Scilight

Journal of Machine Learning and Information Security
https://www.sciltp.com/journals/jmlis

Article

Bayesian Local Differential Privacy for Implicit Feedback Recommendation

Hao Tang [1], Yong Wang [1,2,*], Bo Li [1], Jiangzhou Deng [2,3] and Zhiqiang Zhang [2]

[1] School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] Key Laboratory of Business Intelligence and Management Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3] Mashang Consumer Finance Co., Ltd., Chongqing 400065, China
* Correspondence: wangyong1@cqupt.edu.cn

How To Cite: Tang, H.; Wang, Y.; Li, B.; et al. Bayesian Local Differential Privacy for Implicit Feedback Recommendation. *Journal of Machine Learning and Information Security* **2025**, *1*(1), 6.

Received: 30 August 2025
Revised: 22 October 2025
Accepted: 24 October 2025
Published: 31 October 2025

Abstract: Recommender systems have become essential tools for personalized information delivery, yet their reliance on user interaction data raises significant privacy concerns. Local differential privacy (LDP) provides strong privacy guarantees by perturbing user data before it leaves the client, but the injected noise often severely degrades recommendation accuracy. To address this challenge, we propose a novel LDP-based recommendation algorithm using Bayesian estimation. The method first perturbs users' implicit feedback locally using a randomized response mechanism, then reconstructs the true user–item interaction probabilities on the server through Bayesian inference. This two-step approach effectively mitigates noise while preserving privacy, enabling high-quality model training even under strict privacy constraints. Extensive experiments on three public datasets demonstrate that our method achieves superior recommendation performance compared with state-of-the-art algorithms, striking a favorable balance between privacy protection and utility. This study provides a practical and scalable solution for privacy-preserving recommendations, particularly in scenarios involving untrusted servers and sparse.

Keywords: local differential privacy; bayesian estimation; implicit feedback; recommender systems; privacy-preserving algorithm

1. Introduction

With the rapid advancement of Internet technologies, recommender systems (RSs) have become a key bridge connecting users with vast amounts of information, and are now widely deployed in domains such as e-commerce, online entertainment, and social networks [1,2]. Their fundamental objective is to model users' interest preferences by analyzing historical interaction data, thereby delivering personalized content and alleviating the decision-making burden caused by information overload [3,4].

In the evolution of RSs, the shift in data types has played a pivotal role. Early recommendation algorithms mainly relied on explicit feedback voluntarily provided by users (e.g., ratings and reviews). While explicit feedback can directly reflect user preferences, it is often limited by low participation rates, high collection costs, and small-scale datasets. In contrast, implicit feedback (e.g., click records, browsing time, and purchase history) requires no active user intervention and can be passively collected by the system. This type of data offers advantages such as low acquisition cost, broad coverage, and large scale, and has therefore become the primary data source for training modern recommendation models [5–7]. However, the heavy reliance of RSs on user data introduces significant privacy risks [8,9]. Interaction data often contains sensitive information, including personal preferences, behavioral patterns, and even identifiable attributes. Security incidents in recent years have repeatedly exposed the potential dangers of such data leaks [10,11]. For example, in 2024, the PowerSchool data breach compromised the personal information of over 70 million teachers and students, creating a high risk of identity theft.

To address this dilemma, differential privacy (DP) [12,13] has emerged as a key technique in privacy-preserving

RSs due to its rigorous mathematical privacy guarantees and strong scalability [14–17]. The core principle of DP is to inject carefully calibrated noise into data or computational processes, thereby obfuscating the contribution of any individual user. This ensures that the presence or absence of a single user has little impact on the system's output, which theoretically eliminates the risk of privacy leakage. Existing research on DP-based recommendation algorithms can be broadly categorized into two settings. The first is the centralized model (CDP) [9, 18], which assumes a trusted recommendation server that can directly access users' raw data and apply noise during the aggregation or computation stage. The second is the local model (LDP) [19, 20], designed for scenarios where the server cannot be trusted. In this setting, users perform privacy protection locally on their data before transmission, thereby fundamentally preventing the disclosure of raw information.

Despite its promise, existing studies still face three major limitations. Firstly, most research has focused on explicit feedback data, while studies on privacy protection for implicit feedback remain relatively limited [21, 22]. Given the central role of implicit feedback in real-world applications, this gap severely constrains the practical effectiveness of privacy-preserving techniques in RSs. Secondly, the trade-off between noise injection and recommendation performance has not been effectively resolved. In particular, under the LDP framework, satisfying stringent privacy constraints often requires injecting substantial noise at the client side. This process significantly reduces data utility, making it difficult to support the training of high-performance recommendation models [23–25]. Thirdly, the strong reliance of the centralized model on a "trusted server" assumption is misaligned with real-world conditions. In practice, RS operators, as both custodians and consumers of data, may suffer data breaches due to internal management lapses or external cyberattacks [26–28].

To overcome these limitations, this paper targets privacy protection for implicit feedback recommendation in untrusted server environments and proposes a LDP recommendation algorithm based on Bayesian estimation, aiming to achieve both strong privacy guarantees and high recommendation accuracy. The proposed method adopts a two-layer "local perturbation–server reconstruction" architecture. On the client side, users locally perturb implicit feedback data via a randomized response mechanism, ensuring that raw data is never directly exposed. On the server side, unbiased statistical analysis is conducted on the perturbed data collected from all users, which serves as prior knowledge. This prior is then combined with each user's individually perturbed data as evidence to reconstruct the user's true implicit feedback distribution through Bayesian estimation [29]. By leveraging statistical laws to correct noise interference, the proposed approach effectively mitigates the negative impact of noise on data utility in LDP, enabling the reconstructed data to support the training of high-quality recommendation models.

The main contributions of this paper are summarized as follows:

1. This paper addresses the lack of studies on LDP protection for implicit feedback data, providing a new methodology to balance privacy constraints and data utility. By integrating Bayesian estimation with differential privacy, the proposed method alleviates the inherent trade-off between strong privacy and low performance in traditional local models.
2. The proposed algorithm can be directly applied in untrusted server scenarios, enhancing user trust in RSs while delivering safe and personalized services, thereby promoting the secure and sustainable development of recommendation technologies.
3. Experiments on three public datasets show that compared with other comparison methods, our model provides better recommendation quality while ensuring a higher level of privacy protection.

The remainder of this paper is organized as follows. Section 2 presents the scenario analysis, and Section 3 introduces the proposed algorithm. Section 4 provides a detailed analysis of the algorithm, while Section 5 reports the experimental results. Finally, Section 6 concludes the paper and outlines directions for future work.

2. Scenario Analysis

In the scenario where the RS server is untrusted, as illustrated in Figure 1, the users' and items' real implicit feedback information is not directly transmitted to the server. Instead, the user client retains the authentic implicit feedback data locally. To ensure privacy protection, the data is perturbed before being uploaded. Upon receiving the perturbed implicit feedback data, which contains substantial noise, the server estimates unbiased statistical information and leverages it to construct prior knowledge. This prior knowledge is then combined with the perturbed implicit feedback to reconstruct the data on the server side. The reconstructed implicit feedback data is subsequently used to train the recommendation model and perform Top-N recommendations. Throughout this process, the server never accesses the actual implicit feedback of users, thereby providing a higher level of privacy protection on the user side.

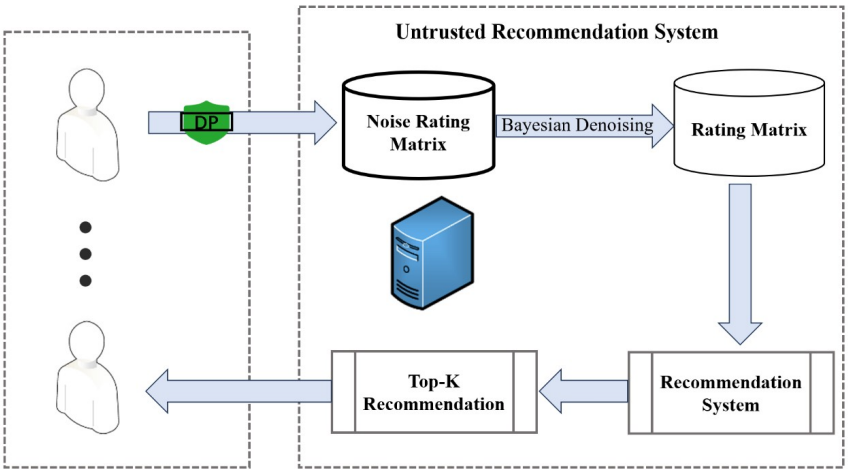


Figure 1. Recommendation framework in an untrusted environment.

3. Algorithm Design

To address the application scenario described in Section 2 and prevent user privacy leakage, the client perturbs all users' implicit feedback data for all items using the randomized response mechanism. This mechanism inevitably introduces a substantial amount of spurious implicit feedback, which adversely impacts subsequent model training. Upon receiving the perturbed implicit feedback matrix $\bar{\mathbf{R}}$, the server first computes the number of unbiased implicit feedback interactions $SeqUser_u$ and $SeqItem_i$ for each user u and each item i, respectively. Using $SeqUser$ and $SeqItem$, the prior probability matrix $\mathbf{P}$ is obtained. Then, based on $\bar{\mathbf{R}}$ and $\mathbf{P}$, the user-item interaction probability matrix $\widetilde{\mathbf{P}}$ is reconstructed using Bayesian estimation. Finally, for each user u, the top $SeqUser_u$ items with the highest interaction probabilities are selected, and for each item i, the top $SeqItem_i$ users with the highest probabilities are chosen to form the final implicit feedback matrix $\widetilde{\mathbf{R}}$.

3.1. Client-Side Perturbation Algorithm

Randomized response (RR) mechanism is a foundational technique in LDP. RR allows individuals to answer sensitive questions while provably hiding their true responses. In the context of LDP, each user perturbs their private data locally—before sending it to an untrusted server—using a randomized protocol that satisfies the formal privacy guarantee of ϵ-LDP.

Specifically, for binary implicit feedback (e.g., whether a user interacted with an item), RR flips the true value with a calibrated probability that depends on the privacy budget ϵ. A smaller ϵ means stronger privacy, while a larger ϵ allows more accurate data transmission at the cost of weaker privacy. Crucially, although each individual report is noisy, the aggregate statistics over many users can be unbiasedly estimated due to the known randomization probabilities. This property enables the server to reconstruct global patterns without ever accessing raw user data—striking a balance between privacy and utility.

In our work, RR serves as the client-side privacy mechanism that ensures ϵ-LDP, while the server leverages Bayesian inference to refine per-user interaction estimates using both the perturbed data and global statistical priors.

In a RS, the user client records the real implicit feedback data of users and uses the randomized response mechanism to flip the local real implicit feedback data and send it to the recommendation server, as shown in Algorithm 1. Let the real implicit feedback sequence of user u be $\mathbf{R}_u \in (0,1)^m$, where $\mathbf{R}_u$ represents the u-th row of the implicit feedback matrix $\mathbf{R}$, and n and m represent the number of users and items, respectively. The probability p_{flipped} of random flipping and the probability $p_{\text{unflipped}}$ of non-random flipping can be calculated according to the given privacy budget parameter ϵ.

$$p_{\text{flipped}} = \frac{1}{1+\exp(\epsilon)}, \tag{1}$$

$$p_{\text{unflipped}} = \frac{\exp(\epsilon)}{1+\exp(\epsilon)}. \tag{2}$$

For a given user u and a specific item i, select the implicit feedback R_{ui}. Then, draw a random value $t \sim Uniform(0,1)$ from the uniform distribution over the interval $[0,1]$. If $t \leq p_{\text{flipped}}$, set $\bar{R}_{ui} = 1 - R_{ui}$; otherwise, set $\bar{R}_{ui} = R_{ui}$. User u repeats this random flipping process for all items to generate the perturbed sequence $\bar{\mathbf{R}}_u$. This procedure is applied to all users, ensuring that each data record satisfies differential privacy requirements,

resulting in the final perturbed implicit feedback matrix $\bar{\mathbf{R}}$.

Algorithm 1: Client-Side Random Response Perturbation Algorithm

Input: The implicit feedback sequence $\mathbf{R}_u \in (0,1)^m$ of user u, privacy budget parameter ϵ.

Output: The perturbed implicit feedback sequence $\bar{\mathbf{R}}_u \in (0,1)^m$.

for $i \in (1, 2, \ldots, m)$ **do**

$$\bar{R}_{ui} = \begin{cases} R_{ui}, & \text{with probability } \frac{\exp(\epsilon)}{1+\exp(\epsilon)} \\ 1 - R_{ui}, & \text{with probability } \frac{1}{1+\exp(\epsilon)} \end{cases}$$

end

return $\bar{\mathbf{R}}_u$

3.2. Server Reconstruction Algorithm

While the randomized response mechanism guarantees user-level privacy, it introduces significant noise into the observed data—especially under stringent privacy budgets. Directly using these perturbed interactions for recommendation leads to poor performance, as the signal-to-noise ratio becomes too low. To address this, we propose a server-side Bayesian estimation procedure that reconstructs a more accurate estimate of each user's true interaction probabilities by combining the noisy observations with global statistical knowledge.

The key idea is simple yet powerful: although we cannot trust any single user's report, we can reliably estimate aggregate statistics across all users because the randomization process is known and unbiased. These global aggregates serve as informative priors in a Bayesian framework.

Specifically, for a given user u and item i, let $y_{ui} \in \{0, 1\}$ denote the true interaction, and $\tilde{y}_{ui}$ the perturbed value received by the server. Under the RR mechanism, the probability of observing $\tilde{y}_{ui}$=1 depends on both y_{ui} and ϵ. Using Bayes' rule, we compute the posterior probability that $y_{ui} = 1$ given $\tilde{y}_{ui}$:

$$P(y_{ui} = 1 \mid \tilde{y}_{ui}) = \frac{P(\tilde{y}_{ui} \mid y_{ui} = 1) \cdot P(y_{ui} = 1)}{P(\tilde{y}_{ui})} \tag{3}$$

where $P(y_{ui} = 1)$ is the prior probability of interaction. Instead of assuming a uniform prior, we estimate this prior from the global data: for user u, we use the estimated total number of interactions $\hat{d}_u$; for item i, we use its estimated popularity $\hat{p}_i$. A common choice is to set $P(y_{ui} = 1) \propto \hat{d}_u \cdot \hat{p}_i$, reflecting the intuition that active users are more likely to interact with popular items.

This posterior probability serves as a soft, denoised label for the user–item pair, replacing the hard 0/1 value from RR. By doing so, we effectively correct the noise in a statistically principled way—leveraging population-level patterns to improve individual-level inference, all while preserving the original LDP guarantee.

As shown in Algorithm 2, the server receives the perturbed sequence $\bar{\mathbf{R}}_u$ from each user and aggregates them into the perturbed matrix $\bar{\mathbf{R}}$. It then calculates the total unbiased data for each user, $SeqUser_u$, and for each item, $SeqItem_i$. Using these totals, the server estimates the prior interaction probability between users and items. The received perturbed matrix $\bar{\mathbf{R}}$ is treated as the evidence to compute the posterior interaction probability for each user–item pair. Finally, the server reconstructs the user–item interaction matrix based on the posterior probabilities, which serves as the training input for the recommendation model.

The server computes the noisy feedback count $Seq\bar{U}ser_u$ and $Seq\bar{I}tem_i$ for each user and each item from the received matrix $\bar{\mathbf{R}}$, respectively:

$$Seq\bar{U}ser_u = sum(\bar{\mathbf{R}}_u), \tag{4}$$

$$Seq\bar{I}tem_i = sum(\bar{\mathbf{R}}_i^T). \tag{5}$$

Suppose that the true feedback count of user u is m_1. After applying the randomized response mechanism, the expected number of true feedback entries that remain unchanged is $m_1 * p_{\text{unflipped}}$. The number of non-interactions for user u is $m - m_1$, and the expected number of these that are flipped into implicit feedback is $(m - m_1) * p_{\text{flipped}}$. Therefore, the expected feedback count for user u after perturbation is:

$$E\left(Seq\bar{U}ser_u\right) = m_1 * p_{\text{unflipped}} + (m - m_1) * p_{\text{flipped}} . \tag{6}$$

The final goal is to estimate the true number of feedback m_1, which can be obtained according to Equation (5):

$$m_1 = \frac{(p_{\text{unflipped}} - 1) * m + E(Seq\bar{U}ser_u)}{2 * p_{\text{unflipped}} - 1}. \tag{7}$$

Algorithm 2: Server-Side Bayesian Reconstruction Algorithm

Input: The perturbed implicit feedback matrix $\overline{\mathbf{R}} \in (0,1)^{n*m}$, the privacy budget parameter ε.
Output: The reconstructed implicit feedback matrix $\widetilde{\mathbf{R}} \in (0,1)^{n*m}$.

// Calculate the unbiased implicit feedback count for each user **for** $u \in (1,2,\ldots,n)$ **do**
  $Seq\bar{U}ser_u = sum\left(\bar{\mathbf{R}}_u\right)$
  // Calculate the unbiased estimate of $SeqUser_u$ by Eq. (7)
end
// Calculate the unbiased implicit feedback count for each item
for $i \in (1,2,\ldots,m)$ **do**
  $Seq\bar{I}tem_i = sum\left(\bar{\mathbf{R}}_i^T\right)$
  // Calculate the unbiased estimate of $SeqItem_i$ by Eq. (8)
end
// Estimate the prior interaction probabilities using Beta model
$\mathbf{P} = Beta\ (SeqUser,\ SeqItem)$
// Reconstruct the interaction probability matrix using Bayesian estimation
for $u \in (1,2,\ldots,n)$ **do**
  for $i \in (1,2,\ldots,m)$ **do**
    if $\overline{\mathbf{R}}_{ui} == 1$ **then**
      Obtain $\widetilde{P}_{ui}$ by Eq. (18)
    end
    else
      Obtain $\widetilde{P}_{ui}$ by Eq. (19)
    end
  end
end
for $u \in (1,2,\ldots,n)$ **do**
  $argsort\left(\widetilde{\mathbf{P}}_u\right)$
  Select the Top $SeqUser_u$ items with the highest probability as the training interactions $\widetilde{\mathbf{R}}_u$ of user u
end
for $i \in (1,2,\ldots,n)$ **do**
  $argsort\left(\widetilde{\mathbf{P}}_i^T\right)$
  Select the top $SeqItem_i$ users with the highest probability as the training interactions $\widetilde{\mathbf{R}}_i^T$ of item i
end
$\widetilde{\mathbf{R}} = \widetilde{\mathbf{R}}_u \cup\ \widetilde{\mathbf{R}}_i^T$
return $\widetilde{\mathbf{R}}$

This statistic is an unbiased estimate of the true number of implicit feedback. Thus, the server can obtain the total unbiased feedback of user u.

$$SeqUser_u = \frac{(p_{\text{unflipped}} - 1) * m + Seq\bar{U}ser_u}{2 * p_{\text{unflipped}} - 1}. \tag{8}$$

Similarly, the total unbiased feedback of item i is

$$SeqItem_i = \frac{(p_{\text{unflipped}} - 1) * n + Seq\bar{I}tem_i}{2 * p_{\text{unflipped}} - 1}. \tag{9}$$

Given the total unbiased feedback $SeqUser_u$ and $SeqItem_i$ of user u and item i, the server employs the widely used β-model [30] from social network analysis to estimate the prior interaction probability between user u and item i as the prior probability. Given user β vector $\beta_U = (\beta_{u_1}, \beta_{u_2}, \ldots, \beta_{u_n})$ and item β vector $\beta_I = (\beta_{i_1}, \beta_{i_2}, \ldots, \beta_{i_m})$, the interaction probability P_{ui} between user u and item i is

$$P_{ui} = \frac{\exp\left(\beta_u + \beta_i\right)}{1 + \exp\left(\beta_u + \beta_i\right)}. \tag{10}$$

For each user u and each item i, the maximum likelihood estimate $\hat{\beta}_u$ and $\hat{\beta}_i$ of their β vectors need to satisfy

$$SeqUser_u = \sum_{i=1}^{m} \frac{\exp\left(\hat{\beta}_u + \hat{\beta}_i\right)}{1 + \exp\left(\hat{\beta}_u + \hat{\beta}_i\right)}, \tag{11}$$

$$SeqItem_i = \sum_{u=1}^{n} \frac{\exp\left(\hat{\beta}_u + \hat{\beta}_i\right)}{1 + \exp\left(\hat{\beta}_i + \hat{\beta}_i\right)}. \tag{12}$$

To meet the above conditions, Chatterjee et al. [31] proposed an efficient method to solve their maximum likelihood estimate by

$$\hat{\beta}_u = log\left(SeqUser_u\right) - log\left(\sum_{i=1}^{m} \frac{1}{\exp\left(-\beta_u\right) + \exp\left(\beta_i\right)}\right), \tag{13}$$

$$\hat{\beta}_i = log\left(SeqItem_i\right) - log\left(\sum_{u=1}^{n} \frac{1}{\exp\left(-\beta_i\right) + \exp\left(\beta_u\right)}\right). \tag{14}$$

Then, alternate solutions are carried out until convergence. Finally, the prior interaction probability between user u and item i is

$$P_{ui} = \frac{\exp\left(\hat{\beta}_u + \hat{\beta}_i\right)}{1 + \exp\left(\hat{\beta}_u + \hat{\beta}_i\right)}. \tag{15}$$

Take the interaction probability matrix $\mathbf{P}$ solved from the perturbed matrix $\bar{\mathbf{R}}$ as the prior, and the received perturbed matrix $\bar{\mathbf{R}}$ as the evidence to calculate the true posterior interaction probability. The server knows the privacy budget ε of the client perturbation. If the true feedback value $R_{ui} = 1$, the probability distribution is

$$\begin{cases} Pr(\bar{R}_{ui} = 1) = p_{\text{unflipped}}, \\ Pr(\bar{R}_{ui} = 0) = p_{\text{flipped}}. \end{cases} \tag{16}$$

If the true feedback value $R_{ui} = 0$, the probability distribution of the perturbed value after random flipping is

$$\begin{cases} Pr(\bar{R}_{ui} = 1) = p_{\text{flipped}}, \\ Pr(\bar{R}_{ui} = 0) = p_{\text{unflipped}}. \end{cases} \tag{17}$$

Combined with the known prior probability distribution, the probability distribution of the finally perturbed $\bar{R}_{ui}$ can be obtained as

$$\begin{cases} Pr(\bar{R}_{ui} = 1) = P_{ui} * p_{\text{unflipped}} + (1 - P_{ui}) * p_{\text{flipped}}, \\ Pr(\bar{R}_{ui} = 0) = P_{ui} * p_{\text{flipped}} + (1 - P_{ui}) * p_{\text{unflipped}}. \end{cases} \tag{18}$$

Based on Bayes' formula, when the observed perturbed value is 1, the probability that the true value is 1 can be obtained as

$$\begin{aligned} Pr(R_{ui} = 1 | \bar{R}_{ui} = 1) &= \frac{Pr(\bar{R}_{ui} = 1 | R_{ui} = 1) * Pr(R_{ui} = 1)}{Pr(\bar{R}_{ui} = 1)} \\ &= \frac{(1 - p_{\text{flipped}}) * P_{ui}}{P_{ui} * (1 - p_{\text{flipped}}) + (1 - P_{ui} * p_{\text{flipped}})}. \end{aligned} \tag{19}$$

when the observed perturbed value is 0, the probability that the true value is 1 can be obtained as

$$\begin{aligned} Pr(R_{ui} = 1 \mid \bar{R}_{ui} = 0) &= \frac{Pr(\bar{R}_{ui} = 0 \mid R_{ui} = 1) * Pr(R_{ui} = 1)}{Pr(\bar{R}_{ui} = 0)} \\ &= \frac{p_{\text{flipped}} * P_{ui}}{(1 - P_{ui}) * (1 - p_{\text{flipped}}) + P_{ui} * p_{\text{flipped}}}. \end{aligned} \tag{20}$$

The final true posterior interaction probability is $\widetilde{\mathbf{P}}$. For each user u, we select the top $SeqUser_u$ items with

the highest probability according to the ranking of $\widetilde{\mathbf{P}}_u$ as the training data. For each item i, we select the top $SeqItem_i$ users with the highest probability according to the ranking of $\widetilde{\mathbf{P}}_i^T$ as the training data. Finally, the two training data are combined to obtain the final training data $\widetilde{\mathbf{R}}$.

4. Algorithm Analysis

4.1. Security Analysis

Theorem 1. *Algorithm 1 satisfies ϵ-LDP.*

Proof. Let $\mathbf{R}_U = (R_{u_1}, R_{u_2}, \ldots, R_{u_n})$ be the original rating sequence of user u, $\mathbf{R}'_u = \left(R'_{u_1}, R'_{u_2}, \ldots, R'_{u_n}\right)$ be the neighbor rating sequence of user u, where only data R_{ui} and R'_{ui} in two sequences is different. Thus, assume that $R_{ui} = 1$ and $R'_{ui} = 0$, then

$$\frac{Pr\left(\mathbf{R}_u \mid R_{ui} = 1\right)}{Pr\left(\mathbf{R}_u \mid R'_{ui} = 1\right)} = \frac{Pr\left(R_{u_1}\right) * Pr\left(R_{u_2}\right) * \ldots * Pr\left(R_{u_n}\right)}{Pr\left(R'_{u_1}\right) * Pr\left(R'_{u_2}\right) * \ldots * Pr\left(R'_{u_n}\right)} = \frac{Pr\left(R_{ui}\right)}{Pr\left(R'_{ui}\right)}. \tag{21}$$

If the output value is 1, then

$$\frac{Pr\left(R_{ui}\right) = 1}{Pr\left(R'_{ui}\right) = 1} = \frac{\frac{\exp(\varepsilon)}{1+\exp(\varepsilon)}}{\frac{1}{1+\exp(\varepsilon)}} = \exp\left(\varepsilon\right). \tag{22}$$

If the output value is 0, then

$$\frac{Pr\left(R_{ui}\right) = 0}{Pr\left(R'_{ui}\right) = 0} = \frac{\frac{1}{1+\exp(\varepsilon)}}{\frac{\exp(\varepsilon)}{1+\exp(\varepsilon)}} = \frac{1}{\exp\left(\varepsilon\right)} \leq \exp\left(\varepsilon\right). \tag{23}$$

□

Since each user satisfies this condition, Algorithm 1 satisfies ϵ-LDP, and the proof of Theorem 1 is completed.

4.2. Utility Analysis

To balance privacy protection and data utility, two complementary strategies are adopted.

(1) Client-side perturbation: The random response algorithm is applied to perturb implicit feedback before transmission, ensuring privacy. Given the high sparsity of recommender datasets, such perturbation may inject considerable noise. To mitigate this, the server estimates unbiased implicit feedback counts from the perturbed data and infers prior interaction probabilities between users and items, thereby partially recovering utility.

(2) Server-side reconstruction: Treating the noisy implicit feedback matrix as observed evidence, a Bayesian estimation framework reconstructs the posterior interaction probability matrix, extracting high-confidence user–item pairs as training data. This process effectively suppresses noise-induced distortions and enhances feature extraction accuracy.

By integrating these two strategies, the proposed method strikes a balance between privacy protection and data utility, minimizing the detrimental effects of privacy noise while preserving the overall recommendation performance.

5. Experimental Analysis

5.1. Datasets

Three public datasets—Movielens100K (ML-100K), Movielens1M (ML-1M), and an Amazon dataset—are used to evaluate the proposed algorithm. The Movielens datasets are collected by the GroupLens research team via a movie recommendation platform, while the Amazon dataset is obtained from the Amazon e-commerce platform and contains product ratings provided by users. In both datasets, ratings range from 1 to 5, offering rich interaction signals for recommender system research. Since this study focuses on implicit feedback scenarios, the original explicit rating matrices are preprocessed via binarization: all non-zero ratings are set to 1, indicating a user–item interaction, while missing entries are treated with negative sampling and assigned 0. This produces an implicit feedback matrix that more closely aligns with real-world recommendation settings. Table 1 summarizes the basic statistics of the datasets. By evaluating the proposed method on these datasets, this work aims to demonstrate its capability to achieve a trade-off between privacy protection and recommendation utility, thereby offering insights into privacy-preserving recommender system design.

Table 1. Basic information of three datasets.

Attribute	ML-100K	ML-1M	Amazon
Number of Users	943	6040	5130
Number of Items	1682	3706	1685
Number of Interactions	100,000	1,000,209	37,126
Sparsity	93.70%	95.53%	99.57%

5.2. Evaluation Metrics

The leave-one-out evaluation protocol is adopted to assess the proposed algorithm. For each user u, one interacted item is randomly selected as the test item i^+, while 99 non-interacted items are randomly sampled as negative instances. These items are combined into a candidate set, and the ranking results are evaluated. Two widely used ranking metrics are considered: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K).

HR@K: Measures whether the test item appears within the top-ranked results.

$$\mathrm{HR@K} = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}\left(\mathrm{rank}_u(i^+) \leq K\right), \tag{24}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\mathrm{rank}_u(i^+)$ denotes the position of i^+ in user u's ranked list.

NDCG@K: Extends HR by considering the position of the test item in the ranked list, assigning higher scores to higher-ranked positions.

$$\mathrm{NDCG@K} = \frac{1}{|U|} \sum_{u \in U} \frac{\mathbb{I}\left(\mathrm{rank}_u(i^+) \leq K\right)}{\log_2\left(\mathrm{rank}_u(i^+) + 1\right)}. \tag{25}$$

5.3. Implementation Details

All the experiments were conducted on a server equipped with Intel Xeon w5-3435X CPU (3.1 GHz, 16 cores total), 256 GB RAM, and three NVIDIA RTX 5000 Ada GPUs (32 GB memory each). The operating system was Ubuntu 20.04 LTS. And we implemented our algorithms in Python 3.9, using PyTorch 1.12 for model training.

In the experiments, the parameter settings of our proposed algorithm are as follows: the latent factor dimension k is set to 4, the maximum number of iterations to 500, the learning rate α to 0.002, and the regularization parameters λp and λq to 0.001. Due to the inherent randomness of DP-induced noise, the experiments were repeated 10 times, and we report the average results as the final outcome for the two evaluation metrics. Absolutely, the training and testing parameters were consistently set across all datasets.

5.4. Comparison of Interaction Quantities before and after Reconstruction

To evaluate the denoising capability of the proposed algorithm, we compare the changes in interaction quantities produced by two LDP recommendation approaches: the random response mechanism and the Bayesian estimation–based method. The results are summarized in Table 2, where, for each dataset, the first column reports the number of interactions obtained after applying the random response mechanism, and the second column presents the number obtained via the proposed Bayesian estimation–based LDP algorithm.

Table 2. The number of interactions before and after reconstruction

Privacy Budget ε	ML-100K	ML-1M	Amazon
0.1	758,904 275,115	10,680,392 2,335,014	4,107,867 855,797
0.2	722,612 180,346	10,172,921 1,833,853	3,893,142 475,624
0.3	689,604 165,935	9,671,758 1,618,660	3,683,095 346,363
0.4	656,172 154,130	9,174,235 1,479,576	3,473,085 260,245
0.6	591,021 135,597	8,223,157 1,365,782	3,071,261 186,536
0.8	528,733 123,683	7,319,803 1,280,900	2,689,365 145,310
1.0	472,792 122,076	6,479,733 1,231,388	2,337,634 124,218
#Real interactions	100,000	1,000,209	37,126

It can be seen from Table 2 that as the privacy budget ϵ increases, the number of recorded interactions in

all datasets exhibits a downward trend. This occurs because a larger privacy budget reduces the probability of random flipping, thereby lowering the likelihood that the majority of unobserved interactions (given that dataset sparsity typically exceeds 90%) are mistakenly flipped into positive interactions. Consequently, the total amount of perturbed interaction data decreases. Furthermore, for every dataset, the proposed Bayesian estimation–based algorithm consistently yields fewer perturbed interactions than the random response method. This advantage arises from its unbiased estimation of user–item interaction counts, which aligns the estimated interaction frequencies more closely with the ground truth, effectively mitigating the influence of noisy data. As ϵ increases, the flipping probability continues to decline, and the variance of the unbiased estimates also decreases, further aligning the reconstructed interactions with the actual interactions.

As shown in Table 2, the random response mechanism alone generates a large number of spurious interactions—especially under strict privacy—due to random flipping of non-interactions into positive signals. In contrast, our method consistently yields interaction counts much closer to the ground-truth values (e.g., 100,000 for ML-100K), thanks to its unbiased estimation of user and item activity levels and subsequent Bayesian correction. This reduction in noise is crucial: it confirms that our server-side reconstruction effectively recovers meaningful interaction structure from heavily perturbed data, thereby preserving data utility without compromising privacy. This capability directly supports our central claim that Bayesian inference can reconcile the tension between strong LDP guarantees and high recommendation accuracy.

Overall, the proposed algorithm leverages Bayesian estimation to effectively denoise interaction data, substantially reducing the noise introduced by random response while preserving privacy. These results demonstrate its strong capability to address sparsity and privacy concerns, offering a robust foundation for deploying RSs in privacy-sensitive applications.

5.5. Comparison with LDP Recommendation Algorithms

To comprehensively assess the effectiveness of the proposed algorithm in mitigating data input perturbations, two representative baseline models for denoised recommendation systems are selected:

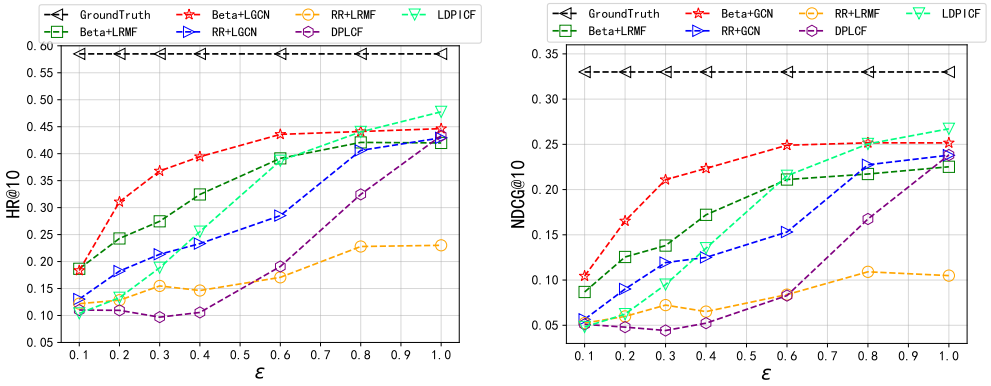
- LRMF [32]: It builts upon a logistic regression framework for matrix factorization. By integrating logistic regression with latent factor modeling, it effectively captures user–item interaction patterns, thereby enabling accurate user preference prediction.
- LGCN [33]: It introduces graph convolutional operations into recommendation, omits nonlinear activation and feature transformation, and focuses on high-order connectivity modeling, which efficiently learns structural patterns in high-order interaction data, yielding superior performance in complex user–item relationship modeling.

The proposed algorithm is integrated into both LRMF and LGCN to validate its robustness across different data types and interaction scenarios. Since our approach is grounded in the random response mechanism, each baseline is evaluated under two privacy-preserving settings: (1) Random Response (RR) and (2) Bayesian estimation–based LDP (Beta). This yields four algorithmic variants: RR+LRMF, RR+LGCN, Beta+LRMF, and Beta+LGCN. Here, RR+LRMF and RR+LGCN apply random response perturbation prior to recommendation, while Beta+LRMF and Beta+LGCN employ the proposed Bayesian estimation–based perturbation. This setup enables a systematic evaluation of the trade-off between privacy protection and recommendation performance, providing empirical evidence for privacy-preserving recommender deployment.

In addition, two similarity-based LDP recommendation algorithms are adopted for further comparison:

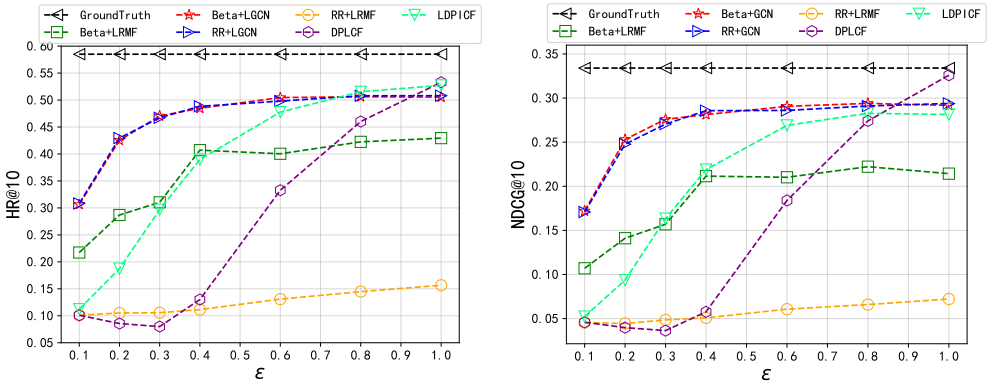
- LDPICF [19]: This algorithm also uses local random flipping for privacy protection. The server reconstructs data via joint frequency estimation, mitigating noise effects in similarity computation and improving the accuracy of recommendations.
- DPLCF [20]: A distributed recommendation method leveraging random flipping at the client side to protect implicit feedback. On the server side, joint cardinality estimation is employed to reconstruct item–item co-occurrence counts, thereby reducing similarity estimation errors and enhancing recommendation accuracy.

The experimental results are reported in Figures 2–4, which present HR@10 and NDCG@10 scores on three datasets, respectively, under different privacy budgets. For reference, the GroundTruth values correspond to the LRMF model without differential privacy.



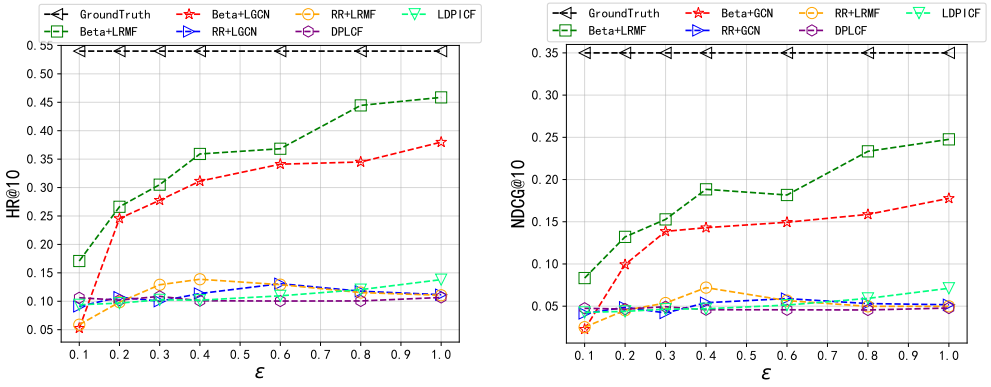
(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 2. Performance of our model and its variants on all datasets.



(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 3. Performance comparison of LDP model on ML-1M.



(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 4. Performance comparison of LDP model on Amazon.

It can be found from Figures 2–4 that as the privacy budget ε increases, the performance of all models improves, which aligns with the general behavior of privacy-preserving models. Under the same experimental settings, the ML-1M dataset consistently outperforms the ML-100K and Amazon datasets in terms of HR@10 and NDCG@10. This can be attributed to the larger number of users involved in item rating in ML-1M, providing richer interaction information and enabling faster model convergence. Meanwhile, the ML-100K dataset generally yields better results than the Amazon dataset. This is because the Amazon dataset exhibits substantially higher sparsity than ML-100K and ML-1M. The results confirm that higher dataset sparsity negatively impacts recommendation performance, highlighting the importance of data richness in enhancing RSs.

Compared with input perturbation based on RR, the proposed LDP recommendation algorithm using Beta model demonstrates a clear advantage in recommendation performance, primarily due to its effective denoising

capability. In the ML-100K dataset, the Beta+LGCN configuration achieves the highest HR@10 and NDCG@10 across all privacy budgets, with Beta+LRMF ranking second. By contrast, RR+LGCN and RR+LRMF suffer significant degradation due to the substantial noise introduced by RR. Specifically, in LGCN, RR-induced noise interferes with the extraction of high-order neighbor features, reducing its effectiveness. The Bayesian-based method mitigates this by more accurately recovering high-order features through denoising, thereby enhancing recommendations. For LRMF, RR similarly introduces noise into direct interaction data, increasing feature extraction errors, whereas the Bayesian-based approach alleviates this impact. In the ML-1M dataset, Beta+LGCN and RR+LGCN exhibit comparable performance, likely due to the dataset's abundant interactions, which enhance the generalization ability of LGCN in high-order feature extraction and partially offset RR noise. However, this effect is absent in LRMF, where noise in direct interaction features still severely affects recommendations. On the Amazon dataset, Beta+LRMF achieves the best results, likely because high sparsity favors direct feature extraction, whereas Beta+LGCN suffers from insufficient high-order neighbor information. This suggests that in sparse settings, models based on direct interactions (e.g., LRMF) may outperform those relying on high-order relationships (e.g., LGCN).

Notably, even under stringent privacy, our approach achieves performance close to the non-private GroundTruth model. This demonstrates that the improved interaction estimation enabled by Bayesian inference translates directly into better ranking quality. In other words, by more accurately reconstructing the underlying user preferences from noisy inputs, our method preserves the signal needed for effective personalization—thereby fulfilling the paper's primary objective of achieving a favorable trade-off between privacy and utility in implicit-feedback recommendation systems.

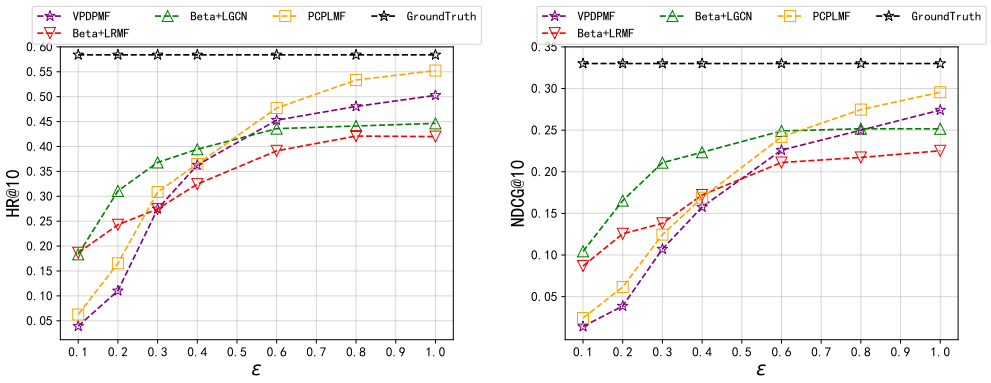
In summary, the proposed recommendation algorithm exhibits strong denoising performance across diverse datasets and privacy budgets. Its advantages are particularly evident in scenarios requiring high-order neighbor modeling or direct interaction modeling. Moreover, experimental results indicate that under sparse data conditions, privacy budget increments have limited effects on improving recommendation quality, and our method is relatively less sensitive to sparsity, making it especially valuable in balancing privacy preservation and recommendation accuracy.

5.6. Comparison with CDP Recommendation Algorithms

In this section, we compare the proposed algorithm with CDP recommendation methods. The selected baseline algorithms include VPDPMF and PCPLMF. VPDPMF, proposed by Ran et al. [34], is a vector perturbation recommendation algorithm based on matrix factorization, while PCPLMF is a polynomial coefficient perturbation logical matrix factorization algorithm. VPDPMF and PCPLMF generally achieve superior recommendation performance across datasets. This advantage stems from their focus on protecting the impact of a single interaction change over the entire dataset, resulting in an introduced noise level of $O(1)$. In contrast, Beta+LGCN and Beta+LRMF need to protect the interaction records of each user, with a stronger privacy protection level. The noise level they need to introduce is $O(n)$, which is much higher than that of CDP algorithms.

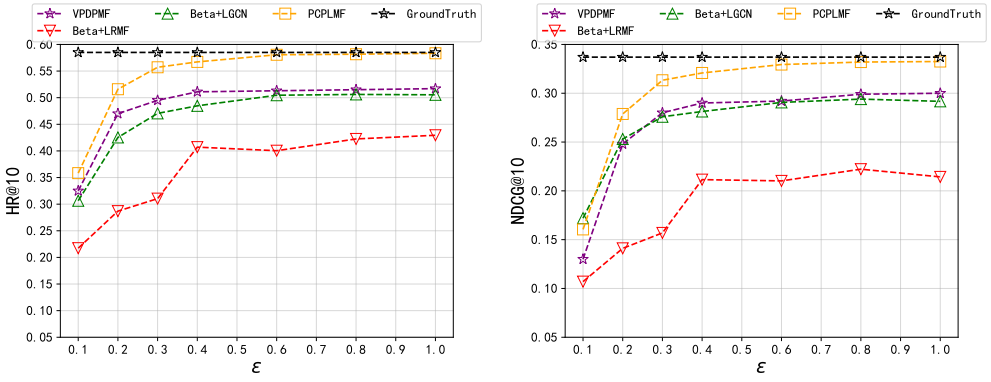
It can be found from Figures 5–7 that on the ML-100K dataset, Beta+LGCN and Beta+LRMF outperform under lower privacy budgets. This is because the Bayesian estimation-based LDP approach tends to assign higher interaction probabilities to frequently interacted items, resembling a "popular items" recommendation effect. Meanwhile, the VPDPMF and PCPLMF emphasize detailed user and item feature extraction, which suffers under high noise levels, degrading recommendation quality. As the privacy budget increases, the recommendation effects of VPDPMF and PCPLMF exceed those of Beta+LGCN and Beta+LRMF. This is because the noise decreases, and the model's feature extraction becomes increasingly accurate. However, Beta+LGCN and Beta+LRMF lack the ability to extract personalized features, resulting in poorer performance. On the ML-1M dataset, VPDPMF and PCPLMF outperform Beta+LGCN and Beta+LRMF even at lower privacy budgets due to the larger data volume, allowing all algorithms to achieve better recommendations. In the Amazon dataset, characterized by high sparsity, all algorithms exhibit limited performance. Since VPDPMF is designed primarily for explicit feedback, it performs worse than Beta+LGCN and Beta+LRMF under lower privacy budgets and struggles to fully leverage implicit feedback. Conversely, PCPLMF, tailored for implicit feedback, consistently achieves the best results.

To sum up, despite the stronger privacy guarantees in LDP settings, the performance gap between Beta+LGCN, Beta+LRMF and CDP-based algorithms remains moderate, demonstrating the competitiveness of the proposed method in privacy-preserving recommendation.



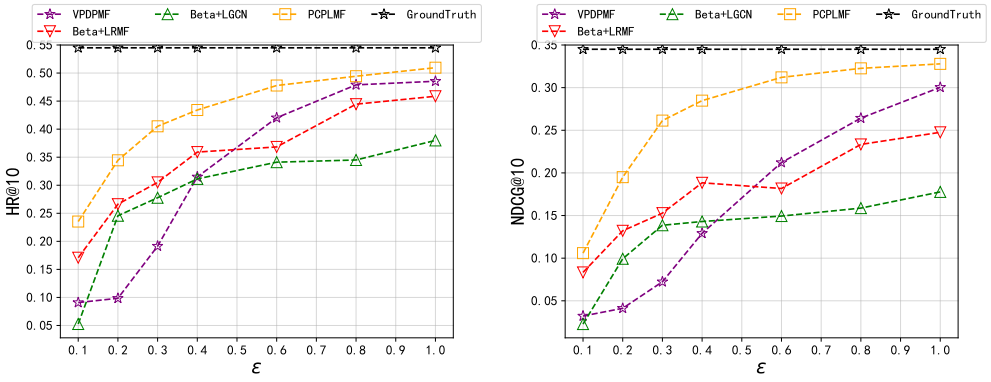
(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 5. Performance comparison of CDP model on ML-100K.



(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 6. Performance comparison of CDP model on ML-1M.



(a) HR@10 vs. ε **(b)** NDCGR@10 vs. ε

Figure 7. Performance comparison of CDP model on Amazon.

5.7. Validity Threats

While our experimental results demonstrate the effectiveness of the proposed method, we acknowledge potential threats to validity and discuss how they are mitigated.

Internal validity threats primarily concern whether the observed performance gains are genuinely attributable to our Bayesian reconstruction framework. To address this, we strictly adhere to the theoretical formulation of ϵ-LDP using the randomized response mechanism. All experiments are conducted under a simulated untrusted server setting where raw user data is never exposed, ensuring alignment with real-world LDP assumptions. Furthermore, we evaluate our method across multiple backbone models (LRMF and LGCN) and consistently observe performance improvements, which supports the causal role of our reconstruction module rather than model-specific artifacts.

External validity threats relate to the generalizability of our findings. Our experiments are based on three widely used public datasets (ML-100K, ML-1M, and Amazon) that span different domains, scales, and sparsity

levels—characteristics commonly found in industrial recommendation scenarios. The privacy budgets ($\epsilon \in [0.1, 1.0]$) cover a practical range from strong to moderate privacy guarantees. Although our current evaluation is simulation-based, the model-agnostic design of our framework suggests broad applicability. Future work will involve real-world deployment studies to validate scalability, latency, and robustness under dynamic user interactions and system constraints.

6. Conclusions and Future Work

This paper proposes a LDP recommendation algorithm based on Bayesian estimation. While LDP ensures a high level of user data protection, it inevitably introduces substantial noise that may degrade recommendation quality. To address this challenge, the proposed method employs Bayesian estimation to compute user–item interaction probabilities, reconstruct the interactions, and effectively mitigate the impact of noise, thereby preserving recommendation performance. Theoretical analysis demonstrates that the algorithm satisfies ϵ-LDP. Experimental results on three public datasets show that the proposed approach achieves a favorable balance between privacy protection and recommendation accuracy, providing an effective solution for privacy-preserving RSs.

Beyond theoretical contributions, our approach holds significant practical promise for real-world RSs—especially in privacy-sensitive industries such as finance, healthcare, and education. In these domains, user interaction data often contains highly sensitive behavioral signals, and regulatory frameworks impose strict limitations on raw data collection and centralized processing. Our method aligns naturally with such constraints: by ensuring that only locally perturbed data leaves the user device, it eliminates the need for a trusted central server while still enabling high-quality recommendations through Bayesian reconstruction on the backend. Moreover, the algorithm is lightweight on the client side (requiring only randomized response) and scalable on the server side (relying on efficient statistical estimation), making it suitable for deployment in large-scale industrial systems. For example, in mobile app recommendation or online banking services—where user trust and data minimization are paramount—our framework offers a viable path toward privacy-preserving personalization without sacrificing utility.

Despite its effectiveness, the proposed algorithm has some limitations. First, performance may be affected in highly sparse datasets or cold-start scenarios. Second, the computational cost of Bayesian reconstruction may increase for very large datasets. Future work will focus on improving prior estimation in sparse settings, reducing computational overhead, and extending the approach to multi-modal and cross-domain recommendation scenarios. We outline three concrete directions for future research. First, to improve prior estimation in sparse or cold-start scenarios, we plan to incorporate auxiliary information, such as user attributes or item metadata into the Bayesian prior. Second, to reduce computational overhead, we will explore efficient strategies like incremental Bayesian updates and distributed inference for large-scale deployment. Third, we aim to extend our framework to multi-modal and cross-domain recommendation settings, enhancing its applicability while preserving local privacy.

Author Contributions

H.T.: methodology, writing—reviewing and editing; Y.W.: conceptualization, methodology; B.L.: software, writing-original draft; J.D.: visualization, investigation; Z.Z: conceptualization, methodology. All authors have read and agreed to the published version of the manuscript

Funding

This work is supported by the National Natural Science Foundation of China (No. 72301050 and 62272077), the Science and Technology Research Program of Chongqing Municipal Education Commission (No. KJZD-M202400604 and KJQN202300605), and the China Postdoctoral Science Foundation (No. 2024M761257).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-assisted Technologies

During the preparation of this work, authors used OpenAI's ChatGPT to polish the language and improve readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Ge, Y.; Liu, S.; Fu, Z.; et al. A survey on trustworthy recommender systems. *ACM Trans. Recomm. Syst.* **2024**, *3*, 1–68.
2. Xu, L.; Zhang, J.; Li, B.; et al. Tapping the potential of large language models as recommender systems: A comprehensive framework and empirical analysis. *ACM Trans. Knowl. Discov. Data* **2024**, *19*, 105.
3. Alzahrani, A.; Asghar, M.Z. Maintaining user security in consumer electronics-based online recommender systems using federated learning. *IEEE Trans. Consum. Electron.* **2024**, *70*, 2657–2665.
4. Deng, J.; Wu, Q.; Wang, S.; et al. A novel joint neural collaborative filtering incorporating rating reliability. *Inf. Sci.* **2024**, *665*, 120406.
5. Rendle, S.; Freudenthaler, C.; Gantner, Z.; et al. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
6. Lin, S.; Zhou, S.; Chen, J.; et al. ReCRec: Reasoning the causes of implicit feedback for debiased recommendation. *ACM Trans. Inf. Syst.* **2024**, *42*, 158.
7. Zhu, H.; Xiong, F.; Chen, H.; et al. Incorporating a triple graph neural network with multiple implicit feedback for social recommendation. *ACM Trans. Web* **2024**, *18*, 23.
8. Liu, H.; Wang, Y.; Zhang, Z.; et al. Matrix factorization recommender based on adaptive Gaussian differential privacy for implicit feedback. *Inf. Process. Manag.* **2024**, *61*, 103720.
9. Luo, C.; Wang, Y.; Zhang, Y.; et al. Distributed differentially private matrix factorization for implicit data via secure aggregation. *IEEE Trans. Comput.* **2024**, *74*, 705–716.
10. Zhang, S.; Yin, H. Comprehensive privacy analysis on federated recommender system against attribute inference attacks. *IEEE Trans. Knowl. Data Eng.* **2022**, *36*, 987–999.
11. Zhu, Z.; Wu, C.; Fan, R.; et al. Membership inference attacks against sequential recommender systems. In *Proceedings of the ACM Web Conference 2023*; Association for Computing Machinery: New York, NY, USA, 2023.
12. Dwork, C.; McSherry, F.; Nissim, K.; et al. Calibrating noise to sensitivity in private data analysis. In *Third Theory of Cryptography Conference (TCC 2006)*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
13. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407.
14. McSherry, F.; Mironov, I. Differentially private recommender systems: building privacy into the Netflix prize contenders. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 627–636.
15. Xu, Z.; Chu, C.; Song, S. An effective federated recommendation framework with differential privacy. *Electronics* **2024**, *13*, 1589.
16. Deng, Y.; Zhou, W.; Haq, A.U.; et al. Differentially private recommender framework with dual semi-autoencoder. *Expert Syst. Appl.* **2025**, *260*, 125447.
17. Di Fazio, A. Enhancing privacy in recommender systems through differential privacy techniques. In Proceedings of the 18th ACM Conference on Recommender Systems, Bari, Italy, 14–18 October 2024; pp. 1348–1352.
18. Demelius, L.; Kern, R.; Trügler, A. Recent advances of differential privacy in centralized deep learning: A systematic survey. *Acm Comput. Surv.* **2025**, *57*, 158.
19. Guo, T.; Luo, J.; Dong, K.; et al. Locally differentially private item-based collaborative filtering. *Inf. Sci.* **2019**, *502*, 229–246.
20. Gao, C.; Huang, C.; Lin, D.; et al. DPLCF: differentially private local collaborative filtering. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 961–970.
21. Hu, W.; Fang, H. Towards differential privacy in sequential recommendation: a noisy graph neural network approach. *Acm Trans. Knowl. Discov. Data* **2024**, *18*, 125.
22. Ferrara, A.; Fazio, A.D.; Mancino, A.C.M.; et al. Enhancing utility in differentially private recommendation data release via exponential mechanism. In *European Conference on Information Retrieval*; Springer Nature: Cham, Switzerland, 2025; pp. 34–51.
23. Neera, J.; Chen, X.; Aslam, N.; et al. Private and utility enhanced recommendations with local differential privacy and Gaussian mixture model. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 4151–4163.

24. Sarkar, S.; Shinde, S.; Shedge, R. Elevating privacy in recommendation systems with hybrid noise in local differential privacy. In *World Conference on Artificial Intelligence: Advances and Applications*; Springer Nature: Singapore, 2024; pp. 241–256.
25. Müllner, P.; Lex, E.; Schedl, M.; et al. The impact of differential privacy on recommendation accuracy and popularity bias. In *European Conference on Information Retrieval*; Springer Nature: Cham, Switzerland, 2024; pp. 466–482.
26. Rezaimehr, F.; Dadkhah, C. A survey of attack detection approaches in collaborative filtering recommender systems. *Artif. Intell. Rev.* **2020**, *54*, 2011–2066.
27. Zhang, M.; Ren, Z.; Wang, Z.; et al. Membership inference attacks against recommender systems. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 864–879.
28. Himeur, Y.; Sohail, S.S.; Bensaali, F.; et al. Latest trends of security and privacy in recommender systems: A comprehensive review and future perspectives. *Comput. Secur.* **2022**, *118*, 102746.
29. Pan, Y.; Wu, Z.S.; Xu, H.; et al. Differentially private Bayesian persuasion. In *Proceedings of the ACM on Web Conference 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 1425–1440.
30. Fienberg, S.E. A brief history of statistical models for network analysis and open challenges. *J. Comput. Graph. Stat.* **2012**, *21*, 825–839.
31. Chatterjee, S.; Diaconis, P.; Sly, A. Random graphs with a given degree sequence. *Ann. Appl. Probab.* **2010**, *21*, 1400–1435.
32. Du, M.; Peng, J.; Hu, Y.;et al. Logistic regression matrix factorization recommendation algorithm for differential privacy. *J. Beijing Univ. Posts Telecommun.* **2023**, *46*, 115–120. (In Chinese)
33. He, X.; Deng, K.; Wang, X.; et al. LightGCN: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 639–648.
34. Ran, X.; Wang, Y.; Zhang, L.; et al. A differentially private nonnegative matrix factorization for recommender system. *Inf. Sci.* **2022**, *592*, 21–35.