满足差分隐私保护的矩阵分解推荐算法

王 永[1,2*]，冉 珣[1]，尹恩民[1]，王 利[1]

(1. 重庆邮电大学电子商务与现代物流重点实验室 重庆 南岸区 400065；2. 桂林电子科技大学广西密码学与信息安全重点实验室 桂林 541004)

【摘要】协同过滤推荐算法在工作过程中需要分析和使用大量的用户数据，存在个人隐私泄露的安全隐患。现有的大多数在推荐系统中实施隐私保护的方法，容易引入过大噪声，导致推荐质量下降。针对此问题，该文提出一种满足差分隐私保护的矩阵分解推荐算法。该算法首先将矩阵分解问题转化为两个交替进行的用户隐因子和项目隐因子优化问题，然后采用遗传算法对这两个优化问题进行求解。将增强指数机制融入到遗传算法的个体选择中，并基于寻找重要隐因子的思想设计了遗传算法的变异过程。理论分析和实验结果显示，该算法可以为用户数据提供良好的差分隐私保护，同时有效保证了推荐的准确性，在推荐系统中具有良好的应用价值。

关 键 词 协同过滤； 差分隐私； 遗传算法； 矩阵分解

中图分类号 TP309.2 **文献标志码** A doi:10.12178/1001-0548.2020359

Matrix Factorization Recommendation Algorithm for Differential Privacy Protection

WANG Yong[1,2*], RAN Xun[1], YIN En-ming[1], and WANG Li[1]

(1. Key Laboratory of E-Commerce and Modern Logistics, Chongqing University of Posts and Telecommunications Nan'an Chongqing 400065;
2. Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology Guilin 541004)

Abstract Collaborative filtering techniques require tremendous amount of personal data to provide personalized recommendation services, which has caused the rising concerns about the risk of privacy leakage. Most existed methods for implementing privacy protection in recommender systems are prone to introduce excessive noises, which significantly degrades the recommendation quality. To address this problem, a matrix factorization algorithm satisfying differential privacy is proposed. The method first converts the matrix factorization problem into two alternate optimization problems, in which user latent factors and item latent factors are optimized respectively. Then a genetic algorithm is introduced to solve these two optimization problems, in which the enhanced exponential mechanism is applied into the individual selection and a novel mutation operation is designed based on the idea of finding important latent factors. Theoretical analysis and experimental results show that the algorithm can not only provide strong differential privacy protection for user data, but also ensure the accuracy of recommendations. Therefore, it has good application value in recommender systems.

Key words collaborative filtering; differential privacy; genetic algorithm; matrix factorization

推荐系统是当前互联网商家为用户提供个性化信息服务的主要技术手段之一。协同过滤作为一类主流的推荐算法，它利用用户对项目的历史评价信息来预测用户对未知项目的好恶并据此进行推荐。协同过滤技术需要使用大量用户数据，存在用户个人隐私泄漏的风险[1]。在基于邻居的协同过滤技术中，攻击者可以通过追踪邻居用户的推荐列表变化，推测目标用户对项目的评分[2]；在基于矩阵分解的协同过滤技术中，由于分解所得的隐因子矩阵携带数据信息，可能被攻击者利用，通过重构攻击等方式推断出用户的评分数据[3-4]。遭泄露的评分可能被进一步用于推测出用户的性别、年龄等信息，侵犯用户隐私[5]。如果用户出于安全考虑拒绝提供部分信息，则可能会导致推荐系统性能下降，甚至无法提供个性化服务。因此，非常有必要在推荐系统中考虑对用户信息进行隐私保护。

文献 [6] 提出了差分隐私的定义，为在推荐系统中实施有效隐私保护提供了良好的理论基础。文

收稿日期：2020－09－23；修回日期：2021－03－21
基金项目：国家自然科学基金 (71901045)；教育部人文社科规划 (20YJAZH102)；重庆市教委科技基金 (KJQN201900649)
作者简介：王永 (1977－)，男，博士，教授，主要从事数据挖掘、隐私保护和信息安全等方面的研究. E-mail: wangyong1@cqupt.edu.cn

献 [7] 将差分隐私保护引入协同过滤技术中，通过扰动项目协方差矩阵实现差分隐私保护。文献 [8] 将差分隐私应用到基于邻居的协同过滤推荐算法中，通过在邻居选择和相似性度量过程中加入噪音，实现隐私保护。文献 [9] 提出了两种分别对原始评分和用户相似性度量过程添加 Laplace 噪音的隐私保护方案。

针对基于矩阵分解的推荐算法，文献 [10] 在考虑推荐系统不可信的情况下，扰动矩阵分解算法的目标函数，将实施了隐私保护的项目隐因子矩阵用于推荐任务。文献 [11] 假设用户有不同程度的隐私保护需求，基于概率矩阵分解提出一种个性化的差分隐私推荐算法。文献 [12] 通过对目标函数进行扰动，提出了基于联合优化的隐私矩阵分解方案。文献 [13-14] 将差分隐私保护应用到矩阵分解推荐算法中，设计了 3 种添加噪音的方式，即分别在输入信息中、训练过程中和输出信息中添加噪音。依据这种思想，文献 [15] 在 SVD++模型上设计了 3 种差分隐私保护模型。目前的工作大多通过对矩阵分解过程的各种结果 (如梯度、隐因子矩阵、目标函数) 加入噪声项以实现差分隐私保护，这类方案存在如下问题：1) 噪声较大。较高的隐私保护需求或敏感度会使噪声分布的方差增大，导致加入过大的噪声；2) 不具通用性。加噪方法可能导致最终解在有约束问题上不可行；3) 没有考虑隐因子的重要程度，影响了算法求解效率。

针对上述问题，本文将遗传算法引入矩阵分解任务，使得差分隐私保护可以通过扰动候选解的选择过程实现，而不依赖于上述加入噪声的方法[16]。此外，遗传算法中解的搜索将在可行域内进行，易于延伸到带约束的矩阵分解问题。然而，直接应用遗传算法存在如下困难：首先，矩阵分解属非凸问题且参数量大，求解难度高；其次，如何减小隐私保护机制引入的扰动也是重要挑战。为解决上述问题，本文改进了遗传算法的关键步骤，提出一种满足差分隐私保护的矩阵分解方案。本文的主要贡献为：1) 将矩阵分解转化为两个交替进行的用户隐因子和项目隐因子优化问题，有效克服了求解过程中存在的解空间高维性和优化中的非凸性问题。2) 考虑用户或项目对隐因子的不同偏重，重新设计了遗传算法的变异过程，提升解的搜索效率；在此基础上利用增强指数机制减轻了算法受扰动程度，更好地实现了隐私保护水平和算法效用之间的平衡。

1 理论知识

1.1 矩阵分解算法

矩阵分解是隐语义推荐模型的典型算法，它将用户和项目均映射到相同的 d 维隐因子空间中[17]。将用户 u 对应的隐因子向量表示为$\boldsymbol{P}_u \in \mathbb{R}^d$，将由所有用户的隐因子向量构成的矩阵表示为$\boldsymbol{P}$；将项目 i 的隐因子向量表示为$\boldsymbol{Q}_i \in \mathbb{R}^d$，将所有项目的隐因子向量构成的矩阵表示为$\boldsymbol{Q}$；则矩阵分解算法就是求解满足式 (1) 的最佳$\boldsymbol{P}$和$\boldsymbol{Q}$：

$$\min f(\boldsymbol{P},\boldsymbol{Q}) = \min \sum_{(u,i)\in\mathcal{K}} \left(r_{ui} - \boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2 \tag{1}$$

式中，r_{ui} 为用户评分矩阵$\boldsymbol{r}$中用户 u 对项目 i 的评分；$\mathcal{K}$为观测到的评分数据对应的用户–项目对(u,i)集合。假设$\boldsymbol{r}$中包含的用户数为 m，项目数为 n，则有$\boldsymbol{r} \in \mathbb{R}^{m\times n}, \boldsymbol{Q} \in \mathbb{R}^{n\times d}, \boldsymbol{P} \in \mathbb{R}^{m\times d}$，其中$d \ll m,n$。

1.2 差分隐私

差分隐私 (differential privacy, DP) 是一种新型隐私保护框架，通过添加可控的噪声到数据的统计结果中，保证隐私不被泄露且数据具有可用性。

定义 1 差分隐私 (DP)[6]：对于任意的邻近数据集 D 和D'至多相差一条数据，且随机算法 A 所有可能的输出$O \subseteq \mathrm{Range}(A)$，当且仅当满足不等式 (2) 时，$A$ 满足 ε-差分隐私：

$$\Pr[A(D) \in O] \leqslant \mathrm{e}^{\varepsilon} \Pr[A(D') \in O] \tag{2}$$

式中，ε 为隐私预算，当 ε 值越小时，隐私保护的需求水平越高。

1.3 指数机制

指数机制[18] 是一种实现差分隐私保护的技术手段，其定义如下。

定义 2 指数机制：设随机算法 M 的输入为数据集 D，输出为$\omega \in \Omega$。函数$Q(D,\omega) \to \mathbb{R}$为$\omega$的可用性函数。若算法 M 以正比于$\exp(\varepsilon Q(D,\omega)/\Delta)$的概率从$\Omega$中选择并输出$\omega$，则算法 M 提供 ε-差分隐私保护，称算法 M 为指数机制。其中，Δ为可用性函数$Q(D,\omega)$的阻尼因子，也称$Q(D,\omega)$的敏感度，表示单个数据的差异对$Q(D,\omega)$造成的最大影响。假设 D'与 D 为邻近数据集，Δ满足不等式：

$$\Delta \geqslant 2 \max_{\omega\in\Omega,D,D'} Q(D,\omega) - Q(D',\omega) \tag{3}$$

1.4 增强指数机制

文献 [16] 针对模型拟合问题设计了增强指数机制，与指数机制相比，增强指数机制的应用限于

可用性函数，具有特定形式：

$$f(D,\omega)=h(\omega)+\sum_{t\in D}q(t,\omega) \tag{4}$$

式中，D 是包含了 n 个元组的数据集；$\mathcal{T}$ 是任意元组 t 的取值范围；$q(t,\omega)$为元组拟合函数，表示模型对 D 中单个元组 t 的拟合程度；$h(\omega)$是独立于数据集 D 的函数。基于此可用性函数，增强指数机制的定义如下。

定义 3　增强指数机制 (enhanced exponential mechanism, EEM)：设随机算法 M 的输入为数据集 D，输出为$\omega\in\Omega$。算法 M 以正比于$\exp(\varepsilon f(D,\omega)/\Delta)$的概率从$\Omega$中选择并输出$\omega$，其中$f(D,\omega)$满足式 (4) 且$\Delta$满足不等式：

$$\Delta\geqslant\min\left\{\begin{array}{l}2\max\limits_{t,t'\in\mathcal{T},\omega\in\Omega}q(t,\omega)-q(t',\omega),\\ 2\max\limits_{t\in\mathcal{T},\omega,\omega'\in\Omega}q(t,\omega)-q(t,\omega')\end{array}\right\} \tag{5}$$

那么算法 M 提供 ε-差分隐私保护，称算法 M 为增强指数机制。

从式 (5) 可知增强指数机制与标准指数机制之间的区别是前者的阻尼因子Δ考虑了$q(t,\omega)$和$q(t,\omega')$之间的最大差异，这比较适合候选解集合中解之间变化程度比较小的情况，原因是此时$\max\limits_{t\in\mathcal{T},\omega,\omega'\in\Omega}q(t,\omega)-q(t,\omega')$可能取得较小值。本文提出的隐私保护方案将利用这一特点改善算法效用。

2　隐私遗传矩阵分解算法

2.1　算法总体流程

本文算法围绕推荐系统的评分矩阵分解展开，将隐因子矩阵$\boldsymbol{P}$和$\boldsymbol{Q}$的求解过程转化为两个交替进行的优化过程。在优化过程中使用遗传算法求解，并在求解过程中引入增强指数机制，进而使矩阵分解过程满足差分隐私保护。本文算法的总体流程如下：

1) 为提高评分预测准确性，对用户评分矩阵 $\boldsymbol{r}$ 进行预处理，即设边界参数为 B，将评分转化到$[-B,B]$的范围，得到新的用户评分矩阵$\boldsymbol{R}$。然后，对矩阵$\boldsymbol{R}$进行隐因子分解，即：

$$\boldsymbol{P},\boldsymbol{Q}=\arg\min_{\boldsymbol{P},\boldsymbol{Q}}\sum_{(u,i)\in\mathcal{K}}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2 \tag{6}$$

式中，R_{ui} 为 $\boldsymbol{R}$ 中用户 u 对项目 i 的真实评分。隐因子分解的目标是找到使预测评分与真实评分误差平方和最小的$\boldsymbol{P}$和$\boldsymbol{Q}$矩阵。

2) 将式 (6) 的目标问题转换成两类特征求解任务：1) 求解用户的隐因子向量；2) 求解项目的隐因子向量。即在求解$\boldsymbol{P}_u$时，将矩阵$\boldsymbol{Q}$看作常数，构建目标函数：

$$f_{\boldsymbol{Q}}^{u}(D_u,\boldsymbol{P}_u)=-\sum_{i\in I_u}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2=\sum_{t\in D_u}q(t,\boldsymbol{P}_u) \tag{7}$$

式中，$q(t,\boldsymbol{P}_u)=-\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2$是针对单个元组$t=(\boldsymbol{Q}_i,R_{ui})$的元组拟合函数，它表征在单个评分上的预测效果；$\boldsymbol{Q}_i=(Q_{i1},Q_{i2},\cdots,Q_{id})$表示项目 i 的隐因子向量；$D_u=\{(\boldsymbol{Q}_i,R_{ui})|i\in I_u\}$是关于用户 u 的二元组集合，I_u 为用户 u 评价过的项目集合。为保障评分预测的准确性，对隐因子$\boldsymbol{Q}_i$设置上下界：$|Q_{ik}|\leqslant 1$，$k\in\{1,2,\cdots,d\}$。

同理，在求解$\boldsymbol{Q}_i$时，保持$\boldsymbol{P}$矩阵不变，构建目标函数：

$$f_{\boldsymbol{P}}^{i}(\mathrm{D}_i,\boldsymbol{Q}_i)=-\sum_{u\in U_i}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2=\sum_{t\in D_i}q(t,\boldsymbol{Q}_i) \tag{8}$$

式中，$q(t,\boldsymbol{Q}_i)=-\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2$；$t=(\boldsymbol{P}_u,R_{ui})$；$\boldsymbol{P}_u=(P_{u1},P_{u2},\cdots,P_{ud})$为用户 u 的隐因子向量；$D_i=\{(\boldsymbol{P}_u,R_{ui})|u\in U_i\}$是关于项目 i 的二元组集合, U_i为评价过项目 i 的用户集合。对隐因子$\boldsymbol{P}_u$设置上下界：$|P_{uk}|\leqslant 1$，$k\in\{1,2,\cdots\cdots,d\}$。

3) 首先保持矩阵 $\boldsymbol{Q}$ 不变，使用 2.2 节设计的隐私遗传算法 (APrivGene) 为每个用户求解式 (7) 所示的优化问题，得到对应的用户隐因子，更新矩阵 $\boldsymbol{P}$。然后，保持矩阵 $\boldsymbol{P}$ 不变，同样使用 2.2 节设计的隐私遗传算法为每个项目求解式 (8) 所示的优化问题，得到对应的项目隐因子，更新矩阵 $\boldsymbol{Q}$。交替重复上述过程，持续优化 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 矩阵，直至达到最大迭代次数 T。

上述隐私遗传矩阵分解算法的伪代码如算法 1 所示，其中改进的隐私遗传算法 APriveGene 将在 2.2 节中进行详细说明。

算法 1　隐私遗传矩阵分解算法 (PGMF)

输入：用户评分矩阵 $\boldsymbol{r}$，用户集合 Users，项目集合 Items，迭代次数 T；

输出：$\boldsymbol{P}$, $\boldsymbol{Q}$；

$\boldsymbol{r}$ 矩阵做预处理得同型矩阵 $\boldsymbol{R}$，建立优化问题如式 (6)；

for t = 1 to T

　　for u in Users

　　构建目标函数$f_{\boldsymbol{Q}}^{u}$；

　　$D_u=\{(\boldsymbol{Q}_i,R_{ui})|i\in I_u\}$ //获取用户二元组集；

　　$\boldsymbol{P}_u=\mathrm{APrivGene}\left(D_u,f_{\boldsymbol{Q}}^{u}\right)$ //求解用户隐因子；

end for

for i in Items

构建子目标函数$f_{\boldsymbol{P}}^{i}$;

$D_i = \{(\boldsymbol{P}_u, R_{ui}) | u \in U_i\}$ //获取项目二元组集;

$\boldsymbol{Q}_i = \text{APrivGene}\left(D_i, f_{\boldsymbol{P}}^{i}\right)$ //求解项目隐因子;

end for

end for

return $\boldsymbol{P}$, $\boldsymbol{Q}$

2.2 改进的隐私遗传算法

本算法对文献 [16] 中的隐私遗传算法进行了改良，提出调整的隐私遗传算法 (adjusted private genetic algorithm, APrivGene)。使用 APrivGene 算法对式 (7) 和式 (8) 所示的优化问题进行求解，在选择阶段引入增强指数机制，实施对矩阵分解过程的隐私保护。按照执行顺序、从初始化、选择和变异 3 个方面介绍 APrivGene 算法。

初始化阶段：设置包括 ε 在内的各个控制参数。然后，随机生成 l 个 d 维的向量作为初始候选解集Ω，计算Ω中每个解的目标函数值$f(D,\boldsymbol{\omega})$作为遗传算法的适应度值。

选择阶段：以$f(D,\boldsymbol{\omega})$为可用性函数，使用$\varepsilon/2TG$作为选择操作的隐私预算，应用增强指数机制 EEM 以正比于$\exp(\varepsilon f(D,\boldsymbol{\omega})/2TG\Delta)$的概率从$\Omega$中挑选出$\boldsymbol{\omega}$。为了有效减轻选择阶段引入的扰动，只选出单个个体进行后续操作，之后将Ω置空，准备接纳新解。

变异阶段：为避免交叉操作造成敏感度过大，只使用了变异操作。为了改善寻优效率，采用全局搜索效率较高的柯西变异算子生成变异扰动，即从标准柯西分布$C(0,1)$中生成随机扰动。然后，以寻找重要程度最高的隐因子为目的，让变异操作对各个隐因子进行变化，且每次只在一个维度 k 上搜索。由于用户或项目对某隐因子的偏好可分为正负两类，对单个隐因子的扰动对应地被设计为正负两个方向。对每个维度进行上述变异，每次变异生成两个新解，加入Ω，最后形成新的候选解集。

生成新集合之后，为逐步减小搜索范围提高寻优效率，使用衰减因子β缩减变异步长η。然后，返回选择环节，进入下一轮循环。当达到最大迭代次数 G 时，使用 EEM 方式选出最终解$\boldsymbol{\omega}^*$。

上述改进的隐私遗传算法的伪代码如算法 2 所示。

算法 2　改进的隐私遗传算法 (APrivGene)

输入：D，即二元组集合D_u或D_i；f，即目标函数f_Q^u或f_P^i;

输出：隐因子向量$\boldsymbol{\omega}^* = (\omega_1, \omega_2, \cdots, \omega_d)$;

初始化算法中的控制参数：设置隐因子个数 d，隐私预算 ε，变异步长η，衰减因子$\beta < 1$，最大迭代次数 G，候选解集Ω的大小 l;

随机生成初始候选解集Ω

for $g = 1$ to $G - 1$ do

对每个$\boldsymbol{\omega} \in \Omega$，计算$f(D,\boldsymbol{\omega})$

$\boldsymbol{\omega} = \text{EEM}_f^{\varepsilon}(D)$ //使用增强指数机制选择个体;

将Ω置空;

for $k = 1$ to d do

$x = C(0,1)$ //按标准柯西分布抽取随机噪声;

$\boldsymbol{v}_1 = (\omega_1, \omega_2, \cdots, \omega_k + \eta x, \cdots, \omega_d)$//正方向变异;

$\boldsymbol{v}_2 = (\omega_1, \omega_2, \cdots, \omega_k - \eta x, \cdots, \omega_d)$//负方向变异;

$\Omega = \Omega \cup \{\boldsymbol{v}_1, \boldsymbol{v}_2\}$

end for

$\eta = \eta\beta$;

end for

对每个$\boldsymbol{\omega} \in \Omega$，计算$f(D,\boldsymbol{\omega})$;

$\boldsymbol{\omega}^* = \text{EEM}_f^{\varepsilon}(D)$;

return $\boldsymbol{\omega}^*$

在算法 2 中，为了发挥增强指数机制的作用，在每次迭代中需要根据当前候选解，求解增强指数机制中的阻尼因子。求解过程如 2.3 节所示。

2.3 阻尼因子求解

在求解隐因子向量时，根据候选集合中个体的适应值$f(D,\boldsymbol{\omega})$和隐私预算 ε，EEM 将按照如下的概率输出用户隐因子向量和项目隐因子向量：

$$\Pr\left[\text{EEM}_f^{\varepsilon}(D) = \boldsymbol{P}_u\right] \propto \exp\left(\varepsilon f_Q^u(D_u, \boldsymbol{P}_u)/2TG\Delta P_u\right)$$

$$\Pr\left[\text{EEM}_f^{\varepsilon}(D) = \boldsymbol{Q}_i\right] \propto \exp\left(\varepsilon f_P^i(D_i, \boldsymbol{Q}_i)/2TG\Delta_{Qi}\right)$$

数据集D_u或D_i中的元组 t 有 d+1 个属性，其中预处理后的评分数据R_{ui}在$[-B,B]$之间，$|P_{uk}| \leqslant 1$和$|Q_{ik}| \leqslant 1$，$k \in \{1,2,\cdots,d\}$，所以元组 t 的取值范围$\mathcal{T} = [-B,B] \times [-1,1]^d$。设$\Delta_{P_u}$为求解用户隐因子向量时的阻尼因子，$\Delta_{Q_i}$为求解项目隐因子向量时的阻尼因子，则根据增强指数机制的定义可得：

$$\Delta_{Pu} \geqslant \min\left\{\begin{array}{l}\Delta_1 = 2\max\limits_{t,t' \in \mathcal{T},\, P_u \in \Omega} q(t,\boldsymbol{P}_u) - q(t',\boldsymbol{P}_u), \\ \Delta_2 = 2\max\limits_{t \in \mathcal{T},\, P_u, P_u' \in \Omega} q(t,\boldsymbol{P}_u) - q(t,\boldsymbol{P}_u')\end{array}\right\} \quad (9)$$

$$\Delta_{Qi} \geqslant \min\left\{\begin{array}{l}\Delta_1 = 2\max\limits_{t,t' \in \mathcal{T},\, Q_i \in \Omega} q(t,\boldsymbol{Q}_i) - q(t',\boldsymbol{Q}_i), \\ \Delta_2 = 2\max\limits_{t \in \mathcal{T},\, Q_i, Q_i' \in \Omega} q(t,\boldsymbol{Q}_i) - q(t,\boldsymbol{Q}_i')\end{array}\right\} \quad (10)$$

为了求解Δ_{Pu}，需要分别求解Δ_1和Δ_2。对于Δ_1有：

$$\Delta_1 = 2\max_{t,t'\in\mathcal{T},P_u\in\Omega} q(t,\boldsymbol{P}_u)-q(t',\boldsymbol{P}_u)=$$

$$2\max_{P_u\in\Omega}\left(\max_{t\in\mathcal{T}} q(t,\boldsymbol{P}_u)-\min_{t\in\mathcal{T}} q(t,\boldsymbol{P}_u)\right)=$$

$$2\max_{P_u\in\Omega}\left(\max_{(R_{ui},Q_i)\in\mathcal{T}}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2-\min_{(R_{ui},Q_i)\in\mathcal{T}}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2\right)$$

由于$\min\limits_{(R_{ui},Q_i)\in\mathcal{T}}\left(R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right)^2=0$，所以：

$$\Delta_1 = 2\max_{P_u\in\Omega}\left(\max_{(R_{ui},Q_i)\in\mathcal{T}}\left|R_{ui}-\boldsymbol{P}_u^{\mathrm{T}}\boldsymbol{Q}_i\right|^2\right)\leqslant$$

$$2\max_{P_u\in\Omega}\left(B^2+\left(\sum_{k=1}^{d}|P_{uk}|\right)^2\right)$$

对于Δ_2有：

$$\Delta_2 = 2\max_{t\in\mathcal{T},P_u,P'_u\in\Omega} q(t,\boldsymbol{P}_u)-q(t,\boldsymbol{P}_u')\leqslant$$

$$2\max_{P_u,P'_u\in\Omega}\left(\begin{array}{l}\max\limits_{(R_{ui},Q_i)\in\mathcal{T}} 2R_{ui}\left(\boldsymbol{P}_u'^{\mathrm{T}}\boldsymbol{Q}_i-\boldsymbol{P}_u{}^{\mathrm{T}}\boldsymbol{Q}_i\right)+\\ \max\limits_{(R_{ui},Q_i)\in\mathcal{T}}\left(\left(\boldsymbol{P}_u{}^{\mathrm{T}}\boldsymbol{Q}_i\right)^2-\left(\boldsymbol{P}_u'^{\mathrm{T}}\boldsymbol{Q}_i\right)^2\right)\end{array}\right)$$

因为

$$\max_{(R_{ui},Q_i)\in\mathcal{T}}\left(\left(\boldsymbol{P}_u{}^{\mathrm{T}}\boldsymbol{Q}_i\right)^2-\left(\boldsymbol{P}_u'^{\mathrm{T}}\boldsymbol{Q}_i\right)^2\right)\leqslant$$

$$\max_{(R_{ui},Q_i)\in\mathcal{T}}\left(\sum_{k=1}^{d}\sum_{s=1}^{d}Q_{ik}Q_{is}\left(P_{uk}P_{us}-P_{uk}'P_{us}'\right)\right)\leqslant$$

$$\left(\sum_{k=1}^{d}\sum_{s=1}^{d}\left|P_{uk}P_{us}-P_{uk}'P_{us}'\right|\right)$$

所以，

$$\Delta_2\leqslant 2\max_{P_u,P'_u\in\Omega}$$

$$\left(2B\sum_{k=1}^{d}\left|P_{uk}-P_{uk}'\right|+\sum_{k=1}^{d}\sum_{s=1}^{d}\left|P_{uk}'P_{us}-P_{uk}'P_{us}'\right|\right)$$

综合Δ_1和Δ_2的上界，得到求解用户隐因子向量时，阻尼因子应满足的条件为：

$$\Delta_{P_u}\geqslant\min\left\{\begin{array}{l}2\max\limits_{P_u\in\Omega}\left(B^2+\left(\sum\limits_{k=1}^{d}|P_{uk}|\right)^2\right),\\ 2\max\limits_{P_u,P_u'\in\Omega}\left(\begin{array}{l}2B\sum\limits_{k=1}^{d}\left|P_{uk}-P_{uk}'\right|\\ +\sum\limits_{k=1}^{d}\sum\limits_{s=1}^{d}\left|P_{uk}P_{us}-P_{uk}'P_{us}'\right|\end{array}\right)\end{array}\right\}\tag{11}$$

同理可得求解项目隐因子向量时阻尼因子Δ_{Q_i}应满足的条件为：

$$\Delta_{Q_i}\geqslant\min\left\{\begin{array}{l}2\max\limits_{Q_i\in\Omega}\left(B^2+\left(\sum\limits_{k=1}^{d}|Q_{ik}|\right)^2\right),\\ 2\max\limits_{Q_i,Q_i'\in\Omega}\left(\begin{array}{l}2B\sum\limits_{k=1}^{d}\left|Q_{ik}-Q_{ik}'\right|\\ +\sum\limits_{k=1}^{d}\sum\limits_{s=1}^{d}\left|Q_{ik}Q_{is}-Q_{ik}'Q_{is}'\right|\end{array}\right)\end{array}\right\}\tag{12}$$

观察Δ_{P_u}和Δ_{Q_i}应满足的条件，可以发现Δ_2衡量的是候选解集中各隐因子向量之间的差异。在多数情况下$\Delta_1>\Delta_2$，这是因为随着 APrivGene 的迭代，$q(t,\boldsymbol{P}_u)-q(t,P_u')$或$q(t,\boldsymbol{Q}_i)-q(t,Q_i')$的值会逐渐减小，但$\Delta_1$的值并不会受到 APrivGene 迭代的影响。所以，随着 APrivGene 迭代次数增加，阻尼因子会减小，增强指数机制可以选择出更精确的解，从而有效保证算法的效用。

3　算法的分析

3.1　安全性分析

定理 1 算法 1 满足 ε-差分隐私。

证明：令D为数据集D_u或D_i，D'与 D 为其邻近数据集，t 和t'分别表示 D 与D'中相异的元组；令ω为隐因子向量$\boldsymbol{P}_u$或$\boldsymbol{Q}_i$，在应用 APrivGene 求解ω时，设 EEM 的隐私预算$\varepsilon'=\varepsilon/2TG$，$T$表示算法 1(PGMF) 中外循环的次数，$G$表示算法 2(APrivGene) 中的最大迭代次数。令Δ为 EEM 的阻尼因子Δ_{P_u}或Δ_{Q_i}，根据 2.3 节中式 (9) 和式 (10)，考虑以下两种情况：

当$\Delta\geqslant\Delta_1$时：

$$\Delta\geqslant 2\max_{t,t'\in\mathcal{T},\omega\in\Omega} q(t,\omega)-q(t',\omega)=$$

$$2\max_{D,D'} f(D,\omega)-f(D',\omega)$$

$$\frac{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D)=\omega\right]}{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D')=\omega\right]}=$$

$$\frac{\exp(\varepsilon' f(D,\omega)/\Delta)}{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon' f(D,\omega')/\Delta)}\Big/\frac{\exp(\varepsilon' f(D',\omega)/\Delta)}{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon' f(D',\omega')/\Delta)}=$$

$$\exp(\varepsilon'(f(D,\omega)-f(D',\omega))/\Delta)\times$$

$$\left(\frac{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon'(f(D',\omega')-f(D,\omega'))/\Delta)\exp(\varepsilon' f(D,\omega')/\Delta)}{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon' f(D,\omega')/\Delta)}\right)\leqslant$$

$$\exp(\varepsilon'/2)\exp(\varepsilon'/2)\left(\frac{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon' f(D,\omega')/\Delta)}{\sum\limits_{\omega'\in\Omega}\exp(\varepsilon' f(D,\omega')/\Delta)}\right)=\exp(\varepsilon')$$

当$\Delta \geqslant \Delta_2$时：

$$\Delta \geqslant 2 \max_{t\in\mathcal{T},\omega,\omega'\in\Omega} q(t,\omega)-q(t,\omega') \geqslant \max_{t,t'\in\mathcal{T},\omega,\omega'\in\Omega} (q(t,\omega)-q(t',\omega))-(q(t,\omega')-q(t',\omega')) = \max_{D,D',\omega,\omega'\in\Omega} (f(D,\omega)-f(D',\omega))-(f(D,\omega')-f(D',\omega'))$$

$$\frac{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D)=\omega\right]}{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D')=\omega\right]} = \frac{\exp(\varepsilon' f(D,\omega)/\Delta)}{\sum_{\omega'\in\Omega}\exp(\varepsilon' f(D,\omega')/\Delta)} \Big/ \frac{\exp(\varepsilon' f(D',\omega)/\Delta)}{\sum_{\omega'\in\Omega}\exp(\varepsilon' f(D',\omega')/\Delta)} \leqslant \frac{\exp(\varepsilon'(f(D,\omega)-f(D',\omega))/\Delta)}{\min_{\omega'\in\Omega}\exp(\varepsilon'(f(D,\omega')-f(D',\omega'))/\Delta)} = \max_{D,D',\omega,\omega'\in\Omega} \exp\left(\varepsilon'\begin{pmatrix}(f(D,\omega)-f(D',\omega))\\-(f(D,\omega')-f(D',\omega'))\end{pmatrix}/\Delta\right) \leqslant \exp(\varepsilon')$$

综上，由于$\Delta \geqslant \min\{\Delta_1,\Delta_2\}$，总有：

$$\frac{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D)=\omega\right]}{\Pr\left[\mathrm{EEM}_f^{\varepsilon'}(D')=\omega\right]} \leqslant \exp(\varepsilon')$$

故应用 APrivGene 算法求解隐因子向量时，其每一轮迭代均满足$\varepsilon/2TG$-差分隐私。由差分隐私保护的序列组合性质可得，更新每个用户或项目的隐因子向量时算法满足$\varepsilon/2T$-差分隐私，算法 1 满足ε-差分隐私。

3.2 效用分析

3.2.1 对问题转化的分析

本文算法将矩阵分解的求解转换为对两个优化问题的求解，这样处理有两点优势：

1) 更好地体现个性化的思想。因为直接求解式 (6) 可能忽视单个个体的推荐质量。转化为式 (7) 和式 (8) 所示的问题后，可以为每个用户或每个项目分别设计其专属的考虑隐私保护的隐因子值，更好地体现个性化的推荐思想，利于提升推荐精度。

2) 提升算法效率和效用。直接对原问题应用遗传算法求解，解的维度将是$d\times(m+n)$，而推荐系统中的用户数 m 和项目数 n 通常都很庞大。采用遗传算法在高维空间中寻优，将会导致效率非常低。同时，原问题关于 $\boldsymbol{P}$，$\boldsymbol{Q}$ 是非凸的，也会导致算法收敛速度慢。过慢的收敛速度，会导致迭代轮次增加。由于需要在每轮迭代中添加隐私保护的噪音，会导致噪声增大，从而使解的质量下降甚至不可用。本算法将原问题分解为两个优化问题，使得各个子问题都是凸问题，且解的维度是隐因子个数 d，它远小于 m 和 n，极大地提高了求解的效率，也利于提高解的效用。

3.2.2 改进隐私遗传算法的分析

APrivGene 算法是 PrivGene 算法的改进算法。PrivGene 算法并没有对变异操作进行专门的设计，它所采用的随机变异方式，将导致解的搜索效率不高，影响最终解的质量。APrivGene 算法在变异操作中，对选择的个体沿着解的各个维度，从正反两个方向使用标准柯西分布生成随机扰动进行变异，具有如下优势：

1) 有助于 EEM 选出更好的解。EEM 的特点是，当候选解之间的变动程度不大时，其敏感度将取得较小值从而减轻选择过程的扰动。单维度变异所生成的新解之间只存在一个隐因子上的差异，此时式 (9) 和式 (10) 中对于Δ_{Pu}和Δ_{Qi}通常有$\Delta_1>\Delta_2$。随着算法逐渐收敛，Δ_2的取值将更小，增强指数机制的阻尼因子减小，使得选中优质解的概率提高。

2) 有助于提高解的搜素效率并减少扰动。矩阵分解中用户和项目共享相同的隐因子，但不同的用户或项目对不同的隐因子会有不同程度的关注，单维度变异将有利于快速找到相对重要的隐因子。用户或项目对隐因子只有正向或负向两类偏好，变异算子在隐因子的正负方向上同时进行搜索，而非随机搜索，符合实际情况。该做法有效提升了解的搜索效率，同时控制了候选解之间的变动程度，减轻选择过程受到的扰动。

3) 标准柯西分布C(0,1)由于有较高的两翼概率特性，具有较好的全局搜索能力，能帮助算法在迭代的初期保持一定程度的多样性。设置了衰减因子β在每次迭代时对步长η进行缩减，利于在迭代后期增强指数机制实现更优的选择。因为随着迭代进行，式 (9) 和式 (10) 中Δ_{Pu}和Δ_{Qi}的值Δ_2会逐渐减小，但Δ_1的值并不会受到影响，这样增强指数机制的阻尼因子会减小，使选择过程受到更少的扰动，做出更优的选择。

4 实验结果与分析

4.1 实验数据

采用两个常用数据集 Movielens100K 和 YahooMusic 进行实验，按 8∶2 的比例随机划分为训练集和测试集。两个数据集的统计属性如表 1 所示。

表 1　实验数据集统计属性

属性名	Movielens100K	YahooMusic
用户数	943	8 089
电影数	1 682	1 000
密度/%	6.3	1.8
评分均值	3.529 9	2.632 1
评分方差	1.267 1	2.382 1
用户平均评分数	106	33
项目平均受评数	59.4	270.1

4.2　实验算法与评估指标

除本文算法外，还对其他一些类似算法进行了对比实验。实验中涉及到的算法及其描述如表 2 所示。

表 2　实验算法汇总

算法名称	描述
PGMF	本文算法
ALSBase[17]	不考虑差分隐私保护，运用交替最小二乘法(alternating least squares, ALS)求解矩阵分解的算法
DPSGD[14]	应用随机梯度下降法(stochastic gradient descent, SGD)求解矩阵分解，对梯度进行扰动，实施隐私保护
DPSGDInput[13]	对原始评分进行扰动之后运用SGD求解矩阵分解的算法
DPALS[14]	对ALS求解的结果进行扰动，实施隐私保护的算法
DPALSObj[19]	对ALS的目标函数进行扰动，实施隐私保护的算法

本文取 10 次实验的平均值作为最终结果。采用均方根误差 (RMSE) 度量算法的性能：

$$\mathrm{RMSE}=\frac{\sum_{i=1}^{T}(r_{ui}-\hat{r}_{ui})^2}{T}$$

式中，T 为有效预测项目的个数；r_{ui}为用户 u 对项目 i 的真实评分；$\hat{r}_{ui}$为用户 u 对项目 i 的预测评分。RMSE 越小则推荐精度越高。

4.3　实验结果

采用文献 [14] 中的预处理方式，将评分区间转换为 [−1,1]，设置隐因子变量域为 [−1,1]。在 APrivGene 中，最大迭代轮次为 23，候选集大小为 85，柯西变异算子的步长为 0.2，步长的衰减率为 0.95。对比算法的参数设置均遵循相应文献中的最优参数设置。

为了保证有效的隐私保护，实验中将隐私预算 ε 设置为较小范围，即$\varepsilon\in[0.1,1]$。图 1 和图 2 分别给出了本算法与其他对比算法在 Movielens100K 和 YahooMusic 两个数据集上的 RMSE 测试结果。其中，将不考虑隐私保护的 ALSBase 算法的实验结果作为对比基线。从整体上看，随着 ε 的增大，各个算法的 RMSE 均逐渐减小，表明随着隐私保护水平的下降，推荐准确性增加。各算法在 Movielens 100K 数据集上的推荐准确性均高于 YahooMusic 数据集，主要原因是 YahooMusic 数据集具有更高的稀疏性。

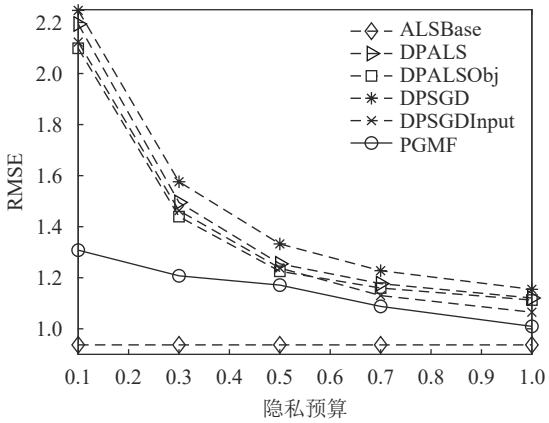

图 1　Movielens100K 数据集上的 RMSE 测试结果

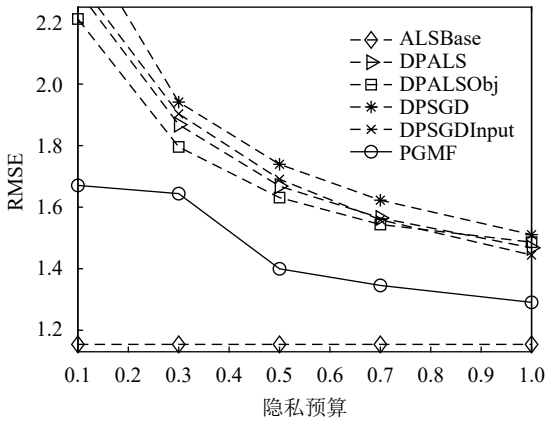

图 2　YahooMusic 数据集上的 RMSE 测试结果

在图 1 中，随着 ε 的变化，PGMF 在 Movielens 100K 数据集上的 RMSE 为：$0.995\leqslant\mathrm{RMSE}\leqslant1.308$，低于其他的隐私保护算法。同样的趋势也存在于 YahooMusic 数据集的测试中。在图 2 中，PGMF 的 RMSE 总是低于其他对比算法，其 RMSE 值的范围为$1.290\leqslant\mathrm{RMSE}\leqslant1.670$，比其他隐私保护算法平均低 0.2 左右，显示出了更好的准确性。在两个数据集上，PGMF 与不考虑任何隐私保护的 ALSBase 算法的 RMSE 差距是最小的，同样证明了 PGMF 具有更好的推荐准确性。

在本实验中，DPALS 算法的推荐准确性比 DPSGD 算法要高。因为在不考虑隐私保护的情况下，ALS 的性能比 SGD 要好，这种优越性在考虑差分隐私的情形下同样存在。但是，这两种方法都

是基于传统优化方式的算法，当隐私预算 ε 越小，DPSGD 和 DPALS 所引入的噪声就越大，导致求解出的隐因子向量与最优解之间差距过大，推荐准确度降低。在图 1 中，$\varepsilon = 0.1$时，DPALS 与 DPSGD 的 RMSE 都超过了 2.1，而 PGMF 的 RMSE 只有 1.3；在图 2 中，$\varepsilon = 0.1$时，DPALS 与 DPSGD 的 RMSE 都超过了 2.3，而 PGMF 的 RMSE 只有 1.67。比较结果说明在隐私保护要求较高时，PGMF 的优势更为明显。

DPSGDInput 算法是文献 [13] 中表现最优的算法，直接对评分数据添加噪音。它不需要在矩阵分解过程中分配隐私预算，在较低隐私保护需求下具有良好的推荐准确性。当 ε=1 时，其 RMSE 值在 Movielens100K 与 YahooMusic 数据集上分别为 1.06 和 1.44，是除 PGMF 算法以外最低的。但是，这种直接对数据集加噪音的方式在高隐私保护需求下会引入过大的噪声。从图 1 和图 2 中可以看出，在 ε<0.5 时，该算法的推荐 RMSE 值显著增加，其推荐准确性比 DPALSObj 算法和 PGMF 更差。

DPALSObj 算法通过对目标函数进行扰动而实现隐私保护。它的推荐精度在高隐私保护条件下，即$\varepsilon \in [0.1,0.5]$时，优于除 PGMF 之外的其他隐私保护算法。这种方法对隐私预算的大小比较敏感，在高隐私保护需求下相对于 PGMF 仍然引入了过大的噪声，即便在其表现更为突出的 YahooMusic 数据集上，其 RMSE 仍然明显比 PGMF 高。

PGMF 的性能优于其他算法的主要原因是采用了独特的进化方式限制了候选解集的方差，又借助增强指数机制改善了解的选择过程。所以，即使在很小的隐私预算条件下，求解出的隐因子向量都不会偏离最优解太远，实现了更高的推荐准确度。

5 结 束 语

本文针对推荐系统中的隐私问题提出了一种满足差分隐私保护的矩阵分解算法。该算法将矩阵分解问题转化为两个交替进行的优化问题。在遗传算法的选择操作中采用了增强指数机制使得整个矩阵因子分解的过程满足差分隐私保护。基于搜索重要隐因子的思想，设计了遗传算法的变异操作，从正反两个方向变异隐因子，不仅提高了算法的效率而且有效增强了解的性能。在两个标准数据集上的实验结果表明本文算法能更好地平衡隐私性和推荐的准确性，尤其在隐私保护需求较高的条件下，仍然可以取得良好的推荐效果，具有很好的应用潜力。

参 考 文 献

[1] KENTHAPADI K, MIRONOV I, THAKURTA A G. Privacy-preserving data mining in industry[C]//Proceedings of the 12th ACM International Conference on Web Search and Data Mining. [S.l.]: ACM, 2019: 840-841.

[2] CALANDRINO J A, KILZER A, NARAYANAN A, et al. " You might also like: " Privacy risks of collaborative filtering[C]//2011 IEEE Symposium on Security and Privacy. [S.l.]: IEEE, 2011: 231-246.

[3] FREDRIKSON M, JHA S, RISTENPART T. Model inversion attacks that exploit confidence information and basic countermeasures[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. [S.l.]: ACM, 2015: 1322-1333.

[4] WEINSBERG U, BHAGAT S, IOANNIDIS S, et al. BlurMe: Inferring and obfuscating user gender based on ratings[C]//Proceedings of the 6th ACM Conference on Recommender Systems. [S.l.]: ACM, 2012: 195-202.

[5] NIKOLAENKO V, IOANNIDIS S, WEINSBERG U, et al. Privacy-preserving matrix factorization[C]//Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. [S.l.]: ACM, 2013: 801-812.

[6] DWORK C. Differential privacy[C]//Proceedings of the 33rd Int Colloquium on Automata, Languages and Programming. Binlin: Springer, 2006: 1-12.

[7] MCSHERRY F, MIRONOV I. Differentially private recommender systems: Building privacy into the net[C]// Proceedings of the 2009 ACM SIGKDD Internationl Conference on Knowledge Discovery and Data Mining. New York: ACM, 2009: 627-636.

[8] ZHU Tian-qing, REN Yong-li, ZHOU Wan-lei, et al. An effective privacy preserving algorithm for neighborhood-based collaborative filtering[J]. Future Generation Computer Systems, 2014, 36: 142-155.

[9] YANG Jing, LI Xiao-ye, SUN Zhen-long, et al. A differential privacy framework for collaborative filtering[J]. Mathematical Problems in Engineering, 2019, DOI: 10.1155/2019/1460234.

[10] HUA Jing-yu, XIA Chang, ZHONG Sheng. Differential private matrix factorization[C]//Proceedings of the 24th International Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 1763-1770.

[11] ZHANG Shun, LIU Lai-xiang, CHEN Zhi-li, et al. Probabilistic matrix factorization with personalized differential privacy[J]. Knowledge-Based Systems, 2019, 183: 104864.

[12] ZHANG F, LEE V E, CHOO K K R. Jo-DPMF: Differentially private matrix factorization learning through joint optimization[J]. Information Sciences, 2018, 467: 271-281.

[13] FRIEDMAN A, BERKOVSKY S, KAAFAR M A. A differential privacy framework for matrix factorization recommender systems[J]. User Modeling and User-Adapted Interaction, 2016, 26(5): 1-34.

[14] BERLIOZ A, FRIEDMAN A, KAAFAR M A, et al. Applying differential privacy to matrix factorization[C]// Proceedings of the 9th ACM Conference on

Recommender Systems. [S.l.]: ACM, 2015: 107-114.

[15] 鲜征征, 李启良, 黄晓宇, 等. 基于差分隐私和 SVD++的协同过滤算法[J]. 控制与决策, 2019, 34(1): 43-54.
XIAN Zheng-zheng, LI Qi-liang, HUANG Xiao-yu, et al. Collaborative filtering via SVD++ with differential privacy[J]. Control and Decision, 2019, 34(1): 43-54.

[16] ZHANG Jun, XIAO Xiao-kui, YANG yin, et al. PrivGene: Differentially private model fitting using genetic algorithms[C]//Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. [S.l.]: ACM, 2013: 665-676.

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

[18] MCSHERRY F, TALWAR K. Mechanism design via differential privacy[C]//The 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). [S.l.]: IEEE, 2007: 94-103.

[19] 鲜征征, 李启良, 李改, 等. 差分隐私在协同过滤算法中的应用研究[J]. 计算机科学, 2017(5): 81-88.
XIAN Zheng-zheng, LI Qi-liang, LI Gai, et al. Research on application of differential privacy in collaborative filtering algorithms[J]. Computer Science, 2017(5): 81-88.

编　辑　叶　芳