引用格式：叶建梅, 杨久裕, 陈钱宏, 等. 结合混洗器的差分隐私矩阵分解推荐算法 [J]. 电子科技大学学报, 2025, 54(3): 432-441.
YE J M, YANG J Y, CHEN Q H, et al. Differential privacy matrix factorization recommendation algorithm combined with shuffler[J]. Journal of University of Electronic Science and Technology of China, 2025, 54(3): 432-441.

结合混洗器的差分隐私矩阵分解推荐算法

叶建梅[1]，杨久裕[2]，陈钱宏[2]，邓江洲[1]，王 永[1, 2*]

（1. 重庆邮电大学 电子商务与现代物流重点实验室，重庆 400065；2. 重庆邮电大学 计算机科学与技术学院，重庆 400065）

摘要：推荐系统需要利用大量用户数据进行运算，存在用户隐私泄露风险。虽然差分隐私技术已被用于保护用户隐私，但在不可信服务器场景下，现有方法由于过多噪声注入会导致推荐效果下降。针对此问题，提出了一种结合混洗器的差分隐私矩阵分解推荐算法，利用混洗操作的隐私放大效应来减少噪声。在此基础上，通过对本地最大的 k 个梯度添加噪音来避免因数据稀疏性引起的推荐性能下降的问题，从而更好地优化了隐私保护与数据效用之间的平衡。理论与实验结果均验证了该算法不仅能有效提升隐私保护力度，而且能够产生良好的推荐效果，展现出良好的应用潜力。

关键词：矩阵分解；差分隐私；混洗器；推荐系统
中图分类号：TP309.2 **文献标志码：**A DOI: 10.12178/1001-0548.2024081

Differential privacy matrix factorization recommendation algorithm combined with shuffler

YE Jianmei[1], YANG Jiuyu[2], CHEN Qianhong[2], DENG Jiangzhou[1], and WANG Yong[1, 2*]

(1. Key Laboratory of E-Commerce and Modern Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
2. College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Recommendation systems require extensive user data for computations, posing a risk to user privacy. While differential privacy techniques have been used to protect user privacy, in untrusted server scenarios, existing methods suffer from reduced recommendation effectiveness due to excessive noise injection. To address this issue, we propose a differential privacy matrix factorization recommendation algorithm that incorporates a shuffler to leverage the privacy amplification effect of shuffling operations for noise reduction. Additionally, we address the problem of recommendation performance degradation caused by data sparsity by adding noise to the top k gradients locally, thus achieving a better balance between privacy protection and data utility optimization. Theoretical and experimental results confirm that this algorithm not only effectively enhances privacy protection but also yields excellent recommendation results, demonstrating its promising application potential.

Key words: matrix factorization; differential privacy; shuffler; recommendation system

推荐系统在当今数字化时代具有极其重要的作用和广泛应用[1−3]。它利用历史数据可以帮助企业提高销售额、用户满意度和用户留存率，同时也可以帮助用户节省时间和精力，提高信息获取效率和准确性。然而，这些数据在采集、存储、使用或传播过程中的任何偏差或失误，都可能轻易触发用户隐私的泄露，甚至可能引发商业风险，造成经济损失。差分隐私技术为较好地保护推荐系统中的数据隐私提供了理论支撑。

在分布式环境中，差分隐私技术根据服务器是否可信被分为两大类：基于中心模型的数据隐私保护方案[4-6]和基于本地模型的数据隐私方案[7-11]。差分隐私技术已被应用于推荐系统中，通过在用户评分矩阵分解过程中引入差分隐私机制，来防止用户数据隐私的泄露。文献 [7] 提出了一种基于可信第三方的差分隐私矩阵分解（differential privacy matrix factorization, DPMF）模型，通过在目标函数中引入差分隐私技术，并根据用户划分添加噪声，实现

收稿日期：2024−04−09
基金项目：国家自然科学基金（62272077，72301050）；中国博士后科学基金（2021M702321）
作者简介：叶建梅，博士，主要从事数据分析、隐私保护与推荐系统等方面的研究。
*通信作者 E-mail: wangyong1@cqupt.edu.cn

了对用户数据隐私的保护。但是，DPMF 只对用户的评分进行了保护，对用户与项目之间交互的存在性没有进行保护。已有研究表明，推荐系统中用户与项目的交互存在性通常包含了用户的敏感信息，应予以保护。针对用户与项目之间交互的存在性，文献 [8] 提出了一种分布式的推荐框架，结合了两阶段随机响应和矩阵分解对其进行了隐私保护。通过降维技术和基于采样的二进制机制，文献 [9] 提出了一种基于差分隐私的梯度下降与降维（differentially private gradient descent with dimensionality reduction, Private GD-DR）模型，该模型不仅对用户评分值和评分存在性同时进行了隐私保护，而且还考虑了推荐的准确性。在综合考虑隐私保护和推荐性能的研究方面，文献 [10] 提出了一种基于局部差分隐私的高斯混合矩阵分解（local differentially private matrix factorization with mixture of Gaussian, BLP-MoG-MF）模型，该模型使用有界拉普拉斯机制（bounded Laplace mechanism, BLP）扰动用户的原始评分，以降低数据聚合器中的噪声，并使用高斯混合方法来估计添加到数据中的噪声，从而有效提高了推荐准确性。文献 [11] 提出了一种基于分段机制（piecewise mechanism, PM）的改进矩阵分解（improved matrix factorization based on piecewise mechanism, IMFPM）模型，该模型不仅保护用户的评分数据，还保护用户评分的项目集。通过分段机制和随机投影技术，该方法有效降低了隐私噪声对预测质量的影响，实验表明其在保护隐私的同时保持了高预测准确性。从已有的研究看，在服务器不可信的矩阵分解场景中，仍然存在以下问题：1）在相同的隐私预算下，相对于服务器可信的情景，需要加入更多的噪音；2）由于矩阵分解中项目隐因子维度较大且数据稀疏，过多无用数据上传会降低推荐系统的可用性。

针对以上问题，本文做出的改进是首先将混洗器引入矩阵分解模型中，通过混洗项目隐因子梯度，减少了隐私保护噪音的引入。其次是改进扰动对象的选择，仅对影响大的梯度进行扰动，更好地保留了数据效用，展现出良好的应用价值。

1　理论知识

1.1　混洗模型

假设混洗模型共有n个用户，其用户数据集$\boldsymbol{X}=(x_1,x_2,\cdots,x_n)\in\mathbb{X}^n$。混洗模型（如图 1 所示）的工作过程如下：首先，每个用户会使用一个映射$\mathcal{R}$，使之满足ϵ_l-LDP；然后，混洗器执行混洗操作s，将接收到的消息按照一个均匀随机的排列π进行混洗；最终，确保混洗后的消息$\mathcal{M}=s\circ\mathcal{R}^n$满足更强的$(\epsilon_c,\delta_c)$-DP[12]，其中$\epsilon_c\leqslant\epsilon_l$。

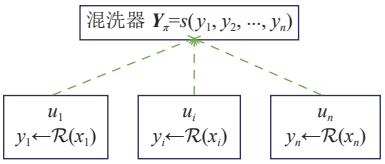

图 1　混洗模型

引理 1[13] 在混洗模型中，如果$\mathcal{R}$是ϵ_l-LDP，其中$\epsilon_l\leqslant\log(n/\log(1/\delta_c))/2$。则$\mathcal{M}$满足$(\epsilon_c,\delta_c)$-DP，其中：

$$\epsilon_c=O\left((1\wedge\epsilon_l)\boldsymbol{e}^{\epsilon_l}\sqrt{\log(1/\delta_c)/n}\right)\tag{1}$$

式中，符号$\wedge$表示取两个数中的较小值；n表示混洗用户的数量。

1.2　矩阵分解

假设推荐系统包含n个用户和m个物品。$\boldsymbol{R}=[r_{ij}]_{n\times m}$表示一个$n$行$m$列的评分矩阵，其中$r_{ij}$表示用户$i$对物品$j$的评分值。矩阵分解[14]是预测$\boldsymbol{R}$中未知评级值的一种最有效技术之一。在矩阵分解模型中，$\boldsymbol{R}$的评分值可以通过两个子矩阵$\boldsymbol{U}$和$\boldsymbol{V}$的内积来预测。这里，$\boldsymbol{U}\in R^{n\times f}$为用户隐因子矩阵，用户$i$的用户隐因子向量用$\boldsymbol{u}_i$表示；$\boldsymbol{V}\in\boldsymbol{R}^{m\times f}$为物品隐因子矩阵，物品$j$项目隐因子向量用$\boldsymbol{v}_j$表示；$f$为隐因子维度，且$f\ll\min(m,n)$。矩阵分解算法就是求解满足式 (2) 的最佳$U$和$V$:

$$\min_{\boldsymbol{U},\boldsymbol{V}}L(\boldsymbol{U},\boldsymbol{V})=\frac{1}{2M}\sum_{(i,j)\in O}\left(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j\right)^2+\frac{\lambda}{2}\left(\|\boldsymbol{u}_i\|^2+\|\boldsymbol{v}_j\|^2\right)\tag{2}$$

式中，O为观测到的用户–项目对(i,j)的集合；M为集合O中元素的个数，即$M=|O|$；$\lambda>0$为正规化参数；$\|\boldsymbol{u}_i\|^2$和$\left\|\boldsymbol{v}_j\right\|^2$分别表示$u_i$和$v_j$的正则化项。

1.3　差分隐私

1.3.1　本地差分隐私

定义 1[15]：给定n个用户，每个用户对应一条记录。给定一个隐私算法$\mathcal{M}$及其定义域$\mathrm{Dom}(\mathcal{M})$和值域$\mathrm{Ran}(\mathcal{M})$，若算法$M$在任意$t,t'\in\mathrm{Dom}(\mathcal{M})$上具有相同的输出结果$t^*\subseteq\mathrm{Ran}(\mathcal{M})$且满足以下不等式，则$\mathcal{M}$满足$(\epsilon,\delta)$-本地化差分隐私：

$$\Pr[\mathcal{M}(t)=t^*]\leqslant\boldsymbol{e}^{\epsilon}\times\Pr[\mathcal{M}(t')=t^*]+\delta\tag{3}$$

式中，ϵ为隐私预算，ϵ的值越小，隐私保护越强；δ代表接受隐私披露的概率。特别地，当$\delta=0$时，

机制$\mathcal{M}$是ϵ-LDP。

1.3.2 分段机制

为了满足局部差分隐私保护，文献 [16] 设计了一种有界值的扰动机制，即分段机制 PM，其定义如下。

定义 2[16]：对于输入值$x_i \in [-1,1]$，其扰动后的输出值x_i^*是从$[-C_p, C_p]$ 中采样得到，且：

$$C_p = \frac{\boldsymbol{e}^{\epsilon/2}+1}{\boldsymbol{e}^{\epsilon/2}-1} \tag{4}$$

同时，x_i^*的概率密度函数为：

$$\mathrm{pdf}(x^*=x|x_i) = \begin{cases} \boldsymbol{p} & x \in [b_1, b_2] \\ \dfrac{\boldsymbol{p}}{\boldsymbol{e}^{\epsilon}} & x \in [-C_p, C_p]\backslash[b_1, b_2] \end{cases} \tag{5}$$

其中：

$$\boldsymbol{p} = \frac{\boldsymbol{e}^{\epsilon}-\boldsymbol{e}^{\epsilon/2}}{2\boldsymbol{e}^{\epsilon/2}+2} \tag{6}$$

$$b_1 = \frac{C_p+1}{2}x_i - \frac{C_p-1}{2} \tag{7}$$

$$b_2 = b_1 + C_p - 1 \tag{8}$$

1.3.3 相关定理

组合属性在差分隐私的中心化模型与本地模型中均适用，并且在结合混洗器的场景中同样有效。

定理 1[17]：对于任意的$\epsilon \geqslant 0$，当t是正整数时，t次使用ϵ-DP机制的组合被称为t折自适应组合，在这种情况下，满足$t\epsilon$-DP。

定理 2[17]：对于任意的正数$\epsilon, \delta, \delta' > 0$，考虑$k$次相同的$(\epsilon,\delta)$-DP算法的自适应组合。则这$k$次自适应组合后满足$\left(\sqrt{2k\ln(1/\delta')}\epsilon + k\epsilon(\boldsymbol{e}^{\epsilon}-1), k\delta+\delta'\right)$-DP。

而一个随机从含有n条记录的数据库中不放回地抽取m个元素的机制，在定理 3 中导致了一种通过抽样来增强隐私的效果。

定理 3[18]：如果映射$\mathcal{M}: X^m \to Y$在大小为m的集合下满足(ϵ,δ)-DP，并且利用随机采样技术进行隐私增强，那么对于同样的映射$\mathcal{M}': X^n \to Y$，在大小为n的集合下，满足：

$$\left(\log\left(1+\frac{m}{n}(\boldsymbol{e}^{\epsilon}-1)\right), \frac{m}{n}\delta\right)\text{-DP}$$

2 结合混洗器的矩阵分解推荐算法

本文将混洗器引入到项目隐因子矩阵的梯度中，通过打乱用户梯度顺序实现隐私放大，进而构建了一种基于混洗器的矩阵分解推荐算法（shuffle model-based matrix factorization, MF-SM）。以此为基础，通过只对本地用户的 Top-k 梯度添加噪音，以进一步减少噪音引入，为此提出了结合混洗器和 Top-k 梯度的矩阵分解推荐算法（shuffle model-based top-k Gradient matrix factorization, TKMF-SM）。本文算法中使用的主要符号如表 1 所示。

表 1 文章算法主要符号说明

符号	含义
s	混洗器
S	服务器
n	用户的数量
m	项目的数量
T	迭代的数量
f	隐因子维度
d	$\nabla_{\boldsymbol{v}^i}$ 的维度，即$d = m \cdot f$
$\boldsymbol{U}$	用户隐因子矩阵
$\boldsymbol{V}$	服务器端项目隐因子矩阵
$\boldsymbol{v}^i$	用户i的项目隐因子矩阵
sk_S	服务器用于解密的私钥，仅服务器可知
pk_a	加密的公钥,所有参与者可知
E_{pk_a}	使用pk_a加密
D_{sk_S}	使用sk_S解密
$\nabla_{\boldsymbol{v}^i}$	用户i的本地梯度
$\nabla_{\tilde{\boldsymbol{v}}^i}$	$\nabla_{\boldsymbol{v}^i}$ 被裁剪和归一化后的梯度矩阵
$\boldsymbol{y}_i$	$\nabla_{\tilde{\boldsymbol{v}}^i}$ 被PM扰动的值

2.1 MF-SM 算法

MF-SM 算法主要包括 3 种参与者。

1）客户端（即每个用户）。主要负责初始化用户隐因子向量$\boldsymbol{u}_i$，在本地更新$\boldsymbol{u}_i$和$\nabla_{\boldsymbol{v}^i}$。其中，评分值$r_{ij}$在本地更新如式 (9)～式 (10) 所示：

$$\boldsymbol{u}_i \leftarrow \boldsymbol{u}_i - \eta\nabla_{\boldsymbol{u}_i} \tag{9}$$

$$\boldsymbol{v}_j \leftarrow \boldsymbol{v}_j - \eta\nabla_{\boldsymbol{v}_j} \tag{10}$$

式中，η为学习率；$\nabla_{\boldsymbol{u}_i}$和$\nabla_{\boldsymbol{v}_j}$分别为$\boldsymbol{u}_i$和$\boldsymbol{v}_j$的梯度。由式 (11)～式 (13) 计算得梯度$\nabla_{\boldsymbol{u}_i}$和$\nabla_{\boldsymbol{v}_j}$，即：

$$e_{ij} = r_{ij} - \boldsymbol{u}_i\boldsymbol{v}_j^{\mathrm{T}} \tag{11}$$

$$\nabla_{\boldsymbol{u}_i} = -2\boldsymbol{e}_{ij}\boldsymbol{v}_j + 2\lambda\boldsymbol{u}_i \tag{12}$$

$$\nabla_{\boldsymbol{v}_j} = -2\boldsymbol{e}_{ij}\boldsymbol{u}_i + 2\lambda\boldsymbol{v}_j \tag{13}$$

式中，e_{ij}代表真实值与预测值之间的误差。因为服务器端只更新项目隐因子矩阵$\boldsymbol{V}$，通过合并用户i的$\nabla_{\boldsymbol{v}_{j\in[m]}}$可以得到$\nabla_{\boldsymbol{v}^i}$。

为了保证$\nabla_{\boldsymbol{v}^i}$在 PM 机制的扰动范围，对$\nabla_{\boldsymbol{v}^i}$进行裁剪，并归一化，使梯度矩阵$\nabla_{\tilde{\boldsymbol{v}}^i}$满足差分隐私要求：

$$(\nabla_{\tilde{\boldsymbol{v}}^i})_{ij}=\begin{cases}-1 & (\nabla_{\boldsymbol{v}^i})_{ij}<-1\\ 1 & (\nabla_{\boldsymbol{v}^i})_{ij}>1\\ (\nabla_{\boldsymbol{v}^i})_{ij}/C & \text{其他}\end{cases}\tag{14}$$

式中，$(\nabla_{\tilde{\boldsymbol{v}}^i})_{ij}$和$(\nabla_{\boldsymbol{v}^i})_{ij}$分别表示$\nabla_{\tilde{\boldsymbol{v}}^i}$和$\nabla_{\boldsymbol{v}^i}$的第$i$行第$j$列；$C$表示对$(\nabla_{\boldsymbol{v}^i})_{ij}$裁剪阈值。

接着使用 PM 机制对$\nabla_{\tilde{\boldsymbol{v}}^i}$进行扰动后传递给混洗器，如式 (15) 所示：

$$(\boldsymbol{y}_i)_{ij}=\mathrm{PM}((\nabla_{\tilde{\boldsymbol{v}}^i})_{ij},\epsilon_{ld})\tag{15}$$

通过组合定理 1，可知 PM 机制满足ϵ_{ld}-LDP，表示在 1 次训练时对 1 个维度的隐私预算，其中$\epsilon_{ld}=\epsilon_l/dt$；$(\boldsymbol{y}_i)_{ij}$表示$\boldsymbol{y}_i$的第$i$行第$j$列。

为了保证混洗器不能聚合$\nabla_{\tilde{\boldsymbol{v}}^{i\in[n]}}$而服务器可以，取$\nabla_{\boldsymbol{v}^i}$的梯度索引值$\mathrm{id}\nabla_{\boldsymbol{v}^i}$，并且对$\boldsymbol{y}_i$使用公钥$\mathrm{pk}_a$进行加密，其公式如下：

$$m_i=\left\langle \mathrm{id}\nabla_{\boldsymbol{v}^i},E_{\mathrm{pk}_a}(\boldsymbol{y}_i)\right\rangle\tag{16}$$

2）混洗器s。主要负责对收到的$m_{i\in[n]}$使用置换π打乱其顺序，具体公式如下：

$$m_{\pi(i)\in[n]}=s(m_{i\in[n]})\tag{17}$$

3）服务器S。用于收集来自混洗器的$m_{\pi(i)\in[n]}$，使用保存在服务器端的私钥sk_S解密，如式 (18) 所示：

$$y_{\pi(i)\in[n]}=D_{\mathrm{sk}_S}(m_{\pi(i)\in[n]})\tag{18}$$

随后，对服务器S进行聚合与标准化，以及更新项目隐因子矩阵，分别为：

$$\bar{z}=\frac{1}{n}\sum_{i=1}^{n}\left\langle \mathrm{id}\nabla_{\boldsymbol{v}^i},y_{\pi(i)}\right\rangle\tag{19}$$

$$z=-C+(C/C_p)(\bar{z}+C)\tag{20}$$

$$\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{V}_{t-1}+z\tag{21}$$

MF-SM 的具体流程如算法 1 所示。

算法 1：MF-SM 算法

input: $S,s,E,n,T,\epsilon_l,C_p,\mathrm{pk}_a,\mathrm{sk}_S$

output: $\boldsymbol{V}^{\mathrm{T}}$

Initialize $\boldsymbol{U}$, $\boldsymbol{V}$ and a counter $t=1$

Publishing public key pk_a by server S

while $t\leqslant T$ do

　Send $\boldsymbol{V}^{t-1}$ of S to shuffler s

　Distribute $\boldsymbol{V}^{t-1}$ among n users by shuffler s

　for $i=1$ to n do

　　Initialize $\nabla_{\boldsymbol{u}_i}$ and $\nabla_{\boldsymbol{v}_j}$

　　Update $\boldsymbol{u}_i$ in (9)

　　Obtain $\nabla_{\tilde{\boldsymbol{v}}^i}$ by performing cropping and normalization $\nabla_{\boldsymbol{v}^i}$ in (14)

　　Introduce perturbation to $\nabla_{\tilde{\boldsymbol{v}}^i}$ by PM of (15)

　　Generate m_i using (16)

　end for

　Gather $m_{i\in[n]}$ from all users

　Derive $m_{\pi(i)\in[n]}$ by shuffling $m_{i\in[n]}$ in (17)

　Decrypt value $y_{\pi(i)\in[n]}=D_{\mathrm{sk}_S}(m_{\pi(i)\in[n]})$

　Aggregate value $\bar{z}=\frac{1}{n}\sum_{i=1}^{n}\left\langle \mathrm{id}\nabla_{\boldsymbol{v}^i},y_{\pi(i)}\right\rangle$

　Normalize $z=-C+(C/C_p)(\bar{z}+C)$

　Update $\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{V}_{t-1}+z$

end for

return $\boldsymbol{V}^{\mathrm{T}}$

2.2　TKMF-SM 算法

MF-SM 算法中将所有梯度加入噪音上传，没有考虑数据的稀疏性，可能会引入过多的噪音。为了避免此问题，本文提出 TKMF-SM 算法，该算法仅上传 Top-k 的$\nabla_{\boldsymbol{v}^i}$。TKMF-SM 算法在算法 1 的基础上，采用算法 2 生成本地m_i以及算法 3 混洗$m_{\pi(i)\in[n]}$。

MF-SM 是对每个维度引入噪音，其隐私预算为$\epsilon_{ld}=\epsilon_l/dt$，而 TKMF-SM 只是对 Top-$k$ 个维度添加噪音，其隐私预算为$\epsilon_{lk}=\epsilon_l/kt$。由于$k\ll d$，因此，在同等条件下，TKMF-SM 的隐私预算更大，具有更好的性能。

2.2.1　m_i的生成

m_i生成过程的算法思路如下。

1）选择l值以满足隐私要求

通过文献 [19] 选择适当的l值以满足隐私要求，其中$l\in\left[\max\frac{1}{v\beta},\frac{v}{v-1+\beta},\left\lceil\frac{1}{\beta}\right\rceil\right]$，$v$表示隐私效果，$\beta$表示采样率，即选取前$\beta$个梯度。

2）梯度选择与扰动

对于每个用户i，选择$\nabla_{\boldsymbol{v}^i}$中 Top-$k$ 对应的索引集合S_{top}，并在其余非S_{top}中选择$k(l-1)$个索引，得到另一个索引集合S_{non}。对于S_{top}中的梯度，本地使用满足差分隐私技术的 PM 机制进行扰动加噪，而对于S_{non}索引的值从均匀分布ω_{PM}中抽取进行填充。

3）生成排列列表

生成一个长度为lk的随机置换π_r，其元素从 1 到lk。

4）返回结果

kl个维度被排列成索引列表$\mathrm{id}\nabla_{\boldsymbol{y}^i}$和扰动值列表$\boldsymbol{y}_i$，通过式 (16) 加密成$l$个消息。

综上，m_i生成算法如算法 2 所示。

算法 2：m_i的生成

input: k, privacy effect v and a small fraction β

output: m_i

Choose a valid $l \in \left[\max \frac{1}{v\beta}, \frac{v}{v-1+\beta}, \left\lceil \frac{1}{\beta} \right\rceil\right]$

Top-k index set $S_{\text{top}} = \{j | j \in \text{Top}(\nabla_{\boldsymbol{y}^i})\}^k$

Non-Top index set

$S_{\text{non}} = \{j | j \in [d] \backslash S_{\text{Top}}\}^{k(l-1)}$

for $j \in S_{\text{top}} \cup S_{\text{non}}$ do

$$y_{i,j} = \begin{cases} \text{PM}_{\epsilon_{lk}} & j \in S_{\text{top}} \\ \omega_{\text{PM}} & j \in S_{\text{non}} \end{cases} \quad (22)$$

end for

Generate a permutation π_r over $[kl]$

Index list $\mathrm{id}\nabla_{\boldsymbol{y}^i} = \{d\pi_r(1), d\pi_r(2), \cdots, d_{\pi_r}(lk)\}$

Perturbed value list $\boldsymbol{y}_i = \{y_{\pi_r}(1), y_{\pi_r}(2), \cdots, y_{\pi_r}(lk)\}$

Generate $\langle \mathrm{id}\nabla_{\boldsymbol{y}^i}, \boldsymbol{y}_i \rangle$

Encrypt value $m_i = \{m_{(i-1)l+1}, \cdots, m_{(i-1)l+l}\}$ by performing $\left\langle \mathrm{id}\nabla_{\boldsymbol{y}^i}, E_{\text{pk}_a}(\boldsymbol{y}_i) \right\rangle$

return m_i

2.2.2 混洗过程

混洗器 s 的混洗过程的算法思路如下。

1）计算每个维度的填充数量

为了保护用户评分的存在性，使用文献 [19] 中虚假填充的方法。混洗器将每个维度$j \in [d]$都填充到相同的大小n_p，其中$n_n = n_p - n_s$，表示每个维度的填充数量，n_s表示每个维度n个用户上传数量。

2）生成填充向量的数量

为了保持每个发送消息的长度相同，使用$\sum_{j=1}^{d} n_{n,j} \big/ k$计算得到填充向量的数目，其中$k$表示每个填充消息的长度。

3) 生成填充消息

针对每个假向量，首先确定非零元素的索引集合S_{dummy}，然后生成长度为k的索引列表$\mathrm{id}\nabla_{\boldsymbol{y}^{n+u}}$和扰动值列表$\boldsymbol{y}_{n+u}$，以及对$\boldsymbol{y}_{n+u}$使用公钥$\text{pk}_a$加密得到$m_{n+u}$。

4）混洗和发送

将来自所有用户的$[nl]$个消息和混洗器的$[v]$个消息一起，生成一个$[nl+v]$的列表。通过混洗后发送给服务器。

综上，混洗器 s 混洗过程算法如算法 3 所示。

算法 3：混洗过程

input: m, n, f, d, n_p

output: $m_{\pi(i) \in [1, nl+v]}$

for $j \in [d]$ do

$$n_{s,j} \leftarrow \sum_{i=1}^{n} \mathbb{I}_{j \in \mathrm{id}\nabla_{\boldsymbol{y}^i}} \text{ where } \mathbb{I}_j \in \{0,1\}$$

$$n_{n,j} \leftarrow n_p - n_{s,j}$$

end for

Compute the number of dummy vector

$$v = \sum_{j=1}^{d} n_{n,j} / k$$

for $u \in [v]$ do

$S_{\text{dummy}} = \{j | j \in [d], n_{n,j} \neq 0\}$

$\mathrm{id}\nabla_{\boldsymbol{y}^{n+u}} = \{j | j \in S_{\text{dummy}}\}^k$

for $j \in \mathrm{id}\nabla_{\boldsymbol{y}^{n+u}}$ do

$n_{n,j} = n_{n,j} - 1$

end for

$\boldsymbol{y}_{n+u} = \{y^* | y^* \leftarrow \omega_{\text{PM}}\}^k$

$m_{n+u} = \left\langle \mathrm{id}\nabla_{\boldsymbol{y}^{n+u}}, E_{\text{pk}_a}(\boldsymbol{y}_{n+u}) \right\rangle$

end for

Generate a permutation π_r over $[nl+v]$

Shuffle and send $m_{\pi(i) \in [1, nl+v]}$ to server S

3 算法的效用分析

3.1 MF-SM 效用分析

定理 4：对于任何一个相邻数据集$X \cong_r X'$相差任意一个用户评分时，MF-SM 算法满足(ϵ_c, δ_c)-DP，其中：

$$\epsilon_c = (t\epsilon_{ct}) \wedge \left(\epsilon_{ct}\sqrt{2t\ln(1/\delta_{ct})} + t\epsilon_{ct}(e^{\epsilon_{ct}} - 1)\right) \quad (23)$$

$$\delta_c = \delta_{ct}(t+1) \quad (24)$$

式中，$\wedge$表示取两个数中的较小值。

证明：首先在一轮训练中，对于任意相邻的数据集$X \cong_r X'$相差任意一个梯度向量时，当$\epsilon_{ld} \leqslant \ln(n/\ln(1/\delta_{cd}))/2$时，通过引理 1 的混洗算法后满足$(\epsilon_{cd}, \delta_{cd})$-DP，且有：

$$\epsilon_{cd} = O\left((1 \wedge \epsilon_{ld})e^{\epsilon_{ld}}\sqrt{\ln(1/\delta_{cd})/n}\right) \quad (25)$$

式中，ϵ_{cd}是差分隐私参数；δ_{cd}代表接受隐私披露的概率；n是训练数据集的大小。此外，由于$\epsilon_{ld} = \epsilon_l/\mathrm{dt}$，通过引理 1 可得$\epsilon_{cd} < \epsilon_{ld}$。

其次，在一轮训练中，对于任意相邻的数据集$X \cong_r X'$相差任意一个用户评分时，通过定理 2 可推出算法满足$(\epsilon_{ct}, \delta_{ct})$-DP，其中：

$$\epsilon_{ct}=(d\epsilon_{cd})\wedge\left(\epsilon_{cd}\sqrt{2d\ln(1/\delta_{cd})}+d\epsilon_{cd}(\boldsymbol{e}^{\epsilon_{cd}}-1)\right) \quad (26)$$

$$\delta_{ct}=\delta_{cd}(d+1) \quad (27)$$

式中，ϵ_{ct}代表差分隐私参数；δ_{ct}代表接受隐私披露的概率；n代表训练数据集的大小。类似地，由定理 2 所示的组合定理可得$\epsilon_{ct}<d\epsilon_{cd}$。

最后，整个训练过程共包含t轮，反复利用定理 2 可得$\epsilon_c<t\epsilon_{ct}$且定理 4 的结论成立。证毕。

定理 4 定量地给出了 MF-SM 算法采用混洗操作后隐私预算的变化情况。同时，根据定理 4，容易得到如下的推论 1。

推论 1：对于 MF-SM 算法，当$\epsilon_l\leqslant dt\ln(n/\ln((d+1)(t+1)/\delta_c))/2$时，$\epsilon_l$隐私放大后的隐私预算为：

$$\epsilon_c=O\left((1\wedge\epsilon_l/dt)\boldsymbol{e}^{\epsilon_l/dt}\ln(dt/\delta_c)\sqrt{dt\ln(t/\delta_c)/n}\right) \quad (28)$$

式中，$\wedge$表示取两个数中的较小值；n为用户的数量；δ_c表示整个训练过程隐私泄露的概率上限。

推论 1 不仅给出了 MF-SM 算法进行混洗操作前后的隐私预算ϵ_c与ϵ_l之间的关系，而且根据推论 1 中的结果并结合引理 1 和定理 2，容易得出$\epsilon_c<\epsilon_l$。所以，MF-SM 算法有效实现了隐私放大的效果。

3.2　TKMF-SM 效用分析

定理 5：对于任何一个相邻数据集$X\cong_r X'$相差任意一个用户评分时，TKMF-SM 算法满足(ϵ_c,δ_c)−DP，其中：

$$\epsilon_c=(t\epsilon_{ct})\wedge\left(\epsilon_{ct}\sqrt{2t\ln(1/\delta_{ct})}+t\epsilon_{ct}(\boldsymbol{e}^{\epsilon_{ct}}-1)\right) \quad (29)$$

$$\delta_c=\delta_{ct}(t+1) \quad (30)$$

式中，$\wedge$表示取两个数中的较小值。

证明：首先在一轮训练中，对于任何一个相邻数据集$X\cong_r X'$相差一个梯度向量时，当$\epsilon_{lk}\leqslant\ln\left(n_p/\ln(1/\delta_{ck})\right)/2$，通过引理 1、定理 3，可得该训练满足$(\epsilon_{cd},\delta_{cd})-\mathrm{DP}$，其中：

$$\epsilon_{ck}=O\left((1\wedge\epsilon_{lk})\boldsymbol{e}^{\epsilon_{lk}}\sqrt{\ln(1/\delta_{ck})/n}\right) \quad (31)$$

$$\epsilon_{cd}=\ln\left(1+\beta(\boldsymbol{e}^{\epsilon_{ck}}-1)\right) \quad (32)$$

式中，由于$\epsilon_{lk}=\epsilon_l/kt$，结合引理 1 和定理 3 可得$\epsilon_{ck}<\epsilon_{lk}$。

其次，在一轮训练中，对于任何一个相邻数据集$X\cong_r X'$相差任意一个用户评分时，通过定理 2 可得该训练满足$(\epsilon_{ct},\delta_{ct})-\mathrm{DP}$，其中：

$$\epsilon_{\mathrm{ct}}=(2d\beta\epsilon_{cd})\wedge\left(\epsilon_{cd}\sqrt{4d\beta\ln(1/\delta_{cd})}+2d\beta\epsilon_{cd}(\boldsymbol{e}^{\epsilon_{cd}}-1)\right) \quad (33)$$

$$\delta_{ct}=\delta_{cd}(2d\beta+1) \quad (34)$$

且通过定理 2 所示的组合定理可得$\epsilon_{ct}<k\epsilon_{ck}$。

最后，由于整个训练过程包含t轮，因此通过反复使用定理 2 可以得到$\epsilon_c<t\epsilon_{ct}$且定理 5 所示的结论成立。证毕。

定理 5 定量地给出了 TKMF-SM 算法采用混洗操作和采样操作后隐私预算的变化情况。同时，根据定理 5 容易得到推论 2。

推论 2：对于 TKMF-SM 算法，当$\epsilon_l\leqslant\beta dt\ln\left(n_p/\ln(\beta(t+1)(\beta d+1)/\delta_c)\right)/2$时，$\epsilon_l$隐私放大后的隐私预算为：

$$\epsilon_c=O\Big((1\wedge\epsilon_l/\beta dt)\boldsymbol{e}^{\epsilon_l/\beta dt}\beta^{1.5}\times\sqrt{\ln(t/\delta_c)\ln(\beta^2dt/\delta_c)\ln(\beta dt/\delta_c)}\Big) \quad (35)$$

式中，$\wedge$表示取两个数中的较小值；n_p表示混合器S在每个维度上要打乱的用户的数量。

推论 2 给出了 TKMF-SM 算法进行混洗操作和采样操作前后的隐私预算ϵ_c与ϵ_l之间的关系。同样地，根据推论 2 中的结果并结合引理 1、定理 2 和定理 3，容易得出$\epsilon_c<\epsilon_l$。所以，TKMF-SM 算法有效实现了隐私放大的效果。

4　实验结果与分析

4.1　实验数据

实验中使用了 2 个公共数据集，分别是 MovieLens100K（ML-100K）和MovieLens1M（ML-1M），如表 2 所示。

表 2　实验数据集统计属性

属性名	Movielens100K	Movielens1M
用户数	943	6 039
项目数	1 682	3 705
密度/%	6.30	3.57

4.2　评估指标

实验中使用 RMSE（均方根误差）和 MAE（平均绝对误差）用作衡量方案性能的指标，如式 (36) 和式 (37) 所示：

$$\mathrm{RMSE}=\sqrt{\frac{1}{|N|}\sum_{(i,j)\in O}(r_{ij}-\hat{r}_{ij})^2} \quad (36)$$

$$\mathrm{MAE}=\frac{1}{|N|}\sum_{(i,j)\in O}\left|r_{ij}-\hat{r}_{ij}\right| \quad (37)$$

式中，O为测试集合；N为测试集合中的评分个数；(i,j)为测试集中的一个用户和物品对；r_{ij}为实际评分值；$\hat{r}_{ij}$为预测评分值。

RMSE 表示实际评分值与预测评分值之间的平均误差的均方根，MAE 表示实际评分值与预测评分值之间的平均绝对误差。RMSE 和 MAE 值越小，则表明推荐系统的准确性越高。

4.3 模型的有效性分析

本文提出的 TKMF-SM 算法首先采用混洗操作来提高对用户数据的隐私保护能力，随后采用 Top-k 梯度添加噪音以提升该算法的推荐效果。为了更好地解释混洗机制和对本地用户的 Top-k 梯度添加噪音对改善预测性能的有效性，分别对构建 TKMF-SM 算法的两种变体以及选择本文提出的 MF-SM 算法进行对比分析。具体如下。

1）带噪声的 MF 算法：其算法框架与 TKMF-SM 算法的框架结构相同，但在整个算法中不采用混洗机制放大隐私，只对本地用户数据的 Top-k 梯度添加噪声的矩阵分解算法，用于验证本文所提出的 TKMF-SM 算法中混洗机制的有效性。为了区别于传统 MF 算法，将此算法简记为 TKMF 算法。

2）Shuffler mechanism and random-k gradients-based matrix factorization（RKMF-SM）算法：是指随机选择本地用户项目梯度矩阵$\boldsymbol{V}$中的 k 个梯度进行上传替代最大 k 个梯度进行上传。该算法用于验证通过随机采样本地用户的 Top-k 个维度添加噪音能更好地提升推荐结果的准确性。

3）MF-SM 算法：即本文中提出的算法 1，通过对用户项目梯度矩阵$\boldsymbol{V}$中的每个梯度都加入噪音，验证随机采用 Top-k 个维度添加噪音的预测效果更佳。

在模型的有效性分析实验中，相应的参数设置为：$\eta=0.03$，$\lambda=10^{-7}$，隐因子维度$f=7$，迭代次数 $t=20$，$\epsilon\in[0.05,1.6]$。对于 TKMF、MF-SM、RKMF-SM 和 TKMF-SM4 个算法，设定$\delta_c=10^{-5}$；除了 MF-SM 之外，其他方法的采样率$\beta=0.02$；在数据集 ML-100K 和 ML-1M 中，n_p的值分别为 600 和 3 000。

由于隐私预算ϵ对算法性能具有重要影响，因此，通过对ϵ取不同值时，测试算法性能的变化。同时为了确保算法有较高的隐私保护强度，将ϵ的取值限定在较小的范围。图 2～图 5 展示了上述 4 种算法在不同数据集、不同ϵ取值下所有方案的 RMSE 和 MAE 值。总体而言，随着ϵ增大，各算法在不同数据集上的 RMSE 与 MAE 值逐渐减小。这符合差分隐私的规律，即随着隐私预算增加，扰动减少，数据效用增加，推荐系统性能提升。

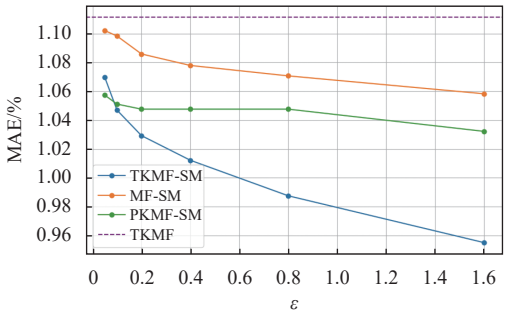

图 2　在 ML-100k 数据集上不同方案的 MAEs

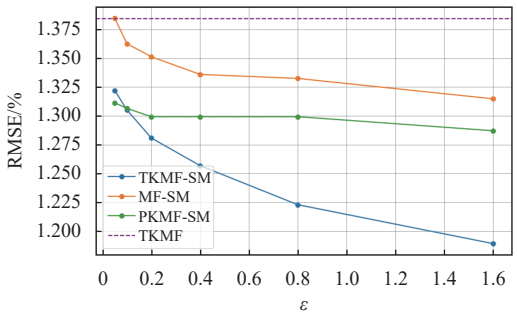

图 3　在 ML-100k 数据集上不同方案的 RMSEs

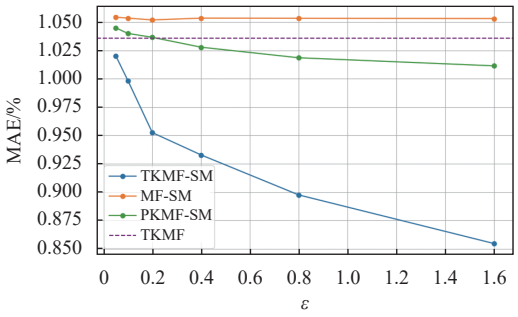

图 4　在 ML-1M 数据集上不同方案的 MAEs

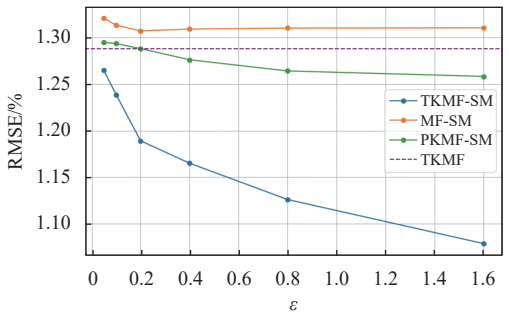

图 5　在 ML-1M 数据集上不同方案的 RMSEs

图 2～图 5 表明，TKMF-SM 算法在数据集 ML-100K 和 ML-1M 上的预测精度明显优于 TKMF

算法，这表明通过引入混洗操作来放大隐私，能够有效保护用户数据隐私，同时保证数据效用。由于未采用隐私放大技术的 TKMF 算法的预测准确度随隐私预算的变化较小，因此，本文提出的 TKMF-SM 算法在利用混洗机制来保护用户隐私、保证数据效用方面是有效的，从而提升了算法的预测效果。

同时 RKMF-SM 算法在数据集 ML-100K 和 ML-1M 上的预测精度上略优于 MF-SM 算法，这说明采用随机采样 k 个梯度添加噪声的方式有助于提升算法的推荐效果。此外，TKMF-SM 算法在数据集 ML-100K 和 ML-1M 上的预测精度上远优于 RKMF-SM，表明采用本地用户的 Top-k个梯度添加噪声，比随机采样本地用户的 k 个梯度添加噪声的策略具有更好的预测性能。由于推荐系统中数据普遍具有稀疏性，矩阵分解类算法在其训练过程中对学习效果起决定作用的主要是梯度变化较大的分量，因此，本文 TKMF-SM 算法通过上传本地用户的梯度变化较大的分量能够有效控制隐私保护噪音的引入，进而提升算法的预测效果。

此外，随着ϵ增大，本文 TKMF-SM 算法的 RMSE 和 MAE 与 TKMF 之间的差异逐渐减小，尤其在数据集 ML-1M 中二者的差异很小。这说明尽管本文 TKMF-SM 算法为了隐私保护在计算中引入噪声，但对数据效用影响较小，特别是随着数据集规模增大，影响更小，很好地实现了数据隐私保护和有效发挥数据效用之间的平衡。

在混洗过程中，为了保护评分的存在性，引入了填充数据。显然填充数据的占比对模型的性能有重要影响。为此，变化填充数据的占比即n/n_p，测试它对模型预测性能的影响。在测试过程中，TKMF-SM 算法的参数设为定$\delta_c = 10^{-5}$，采样率$\beta = 0.02$，对数据集 ML-100K 和 ML-1M 分别设置ϵ_{lk}为 0.5 和 0.2，迭代次数$t = 20$，这相关测试结果如图 6～图 9 所示。总体而言，随着比值n/n_p的增加，加入的假数据逐渐减少，故模型性能逐步提升。在数据集 ML-100K 中，这种现象尤为明显，原因在于该数据集规模较小，假数据对模型的影响较大。在数据集 ML-1M 中，由于其规模更大，可以更好地消除噪音的影响，因此，相对数据集 ML-100K 有更好的性能。在混洗过程中，当参与的用户数量较多时其隐私放大的效果会更为显著，即使在较小的n/n_p比值下，模型也能够表现出较好的效果。因而，可以将n/n_p作为平衡隐私预算和模型性能的一个关键因素。

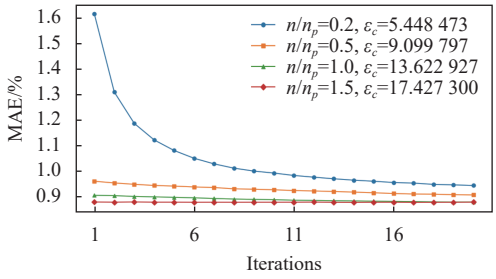

图 6　ML-100K 数据集下不同 n/n_p 比值的 MAEs

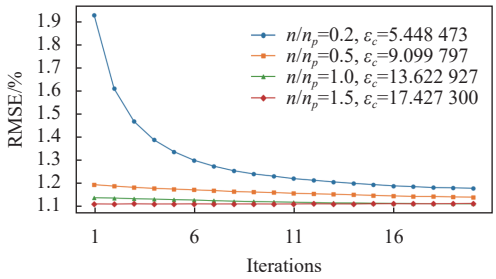

图 7　ML-100K 数据集下不同 n/n_p 比值的 RMSEs

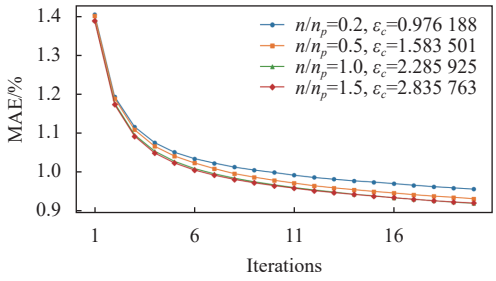

图 8　ML-1M 数据集下不同 n/n_p 比值的 MAEs

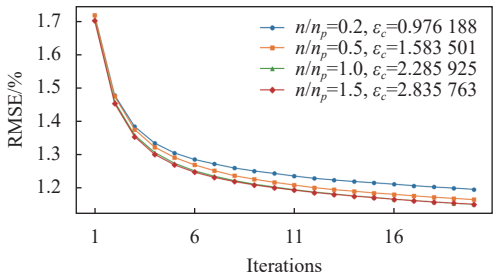

图 9　ML-1M 数据集下不同 n/n_p 比值的 RMSEs

4.4　对比分析

为了分析 TKMF-SM 算法的预测精度，本文选取典型的基于矩阵分解框架的算法进行对比分析，包括 Private-avgSVD 算法[19]、Private GD-DR 算法[9]、LTPS 算法[8]，具体描述如下：1）Private-avgSVD 算法是结合了 MF 和平均 SVD 的一种模型，它将数据集划分为敏感数据集和非敏感数据集，仅向用户的敏感数据添加噪声；2）Private GD-DR 算法在矩阵分解过程中通过对用户发送的

梯度矩阵进行二值扰动来实现隐私保护，并同时考虑了对用户评分和存在性均进行隐私保护；3）LTPS 算法在进行矩阵分解过程中为了提升模型的性能，采用了两阶段的处理方式，即在第 1 阶段对用户评分的存在性进行隐私保护，在第 2 阶段对用户评分值进行隐私保护。

本文提出的 TKMF-SM 算法与 3 个对比算法在隐私保护范围方面的对比结果，如表 3 所示。

表 3　隐私保护方案对比

对比算法	评分值	评分的存在性
TKMF-SM	√	√
Private GD-DR	√	√
LTPS	√	√
Private-avgSVD	√	×

表 3 表明 TKMF-SM、PrivateGD-DR 和 LTPS 对评分值和评分的存在性均进行了隐私保护，而 Private-avgSVD 仅对评分值进了隐私保护，未对评分的存在性进行隐私保护。因此，从隐私保护全面性看，本文提出的 TKMF-SM 算法、PrivateGD-DR 算法、LTPS 比 Private-avgSVD 具有更全面的隐私保护范围。

下面利用 MAE 和 RMSE 值来对比本文提出的 TKMF-SM 算法与其他 3 个算法的预测精度。设定隐私预算$0<\epsilon\leqslant 1.6$，这些算法在不同数据集中的 MAE 和 RMSE 值如图 10～图 13 所示。从图 10～图 13 可知，相较于 Private GD-DR 和 LTPS，本文提出的 TKMF-SM 算法具有更小的 MAE 和 RMSE 值，这表明 TKMF-SM 算法具有更好的预测性能。此外，随着隐私预算ϵ的增加，TKMF-SM 算法的 MAE 和 RMSE 值的波动程度远小于 Private GD-DR 和 LTPS，这说明即使在较小的隐私预算下，TKMF-SM 算法仍具有良好的预测准确性，更适用于严格隐私保护要求的场景。

在数据集 ML-100K 上，图 10 和图 11 显示 Private-avgSVD 的预测精度优于本文 TKMF-SM 算法。这是由于 Private-avgSVD 算法与 TKMF-SM 算法的隐私保护范围不同，Private-avgSVD 仅对用户评分值进行隐私保护，其保护的数据范围更小，故所需要的隐私噪音更少。在数据集 ML-1M 上，图 12 和图 13 显示 Private-avgSVD 的预测精度劣于本文 TKMF-SM 算法。这是因为在用户数量更大的数据集 ML-1M 上，混洗模型的隐私放大效果会显著增强。综上所述，随着用户数据量的增加，混洗模型的隐私放大效果也会增强，从而减少噪声的引入，进而提升算法的预测精度。故本文 TKMF-SM 算法对大规模数据集具有更好的适应性，展现出良好的应用潜力。

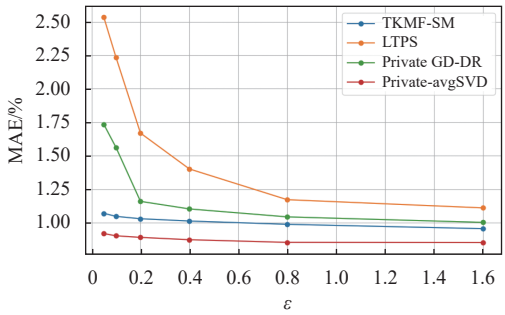

图 10　ML-100K 数据集上对比试验方案的 MAEs

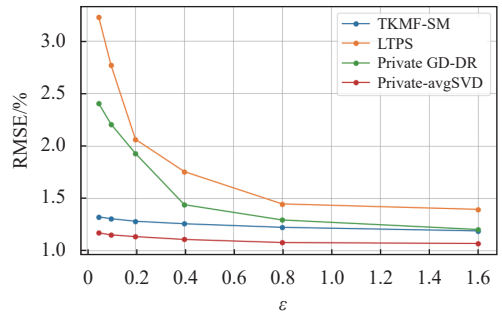

图 11　ML-100K 数据集上对比试验方案的 RMSEs

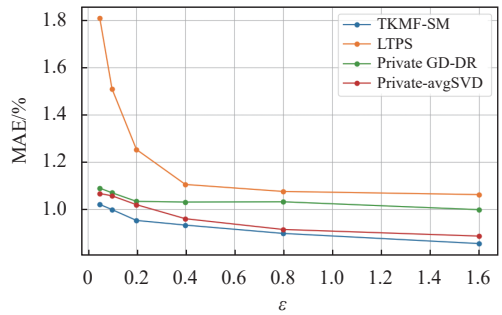

图 12　ML-1M 数据集上对比试验方案的 MAEs

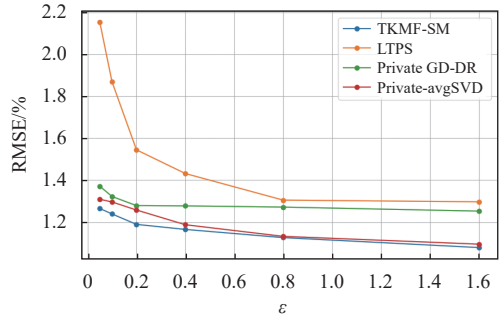

图 13　ML-1M 数据集上对比试验方案的 RMSEs

5　结束语

本文提出了基于混洗模型的矩阵分解推荐算

法，通过利用混洗模型进行隐私放大，提高相同噪音下的隐私保护效果。同时，还提出通过采样本地梯度 TOP-k 值并上传策略，进一步提升了推荐系统的可用性。针对提出的模型，从理论上就混洗操作和采样操作所产生的隐私预算的放大效果给出了定量的分析结果。最后，在两个标准数据集上对所提算法的性能进行了实验验证。实验结果显示，本文算法能更好地平衡隐私性和推荐准确性，特别是在隐私保护需求较高时仍能取得良好的推荐效果。因此，本文算法能够兼顾数据隐私保护和推荐的结果，在典型的本地化数据隐私保护场景中如社交平台的好友推荐、电子商务平台的商品推荐中具有较高的应用潜力。

参 考 文 献

[1] CHEN L, XU Y J, XIE F F, et al. Data poisoning attacks on neighborhood‐based recommender systems[J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(6): e3872.

[2] BARATHY R, CHITRA P. Applying matrix factorization in collaborative filtering recommender systems[C]// Proceedings of the International Conference on Advanced Computing and Communication Systems. New York: IEEE, 2020: 635-639.

[3] 李昌兵, 陈思彤, 罗陈红, 等. 基于物品交互约束的自编码器推荐模型[J]. 重庆邮电大学学报 (自然科学版), 2024, 36(5): 1052-1061.
LI C B, CHEN S T, LUO C H, et al. Item-interaction constraint-based autoencoder model for recommendation[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2024, 36(5): 1052-1061.

[4] ZHU T Q, REN Y L, ZHOU W L, et al. An effective privacy preserving algorithm for neighborhood-based collaborative filtering[J]. Future Generation Computer Systems, 2014, 36: 142-155.

[5] YANG J, LI X Y, SUN Z L, et al. A differential privacy framework for collaborative filtering[J]. Mathematical Problems in Engineering, 2019, S1: 1460234.

[6] RAN X, WANG Y, ZHANG L Y, et al. A differentially private nonnegative matrix factorization for recommender system[J]. Information Sciences, 2022, 592: 21-35.

[7] HUA J, XIA C, ZHONG S. Differentially private matrix factorization[C]//Proceedings of the AAAI Conference on Artificial Intelligence. California: AAAI Press, 2015: 1763-1770.

[8] JIANG J Y, LI C T, LIN S D. Towards a more reliable privacy-preserving recommender system[J]. Information Sciences, 2019, 482: 248-265.

[9] SHIN H J, KIM S, SHIN J, et al. Privacy enhanced matrix factorization for recommendation with local differential privacy[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1770-1782.

[10] NEERA J, CHEN X M, ASLAM N, et al. Local differentially private matrix factorization with MoG for recommendations[C]//Proceedings of IFIP Annual Conference on Data and Applications Security and Privacy. Regensburg: Springer, 2020: 208-220.

[11] WANG Y, GAO M X, RAN X, et al. An improved matrix factorization with local differential privacy based on piecewise mechanism for recommendation systems[J]. Expert Systems with Applications, 2023, 216: 119457.

[12] LIU R X, CAO Y, CHEN H, et al. Flame: Differentially private federated learning in the shuffle model[C]//Proceedings of the AAAI Conference on Artificial Intelligence. California: AAAI Press, 2021: 8688-8696.

[13] BALLE B, BELL J, GASCóN A, et al. The privacy blanket of the shuffle model[C]//Proceedings of Annual International Cryptology Conference on Advances in Cryptology-Crypto 2019. Regensburg: Springer, 2019: 638-667.

[14] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[15] 叶青青, 孟小峰, 朱敏杰, 等. 本地化得分隐私研究综述[J]. 软件学报, 2018, 29(7): 1981-2005.
YE Q Q, MENG X F, ZHU M J, et al. Survey on local differential privacy[J]. Journal of Software, 2018, 29(7): 1981-2005.

[16] WANG N, XIAO X, YANG Y, et al. Collecting and analyzing multidimensional data with local differential privacy[C]//Proceedings of the International Conference on Data Engineering. New York: IEEE, 2019: 638-649.

[17] DWORK C, ROTHBLUM G N, VADHAN S. Boosting and differential privacy[C]//Proceedings of Annual Symposium on Foundations of Computer Science. New York: IEEE, 2010: 51-60.

[18] BALLE B, BARTHE G, GABOARDI M. Privacy amplification by subsampling: Tight analyses via couplings and divergences[J]. Advances in Neural Information Processing Systems, 2018, 31: 6277-6287.

[19] ZHENG X Y, GUAN M P, JIA X M, et al. A matrix factorization recommendation system-based local differential privacy for protecting users' sensitive data[J]. IEEE Transactions on Computational Social Systems, 2022, 10(3): 1189-1198.

编　辑　刘飞阳