文章编号:1674-2974(2022)08-0196-12 DOI:10.16339/j.cnki.hdxbzkb.2022287

基于时间权重因子的隐私保护推荐算法

王永†,王利 ,冉珣 ,肖玲
(重庆邮电大学 电子商务与现代物流重点实验室,重庆 400065)

摘 要:用户兴趣是随时间变化的,若对推荐系统中所有时间段的数据均采用同等程度的隐私保护,容易引入不必要的噪声,降低数据效用.为此,提出一种基于时间权重因子的差分隐私保护推荐算法.首先,设计时间权重因子,用于衡量数据重要性.然后,根据时间权重因子划分隐私预算,对不同时间段的数据施加不同强度的隐私保护.在此基础上,构建基于差分隐私的概率矩阵分解模型,用于完成个性化推荐.实验结果表明,该算法在满足隐私保护的条件下,能够更有效地保留数据效用,提高推荐结果的准确性.

关键词:推荐系统;矩阵分解;隐私保护;差分隐私;时间权重因子
中图分类号:TP399 **文献标志码**:A

Privacy Protection Recommendation Algorithm Based on Time Weight Factor

WANG Yong†, WANG Li, RAN Xun, XIAO Ling
(Key Laboratory of Electronic Commerce and Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: User interests change over time. If the same level of privacy protection is used for data of all periods in the recommender systems, it is easy to introduce unnecessary noise and reduce data utility. Therefore, a differential privacy protection recommendation algorithm based on the time weight factor is proposed. The algorithm first designs a time weight factor to measure the importance of data and then allocates the different privacy budgets to the data according to the time weight factor. That is, different intensity of privacy protection is performed on the data in different periods. Moreover, a probability matrix factorization model based on differential privacy is constructed for a personalized recommendation. Experimental results show that the proposed algorithm can preserve data utility more effectively and improve the accuracy of recommendation results under the condition of privacy protection.

Key words: recommender systems; matrix factorization; privacy protection; differential privacy; time weight factor

---

* **收稿日期**:2021-06-11
基金项目:教育部人文社科规划基金项目(20YJAZH102),MOE Layout Foundation of Humanities and Social Sciences(20YJAZH102);国家自然科学基金资助项目(71901045),National Natural Science Foundation of China(71901045);成渝双城经济圈科技创新项目(KJCX2020027),Science and Technology Innovation Project of The Chengdu-Chongqing Twin Cities Economic Zone (KJCX2020027);重庆市自然科学基金面上项目(CSTC2021JCYJ-MSXMX0557),Natural Science Foundation of Chongqing(CSTC2021JCYJ-MSXMX0557)
作者简介:王永(1977—),男,四川自贡人,重庆邮电大学教授,博士生导师
† 通信联系人,E-mail: wangyong1@cqupt.edu.cn

随着网络中数据的爆炸式增长,用户有效获取有用信息的难度日益增加.推荐算法结合用户的历史数据准确挖掘用户的真实意图,提供精准的个性化推荐服务[1],能帮助用户更快获得有用的信息.然而,个性化推荐需要利用大量个人信息,存在隐私泄露风险[2-3].因此,设计考虑隐私保护的推荐算法是非常必要的.

近年来,将差分隐私技术应用到推荐领域取得了良好的进展,其中一类典型的处理方式是将隐私保护技术与邻居型协同过滤算法相结合.Zhu等人[4]运用指数机制对邻居选择过程进行扰动,减小攻击者推测用户和项目相似性的概率,防止攻击者通过邻居信息推测用户评分数据.Yang等[5]针对用户上下文兴趣建模时的隐私保护问题,在计算用户平均分和相似度时进行差分隐私保护,并利用聚类算法解决数据稀疏问题.Mcsherry等人[6]将推荐算法分为学习阶段和预测阶段,在学习阶段引入噪声实现对项目相似度矩阵的保护.Yang等人[7]根据用户隐私需求特点,将用户隐私需求分为3种不同层次,在计算相似度时,对不同层次隐私需求的用户采用不同强度的拉普拉斯噪声进行扰动,实现个性化差分隐私保护.此类基于邻居选择的隐私保护推荐算法具有良好的可解释性以及推荐性能.然而,该类算法存在高维数据稀疏性问题及可拓展性问题.在历史数据较少时,推荐质量不高.

基于矩阵因式分解的推荐算法是另一类主流推荐算法,具有准确度高、拓展性好、灵活度高等特点.通过将高维稀疏矩阵分解为两个低维特征矩阵,能有效解决数据稀疏性问题,具有良好的应用前景.针对该类算法,Zhang等人[8]根据用户自身的特点,设计了一种特殊的评分数据采样机制,实现个性化差分隐私保护.鲜征征等人[9]致力于将SVD++模型与差分隐私机制相结合,分别从梯度扰动、目标函数扰动、输出结果扰动提出基于差分隐私机制和SVD++结合的模型.郑剑等[10]提出一种融合标签相似度的差分隐私矩阵分解推荐模型,可以同时保护标签数据和用户评分.为减少噪声的引入,Zhang等人[11]设计一种新的目标函数扰动方式,并通过联合学习得到差分隐私分解矩阵.

然而,现有的隐私保护推荐算法大多是基于静态数据进行设计的.现实中,用户兴趣是一个动态变化的过程[12-13].用户兴趣变化导致评分数据的重要程度变化,进而使隐私需求相应发生变化.上述算法对推荐系统进行隐私保护时忽略隐私需求的变化,容易引入不必要的噪声,降低数据效用,进而导致推荐质量降低.为解决上述问题,本文从用户兴趣漂移的行为数据出发,将时间因素作为度量隐私保护程度的关键点,提出一种基于时间权重因子的隐私保护推荐算法.设计时间权重因子刻画数据对用户的重要性,对不同时间段的数据根据其重要性进行不同强度的隐私保护.所提出的算法旨在充分保障用户隐私安全的条件下,有效提升数据的有效性,进而提升推荐质量.

1　预备知识

1.1　概率矩阵分解

概率矩阵分解(Probability Matrix Factorization,PMF)算法作为推荐系统的主流算法之一,在稀疏度高的评分矩阵中表现出良好的推荐精确度[14].相关评分矩阵$\boldsymbol{R}$的条件分布如下:

$$p(\boldsymbol{R}|\boldsymbol{U},\boldsymbol{V},\delta_R^2)=\prod_{(i,j)\in\boldsymbol{R}}\left[N(\boldsymbol{R}_{i,j}|\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j,\delta_R^2)\right]^{I_{ij}} \tag{1}$$

式中:$\boldsymbol{u}_i$、$\boldsymbol{v}_j$分别为K维的用户因子向量和项目因子向量;I_{ij}为指示函数,当用户i评论过项目j时,$I_{ij}=1$,否则$I_{ij}=0$;$N(\boldsymbol{R}_{i,j}|\mu,\delta_R^2)$是服从高斯分布的概率密度函数,其均值为$\mu$,方差为$\delta_R^2$.

当$\boldsymbol{U}$、$\boldsymbol{V}$均为$\mu=0$的高斯球面先验分布时,$\boldsymbol{U}$、$\boldsymbol{V}$的概率密度函数分布分别为:

$$\begin{aligned}p(\boldsymbol{U}|\delta_U^2)&=\prod_{i\in[1,M]}N(\boldsymbol{u}_i|0,\delta_U^2\boldsymbol{I})\\p(\boldsymbol{V}|\delta_V^2)&=\prod_{j\in[1,N]}N(\boldsymbol{v}_j|0,\delta_V^2\boldsymbol{I})\end{aligned} \tag{2}$$

对式(2)的后验分布取对数进行分析,计算公式如下:

$$\begin{aligned}\ln p(\boldsymbol{U},\boldsymbol{V}|\boldsymbol{R},\delta_R^2,\delta_U^2,\delta_V^2)=&\\-\frac{1}{2\delta_R^2}\sum_{(i,j)\in\boldsymbol{R}}I_{ij}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2-&\\\frac{1}{2\delta_U^2}\sum_{i\in[1,N]}\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_i-\frac{1}{2\delta_V^2}\sum_{j\in[1,M]}\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{v}_j-&\\\frac{1}{2}\left[\left(\sum_{(i,j)\in\boldsymbol{R}}I_{ij}\right)\ln\delta_R^2+NK\ln\delta_U^2+MK\ln\delta_V^2\right]+C&\end{aligned} \tag{3}$$

式中:C为常量值.求解以式(3)为目标函数的最大化问题,就能训练出用户因子矩阵$\boldsymbol{U}$和项目因子矩阵$\boldsymbol{V}$,然后根据$\boldsymbol{U}$和$\boldsymbol{V}$进行预测评分,并根据预测结果为用户提供推荐服务.以上问题可以转换为求解

如下最小化问题:

$$\min_{U,V} E(\boldsymbol{U},\boldsymbol{V}) = \frac{1}{2}\sum_{(i,j)\in \boldsymbol{R}} I_{ij}(r_{ij} - \boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2 + \frac{\lambda_u}{2}\sum_{i\in[1,N]}\left\|\boldsymbol{u}_i\right\|_2^2 + \frac{\lambda_v}{2}\sum_{j\in[1,M]}\left\|\boldsymbol{v}_j\right\|_2^2 \tag{4}$$

式中:$\lambda_u > 0, \lambda_v > 0$为正则化参数;$\|\bullet\|_2$为欧几里得范数;$r_{ij}$表示用户$i$对项目$j$的评分;每个用户因子向量$\boldsymbol{u}_i \in \boldsymbol{U}$满足$\left\|\boldsymbol{u}_i\right\|_2 \leqslant 1$.

1.2 差分隐私

差分隐私是当前主流的隐私保护技术,本文所涉及的重要相关概念如下:

定义 1 ε-差分隐私[15]:D和D'为相差一条记录的邻居数据集.给定随机算法A,当A在数据集D和D'上的任意输出结果$O[O \in \mathrm{Range}(A)]$满足式(5),则称算法$A$满足$\varepsilon$-差分隐私.

$$\Pr[A(D) \in S] \leqslant \mathrm{e}^{\varepsilon} \times \Pr[A(D') \in S] \tag{5}$$

式中:$\Pr[\bullet]$表示事件发生的概率;ε为隐私预算.

定义 2 $\boldsymbol{\rho}$-个性化差分隐私[16]:设随机算法$A: D \to \mathrm{Range}(A)$,并且用户-项目评分的隐私预算矩阵$\boldsymbol{\rho} = [\varepsilon_{ij}]_{N\times M}$.如果算法满足式(6),称随机算法$A$满足$\boldsymbol{\rho}$-个性化差分隐私.

$$\Pr[A(D) \in S] \leqslant \mathrm{e}^{\varepsilon_{ij}} \times \Pr[A(D') \in S] \tag{6}$$

式中:ε_{ij}表示r_{ij}的个性化隐私预算.

2 应用场景

本研究的应用场景为集中式推荐系统,采用PMF算法为推荐模型.系统被认为是可靠和可信赖的,这类系统通过收集并利用用户评分数据进行模型训练,为用户提供个性化推荐服务.然而,用户评分不仅直接反映其兴趣偏好,还隐含用户的性别、年龄、收入水平等信息,用户的评分信息如果被他人获取,则个人隐私泄露风险增加.因此,推荐系统应当着力于保障系统中用户评分信息的安全.

文献[11]表明,一个具有隐私保护的推荐系统应该确保攻击者不能学习用户因子矩阵$\boldsymbol{U}$和项目因子矩阵$\boldsymbol{V}$,否则,系统任何评分数据都可以由两个因子矩阵内积$\boldsymbol{U}^{\mathrm{T}} \cdot \boldsymbol{V}$推导出来.为了抵御这种攻击,推荐系统需要保密储存$\boldsymbol{U}$,只发布$\boldsymbol{V}$.此外,发布$\boldsymbol{V}$有助于解决项目评分数据不足问题.例如,不同的推荐系统,拥有相似的项目集,但用户群不同.通过与其他推荐系统共享$\boldsymbol{V}$,推荐者可以使用本地用户信息进一步训练$\boldsymbol{V}$.这样,推荐系统就可以利用多个来自其他系统的数据进行模型训练,有效实现信息共享,缓解信息不足的问题,改进推荐系统的性能.

然而,项目因子矩阵$\boldsymbol{V}$包含用户信息,直接发布真实的$\boldsymbol{V}$依然会带来隐私问题.假设攻击者拥有除用户评分r_{ab}之外的其他所有用户的评分数据和真实的$\boldsymbol{V}$.攻击者想要获得r_{ab}.可以采用以下两种典型的攻击方式:

1)相似性攻击[4]:项目因子矩阵揭示了项目评分之间的相似性,可以帮助攻击者预测用户信息.攻击者通过做一些关于未知评分r_{ab}的假设及观察$\boldsymbol{v}_{\mathrm{b}}$和$\boldsymbol{v}_i, i \in \{x \mid x \in I \backslash b\}$之间相似性的变化,推断出实际的$r_{\mathrm{ab}}$.

2)重构攻击[17]:根据真实项目因子矩阵$\boldsymbol{V}$,攻击者只需要求解如下问题就能够得到用户a的信息$\boldsymbol{u}_{\mathrm{a}}$:

$$\min_{\boldsymbol{u}_{\mathrm{a}}} \frac{1}{2}\left(\sum_{i\in \tilde{I}_{\mathrm{a}}}\left(r_{ai} - \boldsymbol{u}_{\mathrm{a}}^{\mathrm{T}}\boldsymbol{v}_i\right)^2 + \lambda\left\|\boldsymbol{u}_{\mathrm{a}}\right\|_2^2\right) \tag{7}$$

其中,$\tilde{I}_{\mathrm{a}} = \{x \mid x \in I_{\mathrm{a}} \backslash b\}$,同时拥有$\boldsymbol{u}_{\mathrm{a}}$和$\boldsymbol{v}_{\mathrm{b}}$,攻击者就能利用内积$\boldsymbol{u}_{\mathrm{a}}^{\mathrm{T}} \cdot \boldsymbol{v}_{\mathrm{b}}$预测$r_{\mathrm{ab}}$.

为了抵御这两种攻击方式,对推荐系统进行如下处理:首先,推荐系统训练不加扰动的推荐模型,得到$\boldsymbol{U}$并将其保密储存.随后,将用户因子矩阵$\boldsymbol{U}$作为常数,训练满足差分隐私的推荐模型得到并发布扰动后的项目因子矩阵$\bar{\boldsymbol{V}}$.扰动后的$\bar{\boldsymbol{V}}$可以防止攻击者通过获取任意两个项目因子之间的精确距离,能够对相似性进行有效保护.$\bar{\boldsymbol{V}}$也可以防止攻击者获得准确的项目因子矩阵以抵御重构攻击.此外,其他推荐系统仍然可以利用$\bar{\boldsymbol{V}}$训练自己的模型,提高推荐质量.

综上所述,在本文方案中,为保护用户的评分信息,用户因子矩阵$\boldsymbol{U}$和项目因子矩阵$\boldsymbol{V}$均需要进行保护.其中,$\boldsymbol{U}$通过在可信系统内部以保密储存的方式进行保护,$\boldsymbol{V}$通过引入差分隐私以添加噪声的方式进行保护.

3 基于时间权重因子的隐私保护推荐算法

当前大多数隐私保护推荐算法对评分数据进行隐私保护时没有考虑时间的影响,将所有时间段的评分数据视为同等重要程度.然而,时间因素对推荐系统有着重要的影响,且具有很好的研究价值.费洪晓等[18]运用时间窗口调整用户兴趣漂移带来的影响;Pan等[19]提出时间距离越近的信息在推荐时更加

受重视;Jiang等[20]将时间权重信息应用到用户评分数上,削弱用户过去兴趣,突出现在的兴趣;兰燕等[21]认为用户具有兴趣漂移的特性,即用户兴趣是变化的,且信息的影响力随时间阶段性衰减;Chen等[22]认为发生在不同时间的信息对表示用户当前兴趣的贡献值是不一样的,并引入4种遗忘曲线以更好地把握用户近期的兴趣.上述研究表明,用户兴趣偏好会随着时间变化,发生时间不同的评分数据对用户重要程度存在差异,近期数据更能反映用户当下的兴趣偏好,更为重要.对所有时间段的数据采用相同程度的隐私保护,容易引入不必要的噪声,降低推荐算法的性能.因此,有必要考虑时间的影响,对不同时间段的评分数据施加不同强度的隐私保护.从而达到在保障用户隐私安全的前提下,不降低数据的有效性,提升推荐的准确度.

为了论述方便,相关符号说明如表1所示.

表1　符号说明

Tab.1　Description of symbol

符号	含义
ε	统一隐私预算
N	评分矩阵的用户数
M	评分矩阵的项目数
$\boldsymbol{R}_{N\times M}$	N位用户对M个项目的评分矩阵
$\bar{\varepsilon}$	评分数据的平均隐私预算
t_{ij}	用户i对项目j进行评分的时间
t_{now}	计算推荐结果的时间
T_0	评分重要性的半衰期
T_1	评分重要性的保持期
K	隐因子维度
ω	算法迭代次数
λ_u、λ_v	正则化参数
$\boldsymbol{U}_{K\times N}$	用户因子矩阵
$\boldsymbol{V}_{K\times M}$	项目因子矩阵
$\bar{\boldsymbol{V}}_{K\times M}$	扰动后的项目因子矩阵
$\boldsymbol{u}_i$	用户i的因子向量
$\boldsymbol{v}_j$	项目j的因子向量
AVG$(F(t))$	评分数据时间权重的平均值
$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M}$	算法1输出的隐私预算矩阵
$\boldsymbol{D}_s$	评分数据抽样算法2输出的评分矩阵

本文设计了基于时间权重因子的隐私保护推荐方案,总体步骤如下:

步骤1　根据3.1节的算法1计算时间权重因子和评分的隐私预算;

步骤2　利用步骤1得到的隐私预算,根据3.2节的算法2对评分数据进行抽样,得到抽样数据集$\boldsymbol{D}_s$;

步骤3　利用抽样数据集$\boldsymbol{D}_s$,根据3.3节的算法3,生成具有差分隐私保护作用的PMF模型.

本文方案对应的整体框架如图1所示.

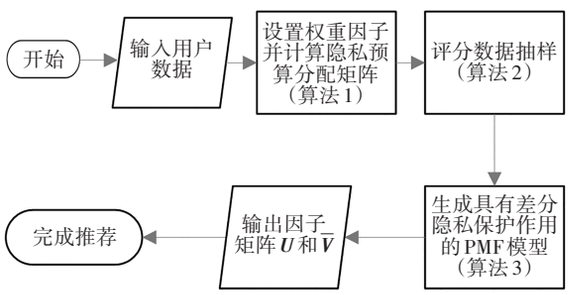


图1　方案框架

Fig.1　The framework of the proposed scheme

3.1　考虑时间权重因子的隐私预算

时间对兴趣点具有深远和广泛的影响.首先用户兴趣会因自身成长、阶段性角色的转变等而发生变化.其次,项目本身也具有时效性,如项目的流行性、生命周期等.兴趣点的改变导致评分数据对推荐系统的重要程度存在差异.因此,引入时间权重因子用于调节信息价值在时间变化中的衰减情况.时间权重因子的设计主要考虑两个因素:评分重要性的半衰期T_0[23],即评分从发布到其重要性减半所需要的时间;评分重要性的保持期T_1[21],即评分重要性基本维持不变的时长.根据以上两个概念,构建时间权重因子$F(t_{ij})$为:

$$F(t_{ij})=\exp\left[\frac{\ln(0.5)}{T_0}\cdot T_1\cdot\text{floor}\left(\frac{t_{\text{now}}-t_{ij}}{T_1}\right)\right] \quad (8)$$

式中:t_{now}为计算推荐结果的时间;t_{ij}为评分r_{ij}发生的时间;floor()为阶梯函数.权重因子$F(t_{ij})$随$(t_{\text{now}}-t_{ij})$的增大而减小,表示评分发生的时间越长,用户兴趣越可能发生改变,重要程度越小.

时间权重因子表示评分的重要程度.时间权重因子越大,评分重要程度越高,应采用较高的隐私保护强度.当时间权重因子低于所设置的阈值时,表明评分信息的重要程度下降,应降低其隐私保护强度以减少噪声的引入.通过该方式对评分数据进行隐私保护更加符合实际情况,能够有效提升推荐的准确性.针对每个评分,采用如下隐私预算分配公式:

$$\varepsilon_{ij}=\begin{cases}\varepsilon\times\dfrac{1}{F(t_{ij})} & \text{if } F(t_{ij})\leqslant \mathrm{AVG}(F(t))\\ \varepsilon & \text{其他}\end{cases} \tag{9}$$

式中:ε为统一隐私预算;ε_{ij}表示评分r_{ij}的隐私预算;$\mathrm{AVG}(F(t))$表示时间权重因子阈值.限制隐私预算范围为:

$$\begin{cases}\varepsilon_{ij} & \text{if } \varepsilon_{ij}\leqslant 10\\ 10 & \text{其他}\end{cases}$$

隐私预算描述了隐私保护的强弱程度,隐私预算越小,相应的隐私保护强度越高.

基于上述分析,设计考虑时间权重因子的隐私预算分配算法如算法1所示.

算法1:考虑时间权重因子的隐私预算分配算法

输入:$\varepsilon,t_{ij},t_{\mathrm{now}},T_0,T_1,\boldsymbol{R}_{N\times M}$

输出:$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M},\bar{\varepsilon}$

步骤1:对$\boldsymbol{R}$中的每个评分,根据t_{ij},计算其时间权重因子$F(t_{ij})$如下:

$$F(t_{ij})=\exp\left[\frac{\ln(0.5)}{T_0}\cdot T_1\cdot\mathrm{floor}\left(\frac{t_{\mathrm{now}}-t_{ij}}{T_1}\right)\right]$$

其中$i=1,2,\cdots,N;\ j=1,2,\cdots,M$

步骤2:计算所有$F(t_{ij})$的平均值$\mathrm{AVG}(F(t))$如下:

$$\mathrm{AVG}(F(t))=\frac{1}{|\boldsymbol{R}|}\sum_{(i,j)\in\boldsymbol{R}}F(t_{ij})$$

步骤3:对$\boldsymbol{R}$中的每个评分项,根据$F(t_{ij})$,计算其隐私预算ε_{ij}如下:

If $F(t_{ij})\leqslant \mathrm{AVG}(F(t))$:

$$\varepsilon_{ij}=\varepsilon\times\frac{1}{F(t_{ij})}$$

Else: $\varepsilon_{ij}=\varepsilon$

步骤4:计算所有评分数据的平均隐私预算$\bar{\varepsilon}$如下:$\bar{\varepsilon}=\dfrac{1}{|\boldsymbol{R}|}\sum_{(i,j)\in\boldsymbol{R}}\varepsilon_{ij}$

步骤5:输出$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M},\bar{\varepsilon}$

3.2 评分数据抽样

数据中每个评分的隐私预算存在差异,为了根据评分的隐私预算进行不同强度的隐私保护,采用随机抽样算法对评分数据进行抽样.随机抽样算法定义如下:

给定数据集$\boldsymbol{R}_{N\times M}$、算法1的输出$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M}$和$\bar{\varepsilon}$.以式(10)所定义的概率$\pi(r_{ij},\bar{\varepsilon})$对$\boldsymbol{R}$中的评分数据进行随机抽样.

$$\pi(r_{ij},\bar{\varepsilon})=\begin{cases}\dfrac{\mathrm{e}^{\varepsilon_{ij}}-1}{\mathrm{e}^{\bar{\varepsilon}}-1} & \text{if } \varepsilon_{ij}<\bar{\varepsilon}\\ 1 & \text{其他}\end{cases} \tag{10}$$

其中$r_{ij}\in\boldsymbol{R}$.未被抽中的评分,将其评分值设为0.

结合隐私预算的随机抽样算法如算法2所示.

算法2:随机抽样算法

输入:$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M},\boldsymbol{R}_{N\times M},\bar{\varepsilon}$

输出:$\boldsymbol{D}_{\mathrm{s}}\in\boldsymbol{R}_{N\times M}$

步骤1:设置$\boldsymbol{D}_{\mathrm{s}}=\boldsymbol{R}$

步骤2:对于$\boldsymbol{D}_{\mathrm{s}}$中的每个评分,根据$\varepsilon_{ij}$,计算抽样概率$\pi(r_{ij},\bar{\varepsilon})$如下:

If $\varepsilon_{ij}<\bar{\varepsilon}$:

$$\pi(r_{ij},\bar{\varepsilon})=\frac{\mathrm{e}^{\varepsilon_{ij}}-1}{\mathrm{e}^{\bar{\varepsilon}}-1}$$

Else: $\pi(r_{ij},\bar{\varepsilon})=1$

步骤3:根据每个评分项的抽样概率$\pi(r_{ij},\bar{\varepsilon})$,对$\boldsymbol{D}_{\mathrm{s}}$中的评分项抽样如下:

If $\pi(r_{ij},\bar{\varepsilon})\neq 1$:

If $\pi(r_{ij},\bar{\varepsilon})$未被选择:

$\boldsymbol{D}_{\mathrm{s}}[i,j]\leftarrow 0$

End if

End if

步骤4:输出$\boldsymbol{D}_{\mathrm{s}}$

在随机抽样算法中,评分数据被分为两个两部分:①算法未抽中的评分数据.当评分数据的隐私预算低于所设定阈值时,有一定概率不被抽中.未被抽中的数据被设置为0,直接不参与推荐流程,能够最大限度保护这些数据.②算法抽中的评分数据$\boldsymbol{D}_{\mathrm{s}}$.被抽中的数据$\boldsymbol{D}_{\mathrm{s}}$将作为输入项,用于3.3节的模型训练中,实现具有隐私保护的个性化推荐.

3.3 基于隐私保护的概率矩阵分解模型

为了实现PMF模型与隐私保护的结合,采用对目标函数添加扰动的方式.扰动后的目标函数如下:

$$\begin{aligned}\min_{U,V}E(\boldsymbol{U},\bar{\boldsymbol{V}})=&\frac{1}{2}\sum_{(i,j)\in\boldsymbol{D}_{\mathrm{s}}}I_{ij}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\bar{\boldsymbol{v}}_j)^2+\\ &\frac{\lambda_u}{2}\sum_{i\in[1,N]}\left\|\boldsymbol{u}_i\right\|_2^2+\frac{\lambda_v}{2}\sum_{j\in[1,M]}\left\|\bar{\boldsymbol{v}}_j\right\|_2^2+\sum_{j\in[1,M]}\boldsymbol{\eta}_j^{\mathrm{T}}\bar{\boldsymbol{v}}_j\end{aligned} \tag{11}$$

式中:$\boldsymbol{\eta}_j$是随机噪声向量且其概率分布满足$P(\boldsymbol{\eta}_j)\propto\mathrm{e}^{\frac{-\bar{\varepsilon}\cdot\left\|\boldsymbol{\eta}_j\right\|_2}{\Delta}}$,$\Delta=r_{\max}-r_{\min}$,为函数的敏感度.

在模型训练中,首先,用交替最小二乘法求解式(4)所示的不加扰动的PMF目标函数.

1)固定$\boldsymbol{U}$,对式(4)的$\boldsymbol{v}_j$求偏导$\partial E(\boldsymbol{U},\boldsymbol{V})/\partial\boldsymbol{v}_j=0$,得到求解$\boldsymbol{v}_j$的公式:

$$\boldsymbol{v}_j=(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\lambda_u\boldsymbol{I})^{-1}\boldsymbol{U}^{\mathrm{T}}r_j$$

2)固定$\boldsymbol{V}$,对式(4)的$\boldsymbol{u}_i$求偏导$\partial E(\boldsymbol{U},\boldsymbol{V})/\partial\boldsymbol{u}_i=0$,得到求解$\boldsymbol{u}_i$的公式:

$$\boldsymbol{u}_i=(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}+\lambda_v\boldsymbol{I})^{-1}\boldsymbol{V}^{\mathrm{T}}r_i$$

迭代上述过程,直到收敛,得到用户因子矩阵$\boldsymbol{U}$.随后,将$\boldsymbol{U}$作为常数,代入式(11),求解扰动后的PMF目标函数,即对式(11)的$\bar{\boldsymbol{v}}_j$求偏导

$\partial E(\boldsymbol{U},\boldsymbol{V})/\partial\bar{\boldsymbol{v}}_j=0$,得到求解$\bar{\boldsymbol{v}}_j$的公式:

$$\bar{\boldsymbol{v}}_j=(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\lambda_u\boldsymbol{I})^{-1}\cdot(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{r}_j-\boldsymbol{\eta}_j)$$

迭代上述过程至收敛,得到扰动后的项目因子矩阵$\bar{\boldsymbol{V}}$.

上述求解过程如算法3所示.

算法3:基于差分隐私的概率矩阵分解推荐算法

输入:$\boldsymbol{D}_{\mathrm{s}},K,\omega,\lambda_u,\lambda_v,\bar{\varepsilon}$

输出:$\boldsymbol{U},\bar{\boldsymbol{V}}$

步骤1:随机高斯初始化因子矩阵$\boldsymbol{U}$、$\boldsymbol{V}$和$\bar{\boldsymbol{V}}$

步骤2:For α from 1 to ω do

　　固定$\boldsymbol{U}$

　　For each $\boldsymbol{v}_j$ in D_{s} do

　　$\boldsymbol{v}_j=(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\lambda_u\boldsymbol{I})^{-1}\boldsymbol{U}^{\mathrm{T}}r_j$

　　End for

　　固定$\boldsymbol{V}$

　　For each $\boldsymbol{u}_i$ in D_{s} do

　　$\boldsymbol{u}_i=(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}+\lambda_v\boldsymbol{I})^{-1}\boldsymbol{V}^{\mathrm{T}}r_i$

　　End for

End for

//将步骤2得到的$\boldsymbol{U}$作为常数代入后续步骤

步骤3:For β from 1 to ω do

　　For each $\bar{\boldsymbol{v}}_j$ in D_{s} do

　　$\bar{\boldsymbol{v}}_j=(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{\lambda}_u\boldsymbol{I})^{-1}\cdot(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{r}_j-\boldsymbol{\eta}_j)$

　　//噪声向量$\boldsymbol{\eta}_j$满足$P(\boldsymbol{\eta}_j)\propto\exp(\frac{-\bar{\varepsilon}\cdot\left\|\boldsymbol{\eta}_j\right\|_2}{\Delta})$

　　End for

End for

步骤4:$\boldsymbol{U}$由系统保密保存,输出$\bar{\boldsymbol{V}}$

为了防止攻击者通过发布的信息来预测用户偏好,将用户因子矩阵$\boldsymbol{U}$进行保密储存,只发布扰动后的项目因子矩阵$\bar{\boldsymbol{V}}$. 推荐系统利用自身保密存储的用户因子矩阵$\boldsymbol{U}$和发布的项目因子矩阵$\bar{\boldsymbol{V}}$,可以预测用户对项目的评分,并据此提供个性化推荐服务.

4　算法分析

4.1　安全性分析

引理1[8]　对概率矩阵分解模型的目标函数添加扰动的方式如下:

$$\min_{U,V}E(\boldsymbol{U},\boldsymbol{V})=\frac{1}{2}\sum_{(i,j)\in\boldsymbol{R}}I_{ij}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2+\frac{\lambda_u}{2}\sum_{i\in[1,N]}\left\|\boldsymbol{u}_i\right\|_2^2+\frac{\lambda_v}{2}\sum_{j\in[1,M]}\left\|\boldsymbol{v}_j\right\|_2^2+\sum_{j\in[1,M]}\boldsymbol{\eta}_j^{\mathrm{T}}\boldsymbol{v}_j \tag{12}$$

若噪声向量$\boldsymbol{\eta}_j$是满足$P(\boldsymbol{\eta}_j)\propto\mathrm{e}^{\frac{-\varepsilon\cdot\left\|\boldsymbol{\eta}_j\right\|_2}{\Delta}}$分布的随机变量,求解式(12)得到的项目因子矩阵$\boldsymbol{V}$满足$\varepsilon$-差分隐私.

引理2[8]　令$\boldsymbol{R}$表示评分数据集,$\boldsymbol{\rho}$表示用户隐私预算矩阵. 抽样算法$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},t)$以式(13)所示的概率$\pi(r_{ij},t)$对原始数据集$\boldsymbol{R}$进行随机抽样. 将$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},t)$抽样后的数据作为输入集,训练任意满足$t$-差分隐私的推荐模型,则所得的模型满足$\boldsymbol{\rho}$-个性化差分隐私.

$$\pi(r_{ij},t)=\begin{cases}\dfrac{\mathrm{e}^{\varepsilon_{ij}}-1}{\mathrm{e}^{t}-1} & \text{if}\quad t>\varepsilon_{ij}\\ 1 & \text{其他}\end{cases} \tag{13}$$

式中:t是一个可调整的值.

定理1　本文提出的基于时间权重因子的隐私保护推荐方案满足$\boldsymbol{\rho}$-个性化差分隐私.

证明　本文方案包括算法1、算法2、算法3,分别对这3种算法进行分析,证明本文方案的安全性.

1)算法1:输出为隐私预算矩阵$\boldsymbol{\rho}=[\varepsilon_{ij}]_{N\times M}$和平均隐私预算$\bar{\varepsilon}=\dfrac{1}{|\boldsymbol{R}|}\sum_{(i,j)\in\boldsymbol{R}}\varepsilon_{ij}$,是算法2和算法3的前提.

2)算法2:在引理2中,取$t=\bar{\varepsilon}$时,算法2采用的抽样概率$\pi(r_{ij},\bar{\varepsilon})$与引理2中的$\pi(r_{ij},t)$一致. 算法2可以表示为$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})$,输出为$\boldsymbol{D}_{\mathrm{s}}$.

3)算法3:算法3对应的目标函数公式(11)在形式上与引理1的式(12)一致,且其随机噪声向量$\boldsymbol{\eta}_j$的概率分布满足$P(\boldsymbol{\eta}_j)\propto\mathrm{e}^{\frac{-\bar{\varepsilon}\cdot\left\|\boldsymbol{\eta}_j\right\|_2}{\Delta}}$. 由引理1可知,算法3发布的$\bar{\boldsymbol{V}}$满足$\bar{\varepsilon}$-差分隐私.

4)将$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})$的输出$\boldsymbol{D}_{\mathrm{s}}$作为输入用于训练算法3满足$\bar{\varepsilon}$-差分隐私的推荐模型,令该步骤为$S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})=\mathrm{DP}^{\bar{\varepsilon}}(\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}))$. 对任意的输出$O\in\mathrm{Range}(\mathrm{S}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}))$,当满足如下公式时定理1成立:

$$\Pr\left[S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})\in O\right]\leqslant\mathrm{e}^{\varepsilon_{pq}}\cdot\Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon})\in O\right] \tag{14}$$

式中:$\boldsymbol{R}=\left[r_{ij}\right]_{N\times M}$和$\boldsymbol{R}'=\left[r'_{ij}\right]_{N\times M}$为仅有一个评分数据不同的邻居数据集. 设它们的评分r_{pq}不同,即假设$\boldsymbol{R}$中评分r_{pq}被随机抽样算法抽中而$\boldsymbol{R}'$中$r'_{pq}=0$未被抽中.

$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})$的输出可以分为两种情况:一种是$r_{pq}$被抽中,另一种是$r_{pq}$未被抽中. $\boldsymbol{D}_{\mathrm{s}}=\left[r''_{ij}\right]_{N\times M}$表示算法$\mathrm{RS}(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon})$的输出,注意,$\boldsymbol{D}_{\mathrm{s}}\leqslant\boldsymbol{R}'$表示对于任意

的 $1 \leqslant i \leqslant N$ 和 $1 \leqslant j \leqslant M$ 有这些关系成立：$r''_{ij} = r'_{ij}$ 或者 $r''_{ij} = 0$. 用 $\boldsymbol{D}_{s+}$ 表示与 $\boldsymbol{D}_s$ 仅有一个评分记录不同的数据集($\boldsymbol{D}_{s+}$ 中 r_{pq} 是有效评分，而 $\boldsymbol{D}_s$ 中同样位置的评分为0). 因此，证明如下：

$$\begin{aligned}&\Pr\left[S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] = \\ &\sum_{Z \leqslant \boldsymbol{R}'}(\pi(r_{pq},\bar{\varepsilon}) \cdot \Pr\left[\mathrm{RS}(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) = \boldsymbol{D}_s\right] \cdot \Pr\left[\mathrm{DP}^{\bar{\varepsilon}}(\boldsymbol{D}_{s+}) \in O\right]) + \\ &\sum_{Z \leqslant \boldsymbol{R}'}((1 - \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[\mathrm{RS}(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) = \boldsymbol{D}_s\right] \cdot \Pr\left[\mathrm{DP}^{\bar{\varepsilon}}(\boldsymbol{D}_s) \in O\right]) = \\ &\sum_{Z \leqslant \boldsymbol{R}'}(\pi(r_{pq},\bar{\varepsilon}) \cdot \Pr\left[\mathrm{RS}(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) = \boldsymbol{D}_s\right] \cdot \Pr\left[\mathrm{DP}^{\varepsilon}(\boldsymbol{D}_{s+}) \in O\right] + \\ &(1 - \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right]\end{aligned} \tag{15}$$

由式(11)可知，$DP^{\bar{\varepsilon}}$ 满足 $\bar{\varepsilon} - DP$，所以式(15)可以继续化简为：

$$\begin{aligned}&\Pr\left[S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] \leqslant \\ &\sum_{Z \leqslant \boldsymbol{R}'}(\pi(r_{pq},\bar{\varepsilon}) \cdot \Pr\left[\mathrm{RS}(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) = \boldsymbol{D}_s\right] \cdot \mathrm{e}^{\bar{\varepsilon}} \cdot \Pr\left[\mathrm{DP}^{\bar{\varepsilon}}(\boldsymbol{D}_s) \in O\right]) + \\ &(1 - \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] = \\ &\mathrm{e}^{\bar{\varepsilon}} \cdot \pi(r_{pq},\bar{\varepsilon}) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] + \\ &(1 - \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] = \\ &(1 - \pi(r_{pq},\bar{\varepsilon}) + \mathrm{e}^{\bar{\varepsilon}} \cdot \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right]\end{aligned} \tag{16}$$

根据随机抽样算法式(10)，对 $\pi(r_{pq},\bar{\varepsilon})$ 分情况进行讨论：

1)当 $\bar{\varepsilon} \leqslant \varepsilon_{ij}$ 时，评分 r_{pq} 被以概率 $\pi(r_{pq},\bar{\varepsilon}) = 1$ 挑选，则式(16)可继续化简为：

$$\begin{aligned}&\Pr\left[S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] \leqslant \\ &(1 - \pi(r_{pq},\bar{\varepsilon}) + \mathrm{e}^{\bar{\varepsilon}} \cdot \pi(r_{pq},\bar{\varepsilon})) \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] = \\ &\mathrm{e}^{\bar{\varepsilon}} \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] \leqslant \mathrm{e}^{\varepsilon_{pq}} \cdot \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right]\end{aligned}$$

2)当 $\bar{\varepsilon} > \varepsilon_{ij}$ 时，评分 r_{pq} 被以概率 $\pi(r_{pq},\bar{\varepsilon}) = (\mathrm{e}^{\varepsilon_{pq}} - 1)/(\mathrm{e}^{\bar{\varepsilon}} - 1)$ 挑选，可以得到：

$$\begin{aligned}&1 - \pi(r_{pq},\bar{\varepsilon}) + \mathrm{e}^{\bar{\varepsilon}} \cdot \pi(r_{pq},\bar{\varepsilon}) = \\ &\quad 1 - \frac{\mathrm{e}^{\varepsilon_{pq}} - 1}{\mathrm{e}^{\bar{\varepsilon}} - 1} + \mathrm{e}^{\bar{\varepsilon}} \cdot \frac{\mathrm{e}^{\varepsilon_{pq}} - 1}{\mathrm{e}^{\bar{\varepsilon}} - 1} = \\ &\quad \frac{\mathrm{e}^{\bar{\varepsilon}} - 1 - \mathrm{e}^{\varepsilon_{pq}} + 1 + \mathrm{e}^{\bar{\varepsilon}} \cdot (\mathrm{e}^{\varepsilon_{pq}} - 1)}{\mathrm{e}^{\bar{\varepsilon}} - 1} = \\ &\quad \frac{\mathrm{e}^{\bar{\varepsilon}} \cdot \mathrm{e}^{\varepsilon_{pq}} - \mathrm{e}^{\varepsilon_{pq}}}{\mathrm{e}^{\bar{\varepsilon}} - 1} = \mathrm{e}^{\varepsilon_{pq}}\end{aligned}$$

则

$$\begin{aligned}&\Pr\left[S(\boldsymbol{R},\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] \leqslant \\ &(1 - \pi(r_{pq},\bar{\varepsilon}) + \mathrm{e}^{\bar{\varepsilon}}\pi(r_{pq},\bar{\varepsilon}))\Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right] \leqslant \\ &\mathrm{e}^{\varepsilon_{pq}} \Pr\left[S(\boldsymbol{R}',\boldsymbol{\rho},\bar{\varepsilon}) \in O\right]\end{aligned}$$

证毕

4.2 复杂度分析

在本文算法中，算法1是对原始评分矩阵进行遍历，时间复杂度为 $O(NM)$. 类似地，算法2时间复杂度为 $O(NM)$. 算法3时间开销与其梯度下降更新公式相关，其时间复杂度为 $O(\omega N)$ 或 $O(\omega M)$. 则本文算法的整体时间复杂度为 $O[N(M + \omega)]$ 或者 $O[M(N + \omega)]$. 同理，算法1和算法2的空间复杂度均为 $O(NM)$；算法3的空间复杂度为 $O(NK)$ 或者 $O(KM)$. 由于 $K << (M$ 或 $N)$，故算法的整体空间复杂度近似于 $O(NM)$. 综上所述，本文算法的时间和空间复杂度均与数据数量呈正线性关系，应用于大规模数据运算时复杂度不会显著增加.

5 实验结果及分析

实验采用推荐系统领域常用的Movielens-100k、Movielens-1M、Epinions、Movielens-10M、Amazon-Books 5个数据集对算法性能进行分析. Movielens-10M、Amazon-Books数据集用于测试算法在大规模数据集上的性能. 数据集包含的统计信息如表2所示.

表2 数据集的统计信息

Tab.2 Statistics for the dataset

数据集	用户数	项目数	评分数	密度/%
Movielens-100k	943	1 682	100 000	6.304
Movielens-1M	6 040	3 900	1 000 209	4.246
Movielens-10M	71 567	10 681	10 000 054	1.308
Epinions	22 164	296 277	922 267	0.014
Amazon-Books	8 026 324	2 330 193	22 507 155	0.000 12

实验的训练集与测试集比例为4∶1，评价指标为均方根误差(RMSE). 实验中默认参数设置为：隐因子维度 $K = 5$，迭代次数 $\omega = 50$，正则化参数 $\lambda_u = \lambda_v = 1$. 为保证结果的有效性，对每个算法进行5次实验，取均值作为实验结果. 所有实验均基于Python实现，使用PC机执行，操作系统为Windows 10 64b，CPU是Intel® Core™ i7-9700 CPU @ 3.00GHz，RMA是16-GB.

实验主要检验3个问题：①时间权重因子对算法准确性的影响；②本文算法预测的准确性；③算法的效率.

5.1 时间权重因子对算法准确性的影响

信息重要性衰减曲线如图2所示. 横轴表示距离评分的时间，纵轴表示信息重要程度随时间的衰减情况.

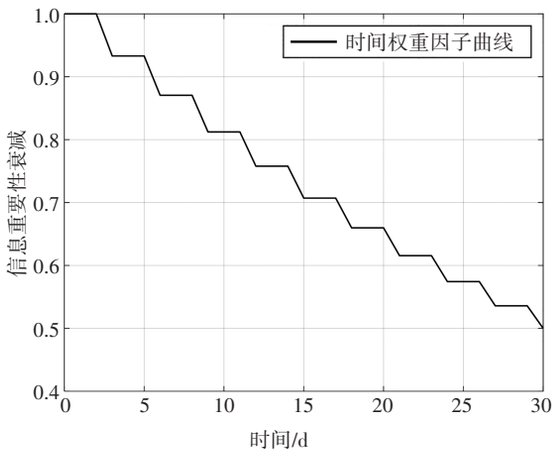
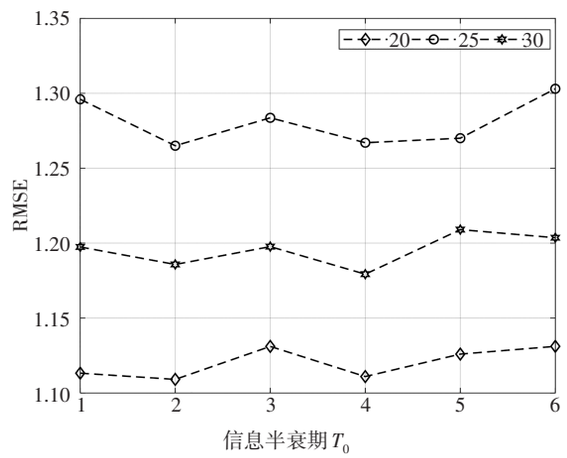


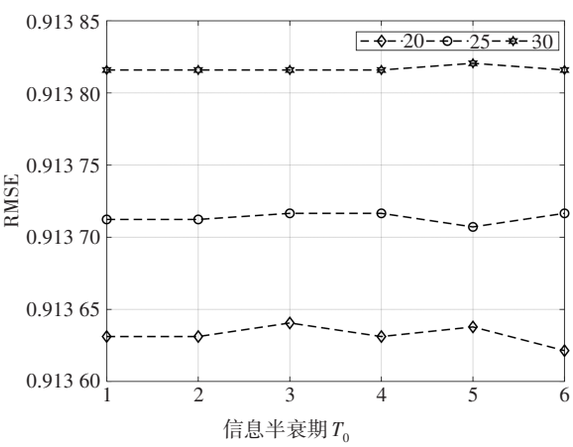
图 2　信息重要性衰减曲线

Fig.2　Information importance decay curve

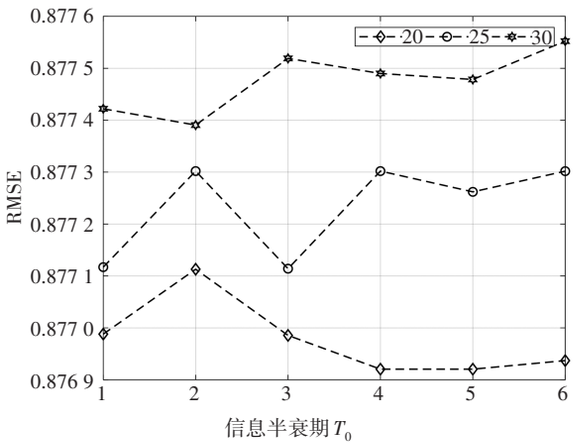
由式(8)可知,在时间权重因子曲线中,信息重要性的衰减程度与参数T_0与T_1相关. 本文算法在进行隐私保护时结合了时间权重因子. 为实现算法的最佳性能,需要首先确定最优的时间权重因子参数. 本节实验T_1分别取 20、25、30. 为方便对比,本节实验只呈现算法中引入时间权重因子的结果. 实验统一隐私预算ε = 0.1. 实验结果如图 3 所示.



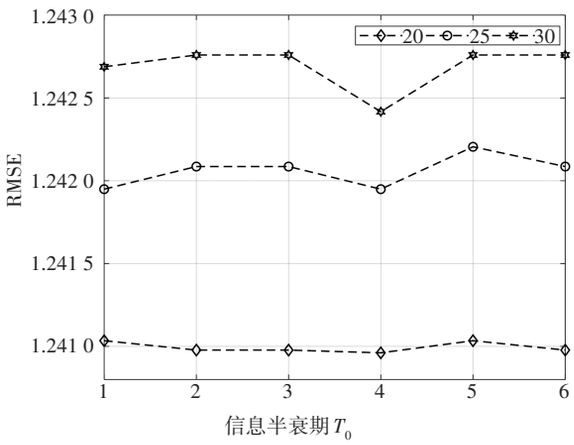
(a)Movielens-100K



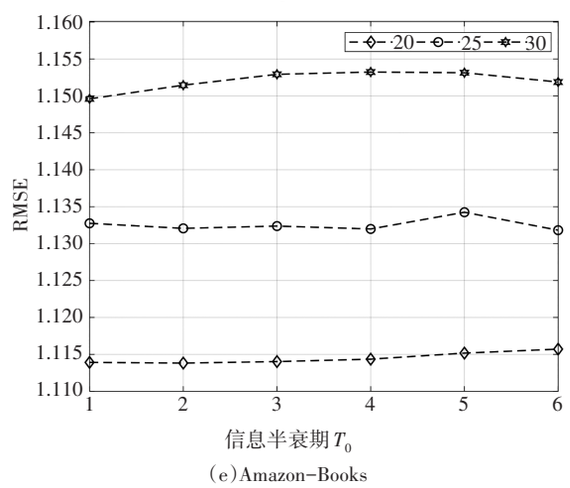
(b)Movielens-1M



(c)Movielens-10M



(d)Epinions



(e)Amazon-Books

图 3　时间参数的影响

Fig.3　Influence of time parameters

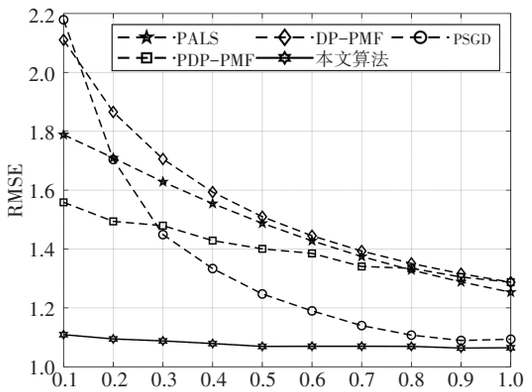
由图 3 可知,在 Movielens、Epinions 与 Amazon-Books 数据集上,算法的准确性由于时间权重因子参数不同存在差异. 在图 3(a)的 Movielens-100K 数据集中,当T_1 = 20 时,算法的 RMSE 整体更低,预测准确性更高,且当T_0为 2 时预测精确度最好. 此时,算法对T_1的改变比较敏感,对T_0的改变不敏

感. 图3 (b)中,RMSE的变化情况与图3 (a)类似,在0.913 62~0.913 82内波动,在T_1 = 20、T_0 = 6时推荐效果最优. 由图3 (c)(d)可知,T_1 = 20、T_0 = 4时算法性能最好. 图4 (e)中,RMSE在T_1值不同时差异较大,但在T_1值相同时波动较小,在T_1 = 20、T_0 = 2时取得最优结果. 上述结果差异主要是由于对于相同时间段发生的评分,其时间权重因子会随着参数变化而变化,进而隐私保护强度水平不同,最终改变算法的预测准确性.

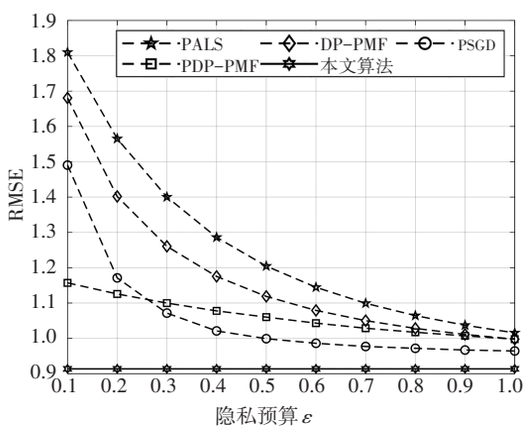
5.2 算法性能对比

为验证模型的有效性,将本文算法与其他4种基于矩阵分解的隐私保护算法进行比较. 涉及的对比算法有:①基于随机梯度扰乱的矩阵分解(Private Stochastic Gradient Perturbation, PSGD)算法[24]. 将差分隐私与矩阵因式分解推荐相结合的典型算法,采用随机梯度下降法更新因子矩阵,在每次迭代过程中加入拉普拉斯噪声. ②基于一般差分隐私保护的概率矩阵分解(Differentially Private Probabilistic Matrix Factorization, DP-PMF)算法[8]. 未引入时间权重因子,通过目标扰动法对PMF的目标函数进行扰动. ③基于个性化差分隐私保护的概率矩阵分解(Personalized Differentially Private Probabilistic Matrix Factorization, PDP-PMF)算法[8]. 将用户分为不同隐私关注人群并以此划分隐私预算,根据评分项的隐私预算对原始数据进行随机抽样,并对抽样后的数据采用一般的差分隐私保护方案. ④基于交替最小二乘法输出加扰的矩阵分解(Private Alternating Least Squares, PALS)算法[25]. 将差分隐私与矩阵因式分解相结合,采用交替最小二乘法更新因子矩阵,并对输出进行扰动. 为合理地进行比较,算法②和算法④均只对项目因子矩阵进行扰动. 根据5.1节实验结果确定T_0和T_1,在Movielens-100k、Movielends-1M、Movielens-10M、Epinions、Amazon-Books数据集上,测试所有算法在不同隐私预算下的RMSE. 结果如图4所示.

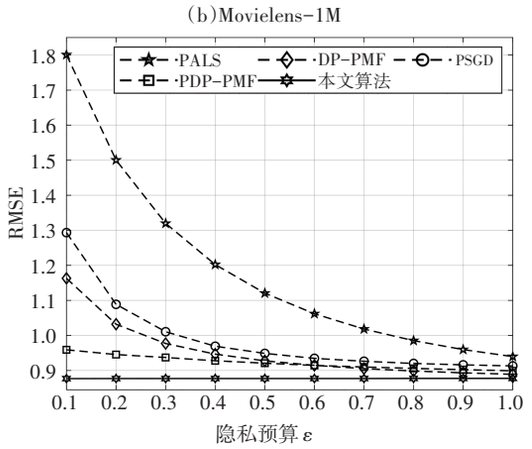
图4呈现了不同数据集上所有算法的性能表现. 整体上看,随着ε增加,除本文算法外的其他算法预测精确度提升. 这体现出差分隐私的性质,隐私预算越大,数据可用性越强,精度越高. 而本文算法对ε的变化并不敏感. 本文算法具有这种特性,主要是由于本文算法根据式(9)分配隐私预算时结合了时间权重因子的影响,对单个评分项进行了个性化的隐私保护,ε的增加主要降低部分近期评分隐私保护强度,平均隐私预算$\bar{\varepsilon}$变化较小,导致算法的整体性能波动较小.



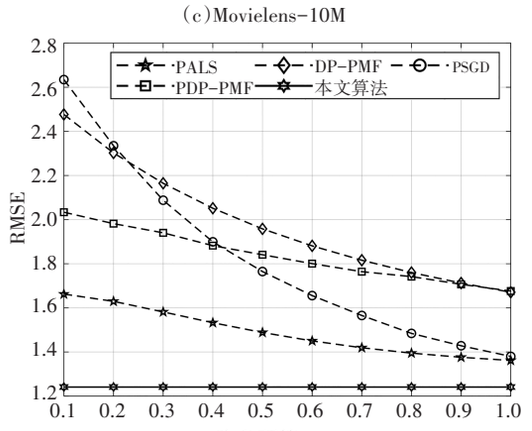
(a)Movielens-100K



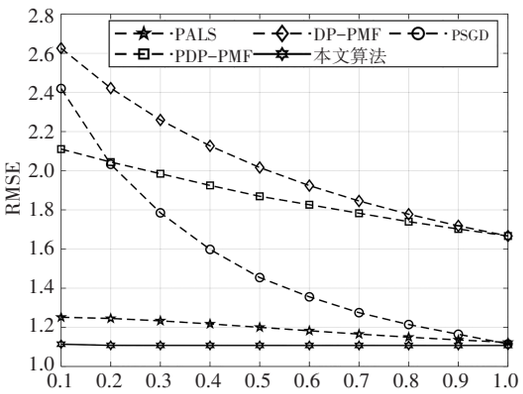
(b)Movielens-1M



(c)Movielens-10M



(d)Epinions



(e)Amazon-Books

图4　不同算法性能对比

Fig.4　Performance for different algorithm

在不同数据集中,本文算法性能表现均优于其他算法. 以图4(b)为例,本文算法RMSE在隐私预算$\varepsilon = 1$时比其他算法中性能最优的算法(PSGD)低0.084. 并且,本文算法对隐私不敏感的这种特性使得算法在高隐私保护水平下,准确性优势更加明显. 例如,在$\varepsilon = 0.1$时,本文算法的RMSE比PSGD算法低0.58.

此外,本文算法在大规模、更稠密的数据集上有更好的效果. 例如,在$\varepsilon = 0.1$时,本文算法的RMSE由Movielens-1M算法的0.975 2下降到Movielens-10M算法的0.876 8. 其他算法的准确性在相同条件下也有增长,比如PSGD算法的RMSE由Movielens-1M数据集的1.490下降到Movielens-10M数据集的1.293. 但是,由于其他算法忽略了时间因素的影响,容易引入过量的噪声,在大规模数据集上,效果仍然不如本文算法. 如在Amazon-Books数据集上,当$\varepsilon = 0.1$时,本文算法的RMSE比性能最好的算法(PALS)低0.138. 如上所述,本文算法在Movielens-10M以及Amazon-Books数据集上的表现验证了其应用于大规模数据集上的潜力.

5.3　效率对比

为了分析本文算法的效率,对本文算法与PALS、PSGD、DP-PMF、PDP-PMF算法进行分析,从理论和实验两个方面分析时间和计算开销.

理论上. 由4.2节可知,本文算法的时间复杂度为$O[N(M+\omega)]$或者$O[M(N+\omega)]$,算法的时间复杂度与用户和项目数量乘积NM成正比. PSGD、PALS与DP-PMF算法按照统一的隐私预算添加噪声,无须在原算法增加额外步骤,故两个算法时间复杂度均为$O(NM)$. PDP-PMF算法在DP-PMF的基础上增加了用户隐私预算分配和评分采样两个步骤,增加的计算复杂度为NM,故算法的时间复杂度仍为$O(NM)$. 理论分析表明,尽管本文算法增加了时间复杂度$O(N\omega)$或者$O(M\omega)$,但与其他算法仍然在同一数量范围$O(kNM)$内,k为常数.

从训练时间和预测时间进行实验分析. 训练时间指算法模型训练完成耗费的时间,可以在用户使用系统前完成;预测时间指系统推荐预测某个用户评分耗费的时间,也是用户等待的时间. 实验开销对比如表3所示. 随着数据规模的增加,所有算法耗费的时间均增多. 整体上看,本文算法的时间比PALS和PSGD要少,主要是因为PALS和PSGD算法均对数据进行了预处理,计算了每个用户和项目的均值. 在大规模数据集上,这种预处理会随着用户和项目数量增大耗费更多时间. 此外,PALS比PSGD耗费时间多是由于PALS采用交替最小二乘法进行优化,增

表3　实验开销对比

Tab.3　Comparison of experimental costs

数据集	本文算法		PALS		DP-PMF		PSGD		PDP-PMF	
	训练时间/s	预测时间/ms	训练时间/s	预测时间/ms	训练时间/s	预测时间/ms	训练时间/s	预测时间/ms	训练时间/s	预测时间/ms
Movielens-100K	10.86	3.96	17.60	2.96	7.26	2.95	19.57	2.97	8.88	3.99
Movielens-1M	88.45	23.27	224.22	24.12	83.43	12.83	221.19	12.36	103.44	23.66
Movielens-10M	8 100.90	125.27	17 411.68	177.40	2 603.01	105.35	2 200.80	163.13	2 764.66	106.65
Epinions	205.24	18.76	610.92	17.79	182.24	8.97	330.87	16.22	204.98	18.50
Amazon-Books	36 395.39	202.67	78 531.64	135.05	32 732.40	103.33	5 669.77	163.92	35 705.55	180.06

加了对矩阵的求逆步骤. PDP-PMF 和 DP-PMF 不需要对数据进行预处理,两个算法的时间整体上均少于 PALS 和 PSGD 算法. 此外,由于 PDP-PMF 算法增加了两个步骤,耗费时间比 DP-PMF 多. 而本文算法由于增加了时间权重因子计算步骤和隐私预算分配,本文算法的时间整体上多于 PDP-PMF 和 DP-PMF.

在相同数据集上,与其他算法相比,尽管本文算法增加了计算时间,但整体计算开销差距并不大. 例如,在 Movielens-10M 数据集上,本文算法预测时间比 DP-PMF 算法多 19.92 ms;训练时间为 8 100.90 s,比 PSGD 的 2 200.80 s 增加约 3.6 倍. 说明尽管本文算法比其他算法耗费的时间更多,但仍然处于同样的数量级别. 在 Movielens-100K 数据集上,本文算法的训练时间为 10.86 s,预测时间为 3.96 ms. 而在 Amazon-Books 数据集上的训练时间为 36 395.39 s,预测时间为 202.67 ms. 说明算法数据的增加更多的是增加模型训练的时间,而对用户偏好的预测时间影响不大.

6 结 论

本文算法的核心思想是从用户兴趣漂移角度出发,解决现有隐私保护推荐算法忽略时间的影响导致推荐质量下降的问题. 通过构建时间权重因子来衡量信息的重要性,并根据重要性对不同时间段的评分数据采用不同强度隐私保护. 对推荐系统进行隐私保护时,这种方式能够减少不必要噪声的引入. 此外,分别从理论和实践上证明算法可行性. 首先从理论角度证明本文算法的安全性,随后,通过实验表明,本文算法即使在较强的隐私预算下也能保证良好的预测精度,并且其推荐结果的准确度也比经典的差分隐私推荐算法更高,具有良好的应用前景.

参考文献

[1] 刘胜宗,樊晓平,廖志芳,等. 基于 PMF 进行潜在特征因子分解的标签推荐[J]. 湖南大学学报(自然科学版),2015,42(10):107-113.
LIU S Z, FAN X P, LIAO Z F, *et al*. A tag recommending algorithm with latent feature factor jointly factorizing based on PMF [J]. Journal of Hunan University (Natural Sciences), 2015, 42(10):107-113.(In Chinese)

[2] 刘纵横,汪海涛,姜瑛,等. 基于混合神经网络的序列推荐算法[J]. 重庆邮电大学学报(自然科学版),2021,33(3):466-474.
LIU Z H, WANG H T, JIANG Y, *et al*. Sequence recommendation algorithm based on a hybrid neural network[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2021, 33(3):466-474.(In Chinese)

[3] LIU J, QIN F L. Protection of user data by differential privacy algorithms[J]. International Journal of Network Security, 2020, 22(5): 838-844.

[4] ZHU T Q, LI G, REN Y L, *et al*. Differential privacy for neighborhood-based collaborative filtering [C]// Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Niagara Falls, ON, Canada: IEEE, 2013: 752-759.

[5] YANG S X, ZHU K L, LIANG W. Differential privacy for context-aware recommender systems[C]//2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing. Milan, Italy: IEEE, 2019: 356-360.

[6] MCSHERRY F, MIRONOV I. Differentially private recommender systems: building privacy into the net[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. Paris: ACM, 2009: 627-635.

[7] YANG M M, ZHU T Q, XIANG Y, *et al*. Personalized privacy preserving collaborative filtering [M]//Green, Pervasive, and Cloud Computing. Cham: Springer International Publishing, 2017: 371-385.

[8] ZHANG S, LIU L X, CHEN Z L, *et al*. Probabilistic matrix factorization with personalized differential privacy [J]. Knowledge-Based Systems, 2019, 183: 104864.

[9] 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 Z Z, LI Q L, HUANG X Y, *et al*. Collaborative filtering via SVD++ with differential privacy[J]. Control and Decision, 2019, 34(1):43-54.(In Chinese)

[10] 郑剑,王啸乾. 融合标签相似度的差分隐私矩阵分解推荐算法[J]. 计算机应用研究,2020,37(3):851-855.
ZHENG J, WANG X Q. Differential privacy matrix factorization recommendation algorithm fusing tag similarity [J]. Application Research of Computers, 2020, 37(3):851-855.(In Chinese)

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

[12] CHEN Y C, HUI L, THAIPISUTIKUL T. A collaborative filtering recommendation system with dynamic time decay [J]. The Journal of Supercomputing, 2021, 77(1): 244-262.

[13] ZAREIE A, SHEIKHAHMADI A, JALILI M. Identification of influential users in social networks based on users interest[J]. Infor-

mation Sciences, 2019, 493: 217-231.

[14] CAO Y L, LI W L, ZHENG D X. A hybrid recommendation approach using LDA and probabilistic matrix factorization[J]. Cluster Computing, 2019, 22(4): 8811-8821.

[15] DWORK C, ROTH A. The algorithmic foundations of differential privacy[J]. Foundations and Trends in Theoretical Computer Science, 2013, 9(3/4): 211-407.

[16] JORGENSEN Z, YU T, CORMODE G. Conservative or liberal? personalized differential privacy [C]//2015 IEEE 31st International Conference on Data Engineering. Seoul, Korea (South): IEEE, 2015: 1023-1034.

[17] 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. Denver, Colorado, USA: ACM, 2015: 1322-1333.

[18] 费洪晓,戴弋,穆珺,等. 基于优化时间窗的用户兴趣漂移方法[J]. 计算机工程,2008,34(16):210-211.
FEI H X, DAI Y, MU J, *et al*. Method of drifting user's interests based on time window optimization[J]. Computer Engineering, 2008, 34(16): 210-211. (In Chinese)

[19] PAN H L, WANG J B, ZHANG Z J. A movie recommendation model combining time information and probability matrix factorisation[J]. International Journal of Embedded Systems, 2021, 14(3): 239-247.

[20] JIANG W J, CHEN J H, JIANG Y R, *et al*. A new time-aware collaborative filtering intelligent recommendation system[J]. Computers, Materials & Continua, 2019, 61(2): 849-859.

[21] 兰艳,曹芳芳. 面向电影推荐的时间加权协同过滤算法的研究[J]. 计算机科学,2017,44(4):295-301.
LAN Y, CAO F F. Research of time weighted collaborative filtering algorithm in movie recommendation[J]. Computer Science, 2017, 44(4): 295-301. (In Chinese)

[22] CHEN J R, WEI L D, ULIJI, *et al*. Dynamic evolutionary clustering approach based on time weight and latent attributes for collaborative filtering recommendation[J]. Chaos, Solitons & Fractals, 2018, 114: 8-18.

[23] BELLOGÍN A, CANTADOR I, DÍEZ F, *et al*. An empirical comparison of social, collaborative filtering, and hybrid recommenders [J]. ACM Transactions on Intelligent Systems and Technology, 2013, 4(1): 1-29.

[24] 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. Vienna, Austria: ACM, 2015: 107-114.

[25] 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): 425-458.