization errors. When privatization error dominates, such as under higher privacy demands or with a small user population, the matrix can be projected to a lower dimension to minimize the error bound.

Table 1. Computation and communication cost of DPMF-RP and FL in non-private & SecAgg settings.

	Server Communication Cost	User Communication Cost	Server Computation Cost	User Computation Cost
Non-Private	$\mathcal{O}(\beta nmkb)$	$\mathcal{O}(mkb)$	$O(\beta nmk)$	$O(mk^2+k^3)$
Pure SecAgg	$\mathcal{O}\left(\beta nmkb+(\beta nb)^2\right)$	$\mathcal{O}(mkb+\beta nb)$	$O(mk\beta^2n^2)$	$O\left(mk^2+k^3+(\beta n)^2\right)$
DPMF-RP	$\mathcal{O}\left(\beta npk\log\frac{\beta n\epsilon}{\alpha pk}+(\beta nb)^2\right)$	$\mathcal{O}(pk\log\frac{\beta n\epsilon}{\alpha pk}+\beta nb)$	$O(pk\beta^2n^2+p^2k)$	$O\left(mk^2+k^3+p^2k+(\beta n)^2\right)$

5.3. Convergence Analysis

We analyze the convergence rate of Algorithm 2 under the choice of p in Section 5.2. To complete the analysis, we further need the following assumption:

Assumption 5.5. The objective $\mathcal{J}=f(x)$ is a Lipschitz continuous with regards to V:

$$\|f(V,U)-f(V',U)\|\leq L_f\|V-V'\|, \tag{26}$$

for any V, V' under any fixed U.

Our algorithm guarantees gradient smoothness with respect to the item latent factor V:

Lemma 5.6. *There exist a constant L_g, such that the objective $\mathcal{J}=f(x)$ satisfies:*

$$\|\nabla_V f(V,U)-\nabla_V f(V',U)\|\leq L_g\|V-V'\|, \tag{27}$$

for any item latent factors V and V'.

Based on the aforementioned assumption and lemma, we can establish the convergence theorem below:

Theorem 5.7. *Under Assumption 5.5, after T iterations, the output of Algorithm 2 satisfies:*

$$\begin{aligned}&\min_t\mathbb{E}\left[\|\nabla f(V^t,U^t)\|^2\right]\\&\leq\mathcal{O}\left(\frac{L_g(mk\Delta_1^2\sqrt{\alpha}/\sqrt{\beta\epsilon}+L_f^2)}{\log T}\right),\end{aligned} \tag{28}$$

by choosing the learning $\gamma^t=\Theta(\frac{1}{t})$ and projected dimension $p=\Theta\left(mn\beta\sqrt{\beta\epsilon}/(k\sqrt{\alpha})\right)$.

Remark 5.8. By substituting $L_g\leq\mu+kB_1^2(1+\kappa)$ (see Appendix F), the bound is approximately linear in the item regularization coefficient μ. Noted that the actual value of L_g depends on the dataset's specific distribution and can be significantly smaller than this upper bound.

6. Experiment

6.1. Dataset and Experiment setting

6.1.1. Dataset

The experiment is performed on three datasets with varying sparsities[1] [2]: a) the 100k MovieLens dataset (ML100K) with ratings from 943 users for 1,682 movies, b) half of the 10M MovieLens dataset (ML5M) including ratings from 34,939 users for 5,338 movies, and c) a subset of the Yelp dataset (Yelp) with ratings from 10,000 users and 30,000 restaurants.

[1]https://grouplens.org/datasets/movielens/

[2]https://www.yelp.com/dataset

6.1.2. Evaluation metrics

The performance are measured using five metrics: Root Mean Square Error (RMSE), accuracy, recall, F1-score, and mean Average Precision (mAP). Refer to appendix H for the definitions.

6.1.3. Hyper-parameter and other settings

We set $\lambda=0.25$,$\mu=0.9$, $\beta_1=0.9$, $\beta_2=0.8$, $\gamma=0.05$, $B_1=0.001$, $\kappa=15$ for all datasets. Additional settings include: $k=14$, $e=1.5$ for ML5M; $k=14$, $e=1$ for ML100K; $k=20$, $e=1.5$ for Yelp. The parameter p is considered as a constant ratio of m, with m/p ranging from 1 to 10. Unless otherwise specified, $\beta=0.1$ of users would participate in each iteration. We consider privacy budget for the entire training process, with ϵ ranging from 0.1 to 10 for $\alpha=2$. Each reported value is taken as the average of four experiments.

6.2. Experiment Results

6.2.1. Comparison with Centralized MF

We benchmark our proposed algorithm DPMF-RP under $\epsilon=1$ against Matrix Factorization with Alternating Least Squares update (CentralMF), with performance reported in table 2.

In Table 2 for ML100K, the values of RMSE and mAP in the central setting are similar to those for Algorithm 2. The proposed federated model possesses higher recall, comparing with CentralMF model, at the cost of reduced its recommendation precision and thus F1 values. Moreover, the central model provides higher precision and F1 values. Similar patterns are presented in the measured metrics of the experiments employing ML5M and Yelp datasets.

6.2.2. Selection of p

In Figure 2, we investigate how the privacy budget ϵ and the participation rate β influence projection and privatization errors, thus affecting the optimal choice of p. The two types

Table 2. Performance for centralized and federated setting.

	ML100K	ML5M	Yelp
	CentralMF		
RMSE	0.9968±0.0108	0.9007±0.0021	1.0008±0.0018
Precision	0.2776±0.0048	0.2955±0.0023	0.0414±0.0007
Recall	0.2418±0.0040	0.2013±0.0022	0.0826±0.0010
F1	0.2451±0.0041	0.2394±0.0023	0.0571±0.0008
mAP	0.1395±0.0018	0.1178±0.0027	0.0299±0.0006
	DPMF-RP		
RMSE	1.0083±0.0024	0.9459±0.0019	1.0355±0.0044
Precision	0.1965±0.0103	0.1822±0.0023	0.0379±0.0011
Recall	0.2472±0.0125	0.2329±0.0047	0.0869±0.0029
F1	0.2189±0.0112	0.2045±0.0026	0.0528±0.0016
mAP	0.1158±0.0095	0.1150±0.0036	0.0276±0.0010

of errors are computed by ablation studies in RMSE under $m/p = 3$. Specifically, projection error is the average of: a) case 4 minus case 1, and b) DPMF-RP minus case 3. Similarly, the privatization error is the average of: a) case 3 minus case 1, and b) DPMF-RP minus case 4.

Under $\epsilon = 1$ and $\beta = 0.1$, privatization error is slightly less than the projection error, comprising 43% of the total error, and the RMSE is minimized around $m/p = 3$. When the privacy budget ϵ decreases to 0.01, the proportion of privatization error is reduced to around 45%, shifting the optimal m/p to around 4. A similar pattern is observed for $\epsilon = 1$ and $\beta = 0.01$. The highest proportion of privatization error, 62%, occurs with $\epsilon = 0.01$ and $\beta = 0.01$, where the optimal m/p increases to approximately 6.

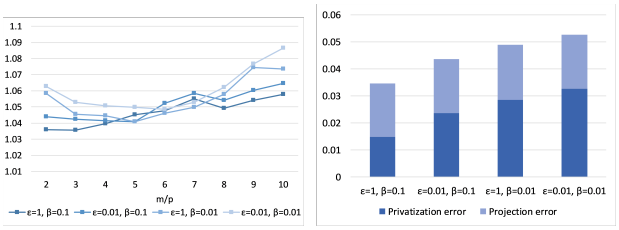

Figure 2. Impact of projected dimension m/p (left) and error analysis (right) under various privacy budget ϵ and participation rate β on Yelp dataset.

6.3. Hyperparameter Analysis

Figure 4 presents the impact of different hyperparameters on F1 and mAP for ML5M and ML100K datasets. The parameter B_1 is used to clip the latent factor matrix U and calculate δ_1 in Algorithm 2. A higher value of B_1 results in smaller clipping error while greater loss in the privatization step. As the parameter value increases before reaching 0.001, the values of F1 and mAP for both datasets gradually increase. The values of F1 and mAP slightly decrease after $B_1 = 0.001$. The result indicates that an optimal choice of B_1 could be around 0.001.

6.3.1. Communication Cost

Figure 3 shows the per user communication cost in realistic experiments. As shown in Figure **??**, with a constant dimension reduction ratio m/p, the communication cost increases linearly with the latent factor size k. Figure **??** demonstrates that when item size m increases, the communication cost rises at varying rates depending on the dimension reduction ratio. Higher dimension reduction ratio m/p results in less expensive communication costs. For $m/p = 2$ and $m/p = 3$ the payload is approximately 50% and 34% of the case without random projection.

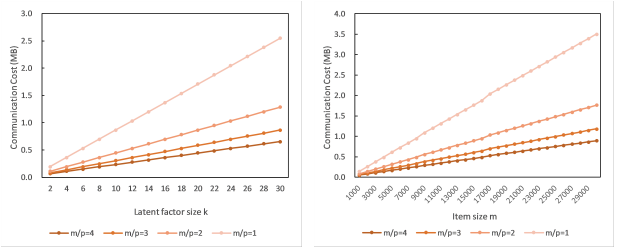

Figure 3. Communication cost by latent factor k (left) and item size m (right) under user size $n = 10000$ and participation rate $\beta = 0.1$.

7. Conclusion and Future Work

This paper proposes DPMF-RP, a differentially private matrix factorization framework enhanced by secure aggregation and random projection. Each user applies random projection on the item gradient matrix to reduce communication overhead. The projected gradient matrices are summed by SecAgg which lowers the magnitude of noises added to the gradients. We employ PBM, a state-of-art Distributed DP mechanism, that returns unbiased estimator and results in lower communication cost in high-privacy regime. We compute a theoretical error bound on the perturbed gradient matrix and derive the choice of p that achieves optimal tradeoff between privacy budget and MSE. Empirical studies show that: (i) our approach suffers little loss in accuracy with $\epsilon \geq 1$ and $n/p \geq 2$, saving the communication overhead by at least 50%; (ii) SecAgg drastically improves the model performance with lower noise levels.

Our work opens up research opportunities for the following directions. Firstly, existing literature focuses on the linear random projection, while few research has studied the nonlinear dimension reduction algorithms. Integrating nonlinear random projection with Distributed DP mechanism could produce better utility guarantees. Secondly, it is interesting to evaluate our framework on other applications such as image recognition and natural language understanding.

References

Ammad-Ud-Din, M., Ivannikova, E., Khan, S. A., Oyomno, W., Fu, Q., Tan, K. E., and Flanagan, A. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.

Balu, R. and Furon, T. Differentially private matrix factorization using sketching techniques. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pp. 57–62, 2016.

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1253–1269, 2020.

Berlioz, A., Friedman, A., Kaafar, M. A., Boreli, R., and Berkovsky, S. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 107–114, 2015.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019.

Chai, D., Wang, L., Chen, K., and Yang, Q. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5):11–20, 2020.

Chen, W.-N., Ozgur, A., and Kairouz, P. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 3490–3506. PMLR, 2022.

Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., and Wang, T. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 1655–1658, 2018.

Ding, C., Zhou, D., He, X., and Zha, H. R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pp. 281–288, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Goryczka, S. and Xiong, L. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE transactions on dependable and secure computing*, 14(5):463–477, 2015.

Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pp. 263–272. Ieee, 2008.

Kairouz, P., Liu, Z., and Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 5201–5212. PMLR, 2021.

Kane, D. M. and Nelson, J. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.

Khan, F. K., Flanagan, A., Tan, K. E., Alamgir, Z., and Ammad-Ud-Din, M. A payload optimization method for federated recommender systems. In *Fifteenth ACM Conference on Recommender Systems*, pp. 432–442, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Lin, Z., Pan, W., and Ming, Z. Fr-fmss: federated recommendation via fake marks and secret sharing. In *Fifteenth ACM Conference on Recommender Systems*, pp. 668–673, 2021.

Liu, Z., Wang, Y.-X., and Smola, A. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 171–178, 2015.

McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2, 2016.

Minto, L., Haller, M., Livshits, B., and Haddadi, H. Stronger privacy for federated collaborative filtering with implicit feedback. In *Fifteenth ACM Conference on Recommender Systems*, pp. 342–350, 2021.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H., and Shin, J. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.

Pasquini, D., Francati, D., and Ateniese, G. Eluding secure aggregation in federated learning via model inconsistency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2429–2443, 2022.

Rastegarpanah, B., Gummadi, K. P., and Crovella, M. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 231–239, 2019.

Shin, H., Kim, S., Shin, J., and Xiao, X. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1770–1782, 2018.

Song, C. and Shmatikov, V. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–206, 2019.

Stevens, T., Skalka, C., Vincent, C., Ring, J., Clark, S., and Near, J. Efficient differentially private secure aggregation for federated learning via hardness of learning with errors. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1379–1395, 2022.

Su, X. and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

Wang, L., Huang, Z., Pei, Q., and Wang, S. Federated cf: Privacy-preserving collaborative filtering cross multiple datasets. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6. IEEE, 2020.

William, B. J. and Lindenstrauss, J. Extensions of lipschitz mapping into hilbert space. *Contemporary mathematics*, 26(189-206):323–341, 1984.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Zhang, S., Liu, L., Chen, Z., and Zhong, H. Probabilistic matrix factorization with personalized differential privacy. *Knowledge-Based Systems*, 183:104864, 2019.

A. Notations

Table 3 lists the notations and their descriptions used throughout this paper.

Table 3. Notations used in the paper.

Notation	Description	Notation	Description
n	The number of users	m	The number of items
k	Dimension of latent factor	l_x	Dimension of user attributes
l_y	Dimension of item attributes	$R,\ r_{i,j}, \hat{r}_{i,j}$	Rating matrix
$U,\ u_i$	User latent factor	$V,\ v_j$	Item latent factor
c, κ	Uncertainty coefficient and parameter	α, ϵ	Privacy parameter
λ, μ	Regularization coefficient	S	Sparse random projection matrix
p	Dimension of compressed gradient matrix after random projection	B	Compressed matrix for V
∇V	Gradient matrix of V	∇B	Compressed gradient matrix of V
B_1	l_1-norm of user latent factor	Δ_1	l_1-sensitivity of gradient matrix ∇V
Z^i	Privatized gradient	β	Proportion of participated users
q	Quantization count in PBM	θ	Scaling factor in PBM
C_0	Sufficiently large universal constant	$[d]$	Set of integers $\{1, 2, ..., d\}$

B. Algorithms

Algorithms 3, and 4 presents the auxiliary algorithms for DPMF-RP.

Algorithm 3 Flatten matrix (FlatMat)

Input: compressed gradient matrix $B \in \mathbb{R}^{p\times k}$
Output: flattened gradient matrix $\tilde{B} \in \mathbb{R}^{p'\times k}$
Select p' be the smallest number of power 2 no less than p, and construct B' by padding B with $p' - p$ rows of 0s.
Flatten the padded matrix $\tilde{B} = HD_\xi B'$, where $H \in \{-1/\sqrt{p'}, +1/\sqrt{p'}\}^{p'\times p'}$ is a Walsh-Hadamard matrix satisfying $H^T H = I$ and $D_\xi \in \{-1, 0, +1\}^{p'\times p'}$ is a diagonal matrix with $\xi \in \{-1, +1\}$ on the diagonal.
Return $\tilde{B} \in \mathbb{R}^{p'\times k}$ that satisfies $\|\tilde{B}\|_\infty \leq \frac{\Delta_1}{\sqrt{p'}}$

Algorithm 4 Reverse matrix (RevMat)

Input: flattened matrix $\tilde{B} \in \mathbb{R}^{p'\times k}$; dimension after projection p
Output: compressed matrix $B \in \mathbb{R}^{p\times k}$
Reverse the matrix $B' = D_\xi H^T \tilde{B}$
Construct B by selecting the first p rows in B'.
Return $B \in \mathbb{R}^{p\times k}$

C. Proof of Privacy Bound

C.1. Proof of Lemma 4.2

The element in the matrix is computed by:

$$\begin{aligned}(S\nabla V - S\nabla V')_{ij} &= \langle S_i, \nabla V_j\rangle - \langle S_i, \nabla V'_j\rangle \\ &= \sum_{h=1}^{n} S_{ih}(\nabla V_{hj} - \nabla V'_{hj}).\end{aligned} \tag{29}$$

Then the $l_{1,1}$-norm is bounded by:

$$\begin{aligned}\sum_i \sum_j |S\nabla V - S\nabla V'|_{ij} &= \sum_i \sum_j |\sum_{h=1}^{n} S_{ih}(\nabla V_{hj} \\ -\nabla V'_{hj})| &\leq \sum_i \sum_j \sum_h |S_{ih}(\nabla V_{hj} - \nabla V'_{hj})| \\ &= \sum_j \sum_h (\sum_i |S_{ih}(\nabla V_{hj} - \nabla V'_{hj})|) \\ &= \sum_j \sum_h |\nabla V_{hj} - \nabla V'_{hj}| = \Delta_1.\end{aligned} \tag{30}$$

C.2. Proof of Lemma 4.3

The proof follows by: (a) decompose z into Bernoulli distribution, (b) bound the Rényi divergence with maximum achieved at extreme points, (c) remove common parts of random variables, and (d) bound the Rényi divergence with moment generating function. Refer to (Chen et al., 2022) for the detailed proof.

C.3. Privacy Bound without Random Projection

In the following, we provide the privacy guarantee for the case without sparse random projection.

Proposition C.1. *For any $\alpha > 1$, Algorithm 2 without sparse random projection satisfies (α, ϵ)-RDP with:*

$$\epsilon = C_0 \frac{\theta^2}{(1-2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1}\right) \frac{qmk}{\beta n}, \tag{31}$$

for some large enough C_0.

Proof. In this case, each user directly transmits the privatized version of ∇V, instead of ∇B, to the server. Let ∇V and $\nabla V'$ be any two $m \times k$ matrices with l_1 sensitivity bound $\|\nabla V - \nabla V'\|_1 \leq \Delta_1$.

Lemma 4.3 states that the scalar version of PBM with one dimension satisfies (α, ϵ)-RDP for any $\alpha > 1$ and:

$$\epsilon = C_0 \frac{\theta^2}{(1-2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1}\right) \frac{q}{\beta n}. \tag{32}$$

Then we can obtain the privacy guarantee for the multi-dimensional PBM:

$$\epsilon = \sum_k \sum_l \epsilon_{kl} = C_0 \frac{\theta^2}{(1-2\theta)^4} \left(\frac{\alpha^2}{\alpha - 1}\right) \frac{qmk}{\beta n}. \tag{33}$$

□

D. Proof of Lemma 5.2

Proof. Given $\sigma^2 = \mathbb{E}\left[(\nabla B'_{ij} - \nabla B_{ij})^2\right]$ for any i, j, it holds that:

$$\begin{aligned}\mathbb{E}\left[\|S^T(\nabla B' - \nabla B)\|_F^2\right] &= \sum_{i=1}^{k} \sum_{j=1}^{m} \mathbb{E}\left[\left(\sum_{l=1}^{p} S_{lj}(\delta_{li})\right)^2\right] \\ &= \sum_{i=1}^{k} \sum_{j=1}^{m} \left[\sum_{l=1}^{p} \mathbb{E}[S_{lj}^2 \delta_{li}^2] + 2 \sum_{l \neq h} \mathbb{E}[S_{lj}\delta_{li} S_{hj} \delta_{hi}]\right] \\ &= \sum_{i=1}^{k} \sum_{j=1}^{m} \sum_{k=1}^{p} \frac{\sigma^2}{p} = mk\sigma^2,\end{aligned} \tag{34}$$

where $\delta_{ki} = \nabla B'_{ki} - \nabla B_{ki}$ for all k, i.

We proceed to bounding σ:

$$\begin{aligned}\mathbb{E}\left[\|\nabla B' - \nabla B\|_F^2\right] &= \mathbb{E}\left[\|U^T\nabla\tilde{B}' - U^TU\nabla B\|_F^2\right] \\ &\overset{(a)}{\leq} p'\mathbb{E}\left[\|\nabla\tilde{B}' - U\nabla B\|_F^2\right] \\ &= \frac{\Delta_1^2}{\beta^2n^2q^2\theta^2}\cdot\sum_{i\in\mathcal{A}_t(u)}\mathbb{E}\left[\|Z^i - \mathbb{E}[Z^i]\|_F^2\right] \\ &= \frac{\Delta_1^2}{\beta^2n^2q^2\theta^2}\cdot\sum_{i\in\mathcal{A}_t(u)}\sum_j\sum_h \mathrm{Var}(Z^i_{jh}) \leq \frac{p'k\Delta_1^2}{4\beta nq\theta^2},\end{aligned} \tag{35}$$

where U is the rotation matrix HD_ξ, and (a) follows from the Cauchy–Schwarz inequality.

Hence, σ is bounded by:

$$\sigma^2 = \mathbb{E}\left[\|\nabla B' - \nabla B\|_F^2\right]/(p'k) \leq \frac{\Delta_1^2}{4\beta nq\theta^2}. \tag{36}$$

Plugging 36 into 34, we have:

$$\mathbb{E}\left[\|S^T(\nabla B' - \nabla B)\|_F^2\right] \leq \frac{mk\Delta_1^2}{4\beta nq\theta^2}. \tag{37}$$

□

E. Proof of Theorem 5.4

Under a given ϵ, we can rewrite q in terms of ϵ and θ:

$$q = \frac{(1-2\theta)^4}{\theta^2}\left(\frac{\alpha-1}{\alpha^2}\right)\frac{\beta n\epsilon}{C_0pk}. \tag{38}$$

Plugging in the expression of Theorem 5.4, it holds that:

$$\begin{aligned}\mathbb{E}\left[\|S^T\nabla B' - \nabla V\|_F^2\right] &\leq \frac{2m}{p}\|\nabla V\|_F^2 \\ &+\frac{C_0pmk^2\Delta_1^2}{4\beta^2n^2\epsilon}\frac{1}{(1-2\theta)^4}\left(\frac{\alpha^2}{\alpha-1}\right).\end{aligned} \tag{39}$$

Hence, the optimal p can be achieved by:

$$p = \sqrt{\frac{8n^2\beta^2\epsilon(1-2\theta)^4(\alpha-1)\|\nabla V\|_F^2}{C_0k^2\Delta_1^2\alpha^2}}. \tag{40}$$

Next, we rewrite the bound of $\|V\|_F^2$ in terms of Δ_1:

$$\begin{aligned}\|\nabla V\|_F^2 &= \frac{2}{\beta^2}\sum_{j=1}^n\sum_{h=1}^k\left(\sum_{i=1}^{\beta m}c_{ij}(r_{ij} - u_iv_j^T)u_{ih}\right)^2 \\ &\leq \frac{2(1+\alpha)^2e^2}{\beta^2}\sum_{j=1}^n\sum_{h=1}^k\sum_{i=1}^{\beta m}U_{ih}^2 \\ &\leq \frac{4\beta mn(1+\alpha)^2e^2B_1^2}{\beta^2} = \frac{\beta mn\Delta_1^2}{4}.\end{aligned} \tag{41}$$

Considering that $\theta \leq \frac{1}{4}$, the MSE can be minimized at:

$$p = \Theta\left(\frac{n\beta\sqrt{\beta mn\epsilon}}{k\sqrt{\alpha}}\right). \tag{42}$$

By replacing p in expression 39, we can obtain the optimal MSE given by $\Theta\left(mk\Delta_1^2\sqrt{\alpha m}/\sqrt{\beta n\epsilon}\right)$.

F. Proof of Theorem 5.7

We begin by proving Lemma 5.6. The Hessian matrix of item latent factor V is given by:

$$\nabla_V^2 f(V,U) = \begin{bmatrix} U^T C^1 U + \mu I & & \\ & \ddots & \\ & & U^T C^m U + \mu I \end{bmatrix}, \tag{43}$$

where $C^i \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $C^i_{jj} = c_{ji}$.

The eigenvalues of each $U^T C^i U + \mu I$ is lower bounded by μ since $U^T C^i U$ is a semi-positive definite matrix. Then we examine the upper bound of the eigenvalues. The absolute value of each element in the $U^T C^i U$ is bounded by:

$$\left|\sum_o u_{ho} u_{lo} c_{oi}\right| \leq \left|\sum_o u_{ho}\right| \left|\sum_o u_{lo}\right| (1+\kappa) \leq B_1^2 (1+\kappa). \tag{44}$$

The bound for the eigenvalues are derived from the Gershgorin Circle Theorem:

$$\sigma_{max}(U^T C^i U + \mu I) \leq \mu + k B_1^2 (1+\kappa). \tag{45}$$

Therefore, there exist a constant $L_g \leq \mu + k B_1^2 (1+\kappa)$, such that expression 27 is satisfied.

Next we proceed to the proof of Theorem 5.7. The computation of user latent factor from equation 5 ensures that $\nabla_U f(V,U) = 0$, and thus we focus on the convergence bound of $\nabla_V f(V,U)$. Based on Lemma 5.6, it holds that:

$$\begin{aligned} f(V^{t+1}, U^{t+1}) &\leq f(V^t, U^{t+1}) \\ -\gamma^t \langle \nabla_V f(V^t, U^{t+1}), S^T \nabla B^t \rangle &+ \frac{L_g (\gamma^t)^2}{2} \left[\|S^T \nabla B^t\|_F^2 \right]. \end{aligned} \tag{46}$$

Noted that $\|S^T \nabla B^t\|_F^2 \leq \|S^T \nabla B^t - \nabla V\|_F^2 + \|\nabla V\|_F^2$. By taking the expectation, we have:

$$\begin{aligned} \mathbb{E}[f(V^{t+1}, U^{t+1})] &\leq \mathbb{E}[f(V^t, U^{t+1})] \\ -\gamma^t \mathbb{E}[\|\nabla_V f(V^t, U^{t+1})\|^2] &+ \frac{L_g (\gamma^t)^2}{2} \left[\sigma_{\text{mse}}^2 + \|\nabla V\|_F^2 \right], \end{aligned} \tag{47}$$

where:

$$\sigma_{\text{mse}}^2 = \mathbb{E}\left[\|S^T \nabla B' - \nabla V\|_F^2\right] \leq \Theta\left(mk\Delta_1^2 \sqrt{\alpha}/\sqrt{\beta\epsilon}\right). \tag{48}$$

The Lipschitz continuous assumption suggests that:

$$\|\nabla V\|_F^2 \leq L_f^2. \tag{49}$$

For a fix V, the loss function is convex in U given the positive-definite Hessian matrix. Therefore, the update of U in equation 5 ensures that:

$$f(V^t, U^{t+1}) \leq f(V^t, U^t). \tag{50}$$

Therefore, we can arrive at the following expression by aggregating both sides of equation 46 and 50 over all iterations:

$$\min_t \mathbb{E}\left[\|\nabla_V f(V^t, U^t)\|^2\right] \leq \mathcal{O}\left(\frac{L_g (mk\Delta_1^2 \sqrt{\alpha}/\sqrt{\beta\epsilon} + L_f^2)}{\log T}\right). \tag{51}$$

Given that our algorithm guarantees $\|\nabla_U f(V,U)\| = 0$, we can complete the proof.

G. Datasets

Table 4 summarizes the statistics for the datasets.

Table 4. Statistics of the datasets.

	# Users	# Items	# Ratings	Density
ML100K	943	1,682	100,000	6.30%
ML5M	34,939	5,338	4,327,872	2.32%
Yelp	10,000	30,000	308,354	0.10%

H. Evaluation metrics

Our experiment sets the rating threshold to be 4, and the number of items recommended to be 10 per user. The metrics are defined as:

$$RMSE = \frac{1}{|usr|}\sum_i \sqrt{\frac{1}{|O_i|}\sum_{j\in O_i}(\hat{r}_{i,j} - r_{i,j})^2}, \tag{52}$$

$$Precision = \frac{1}{|usr|}\sum_i \frac{t_p^i}{t_p^u + f_p^i}, \tag{53}$$

$$Recall = \frac{1}{|usr|}\sum_i \frac{t_p^i}{t_p^i + f_n^i}, \tag{54}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{55}$$

where t_p^i, f_p^i, and f_n^i denotes the true positive, false positive, and false negative for user i respectively.

The computation of mAP relies on the following expression of average precision (AP):

$$AP = \frac{1}{t_p}\sum_{k=1}^{10} P(k) \cdot rel(k), \tag{56}$$

where $P(k)$ denotes the precision up to the k^{th} item in the recommendation list, and $rel(k)$ indicates whether the k^{th} item is relevant. Then the mAP is obtained by:

$$mAP = \frac{1}{|usr|}\sum_i AP_i. \tag{57}$$

I. Impact by User Activeness

We investigate the performance on users with varying levels of activeness to assess the differential impact of random projection and PBM. Using the Yelp dataset, users are divided into five groups based on the number of ratings: [0, 25), [25, 50), [50, 100), [100, 500), and [500, $+\infty$). The differential impact is measured by the variance of rating prediction error (Rastegarpanah et al., 2019). For a partition $G = \{G_1, ..., G_g\}$, we compute the mean squared error (MSE) for group k as:

$$L_k = \frac{1}{|\Omega_k|}\sum_{(i,j)\in\Omega_k}(\hat{r}_{i,j} - r_{i,j})^2, \tag{58}$$

where Ω_k denotes the set of ratings in group k. The variance is then computed by:

$$Var = \frac{1}{g^2}\sum_{k=1}^{g}\sum_{l>k}(L_k - L_l)^2. \tag{59}$$

Table 5 presents the MSE per group and the variance across groups. DPMF-RP increases the average MSE by 7% compared to central MF, with the most significant impact observed among users with fewer than 25 ratings, where MSE rises by 22%. Additionally, the variance across the five groups increases from 0.23 in central MF to 0.31 in DPMF-RP.

Table 5. MSE by user activeness on yelp dataset.

		Central MF		DPMF-RP	
	Partition	MSE	Var	MSE	Var
Group 1	[0, 25)	$1.81_{\pm 0.14}$	$0.23_{\pm 0.03}$	$2.21_{\pm 0.11}$	$0.31_{\pm 0.04}$
Group 2	[25, 50)	$2.04_{\pm 0.03}$		$2.09_{\pm 0.05}$	
Group 3	[50, 100)	$1.35_{\pm 0.02}$		$1.38_{\pm 0.00}$	
Group 4	[100, 500)	$1.02_{\pm 0.01}$		$1.07_{\pm 0.01}$	
Group 5	[500, $+\infty$)	$0.78_{\pm 0.00}$		$0.81_{\pm 0.00}$	

J. Ablation Studies

Our proposed Algorithm 2 involves DP mechanism given by Algorithm 1, secure aggregation, and random projection. To measure their affect on the model effectiveness, several cases in Table 6 are used to study the performance under the same privacy protection budget. Table 6 introduces the accuracy result of them.

Table 6. Ablation studies. RanProj represents random projection. In case 2, q and θ are computed with $\beta n = 1$.

	PBM	SegAgg	RanProj
DPMF-RP	$\surd$	$\surd$	$\surd$
Case 1: non-private	$\times$	$\times$	$\times$
Case 2: non-SecAgg	$\surd$	$\times$	$\surd$
Case 3: non-projected	$\surd$	$\surd$	$\times$
Case 4: non-DP	$\times$	$\surd$	$\surd$

In Table 7, we can observe that the performance of case 2 is significantly worse than other cases, as the absence of SecAgg necessitates larger scale of noises to retain the same level of privacy protection. Meanwhile, the F1 score and mAP listed in other three cases show similar level as the full algorithm, indicating that the noise mechanism and random projection operations have acceptable impact on the accuracy.

Table 7. Result of ablation studies.

		Case 1	Case 2	Case 3	Case 4
ML100K	RMSE	$1.00_{\pm 0.006}$	$1.10_{\pm 0.012}$	$1.01_{\pm 0.000}$	$1.01_{\pm 0.005}$
	Precision	$0.27_{\pm 0.002}$	$0.12_{\pm 0.003}$	$0.23_{\pm 0.002}$	$0.20_{\pm 0.001}$
	Recall	$0.23_{\pm 0.011}$	$0.13_{\pm 0.005}$	$0.26_{\pm 0.001}$	$0.24_{\pm 0.004}$
	F1	$0.24_{\pm 0.009}$	$0.13_{\pm 0.003}$	$0.24_{\pm 0.001}$	$0.22_{\pm 0.008}$
	mAP	$0.13_{\pm 0.006}$	$0.05_{\pm 0.002}$	$0.13_{\pm 0.003}$	$0.11_{\pm 0.003}$
ML5M	RMSE	$0.94_{\pm 0.001}$	$1.03_{\pm 0.022}$	$0.95_{\pm 0.001}$	$0.94_{\pm 0.002}$
	Precision	$0.27_{\pm 0.004}$	$0.11_{\pm 0.003}$	$0.20_{\pm 0.002}$	$0.19_{\pm 0.003}$
	Recall	$0.18_{\pm 0.008}$	$0.15_{\pm 0.004}$	$0.24_{\pm 0.001}$	$0.23_{\pm 0.004}$
	F1	$0.22_{\pm 0.005}$	$0.13_{\pm 0.002}$	$0.22_{\pm 0.002}$	$0.20_{\pm 0.006}$
	mAP	$0.12_{\pm 0.003}$	$0.06_{\pm 0.003}$	$0.12_{\pm 0.002}$	$0.11_{\pm 0.003}$
Yelp	RMSE	$1.00_{\pm 0.002}$	$1.26_{\pm 0.005}$	$1.03_{\pm 0.002}$	$1.04_{\pm 0.002}$
	Precision	$0.04_{\pm 0.001}$	$0.03_{\pm 0.001}$	$0.04_{\pm 0.000}$	$0.04_{\pm 0.001}$
	Recall	$0.08_{\pm 0.001}$	$0.06_{\pm 0.002}$	$0.08_{\pm 0.001}$	$0.09_{\pm 0.003}$
	F1	$0.06_{\pm 0.001}$	$0.04_{\pm 0.001}$	$0.05_{\pm 0.002}$	$0.05_{\pm 0.002}$
	mAP	$0.03_{\pm 0.000}$	$0.02_{\pm 0.001}$	$0.03_{\pm 0.001}$	$0.03_{\pm 0.001}$

K. Hyperparameter Analysis

Figure 4 presents the impact of different hyperparameters on F1 and mAP for ML5M and ML100K datasets. The parameter B_1 is used to clip the latent factor matrix U and calculate δ_1 in Algorithm 2. A higher value of B_1 results in smaller clipping error while greater loss in the privatization step. As the parameter value increases before reaching 0.001, the values of F1 and mAP for both datasets gradually increase. The values of F1 and mAP slightly decrease after $B_1 = 0.001$. The result indicates that an optimal choice of B_1 could be around 0.001.

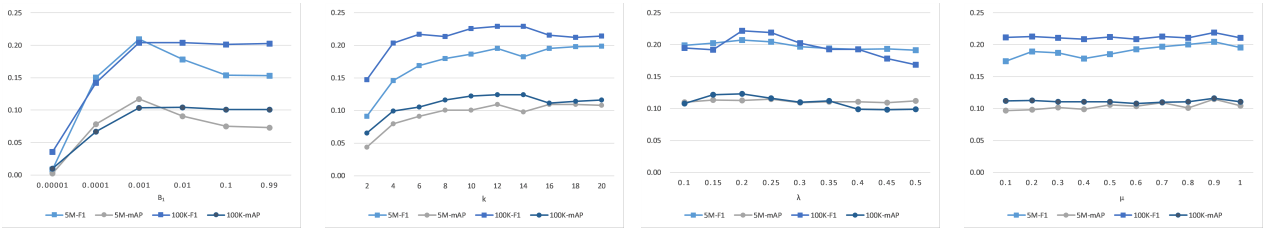

Figure 4. Accuracy with varying clipping norm B_1, latent factor size k, user regularization coefficient λ, and item regularization coefficient μ.

Regarding the size of the latent factor k, for ML100K, the values of F1 and mAP increase when k is smaller and equal to 14, and decrease slightly after that. For ML5M, the curve has a small fluctuation when k is equal to or greater than 14. Overall, the change of k had little effect on the results when its value reaches 10.

The user regularization coefficient λ has a greater impact on accuracies for ML100K than ML5M. We can observe that F1 and mAP reach the maximum value around $\lambda = 0.2$, with accuracy slightly declining after that point. For item regularization coefficient μ, the best performance occurs at $\mu = 0.9$ for ML5M dataset, while the impact of μ is less pronounced for ML100K.