文章编号: 1001-0920(2019)01-0043-12 DOI: 10.13195/j.kzyjc.2017.0961

基于差分隐私和SVD++的协同过滤算法

鲜征征[1†], 李启良[2], 黄晓宇[3], 吕 威[4], 陆寄远[1]

(1. 广东金融学院 互联网金融与信息工程学院，广州 510521；2. 华为技术有限公司，广东 深圳 518129；
3. 华南理工大学 经济与贸易学院，广州 510006；4. 北京师范大学珠海分校 信息技术学院，广东 珠海 519085)

摘 要: 协同过滤技术在推荐系统的实现中具有广泛的应用,协同过滤以用户对商品项目的评价分数为依据,而这些评价有可能反映用户某些不欲为人知的喜好特点,因此,对具备隐私保护能力的协同过滤模型的研究引起了普遍的关注. SVD++是当前最为常用的协同过滤模型之一,差分隐私模型则是近十年来隐私保护理论最重要的研究进展之一,将两者相结合提出3种基于差分隐私和SVD++的协同过滤模型:基于梯度扰动的SVD++隐私保护模型、基于目标函数扰动的SVD++隐私保护模型和基于输出结果扰动的隐私保护模型. 理论分析和实验结果显示,所提出的算法不仅能为用户的隐私安全提供可靠的保障,而且还可保持较高的预测准确度.

关键词: 协同过滤；隐私保护；差分隐私；矩阵分解

中图分类号: TP181 **文献标志码:** A

Collaborative filtering via SVD++ with differential privacy

XIAN Zheng-zheng[1†], *LI Qi-liang*[2], *HUANG Xiao-yu*[3], *LYU Wei*[4], *LU Ji-yuan*[1]

(1. School of Internet Finance and Information Engineering，Guangdong University of Finance，Guangzhou 510521，China；2. Huawei Technologies Co., Ltd.，Shenzhen 518129，China；3. School of Economics and Commerce，South China University of Technology，Guangzhou 510006，China；4. School of Information Technology，Beijing Normal University at Zhuhai，Zhuhai 519085，China)

Abstract: Collaborative filtering(CF), as a technique that automatically predicts the interest of an user by collecting rating information from other similar users or items, has been widely deployed in various recommendation systems. However, CF prediction is based on the users' historical ratings, indicating that it may reflect some of the users' private preferences. Consequently, enhancing CF model with privacy preservation guarantee has attracted much research attention. In this paper, we propose three privacy preserving collaborative filtering algorithms: DPSS++, DPSAObj++ and DPSAOut++. All the algorithms are based on SVD++, one of the most used CF algorithms, and differential privacy model, one the most important advance in the area of privacy preserving in the last decade. Our analysis shows that the proposed algorithms not only can provide reliable guarantee in terms of privacy preserving, but also keep high prediction accuracy.

Keywords: collaborative filtering；privacy preserving；differential privacy；matrix factorization

0 引 言

协同过滤(Collaborative filtering)[1]是在线推荐系统的基础实现技术之一,通过协同过滤技术推荐系统能够基于用户对某些商品给出的历史评价记录,获得他们对新商品的喜好程度的估计,进而可以根据预测的结果为用户提供合适的推荐项目,由此实现销售业绩的提升. 当前,协同过滤技术已经得到了广泛的应用,其采纳者甚至包括一些业界的巨头[2-3]. 然而,在另一方面,由于协同过滤技术以用户主动贡献的行为数据(如用户对某部电影或某本图书的评分)为基础,因而相应地,基于这些数据学习获得的结果也存在用户行为信息被泄露的可能. 如在平凡的情况下,对于每一个待评分的项目,都把它现有的评分的均值作为所有缺失的用户评分的估计,考虑下述情形:假设电影《大话西游》目前已有100个用户参与评分,平均分为4.5(5级评分制,1分最低,5分最高),则

收稿日期: 2017-07-18；**修回日期:** 2017-09-24.

基金项目: 广东省自然科学基金项目(2014A030313662, 2016A030310018, 2016A030313385)；广东省公益研究与能力建设项目(2015A030402003)；广东省科技项目(2016ZC0039)；广东省哲学社科项目(GD15CGL05)；华南理工大学中央高校业务经费项目(2015QNXM20).

责任编委: 阳春华.

作者简介: 鲜征征(1977−), 女, 讲师, 博士, 从事机器学习及隐私保护理论等研究；李启良(1990−), 男, 硕士, 从事机器学习和隐私保护研发应用的研究.

[†]通讯作者. E-mail: xianzhengzheng@126.com.

对于所有未就该电影给出评价的用户,都把他们的分数视为4.5分. 但若有一位新用户也参与了评价,并且在他评价后,《大话西游》的平均分被更新为4.47分,则即使该用户没有公布他给出的分数,也可以根据$\llcorner 101\times 4.47-100\times 4.5\lrcorner=1$推知该用户的分数(这里"$\llcorner\lrcorner$"是向下取整符号). 进一步可以猜测,该用户可能不喜欢周星驰式的夸张、无厘头的电影. 特别地,若用户对某些敏感题材的电影给出了高分评价,则有可能反映出用户对这些题材有特殊的喜好,而用户本人可能并不希望他的这些喜好为人所知.

近年来,随着大数据相关的研究和应用的发展,用户个人隐私信息泄露的风险也随之提高. 例如,在Netflix百万美金协同过滤大赛中,虽然组织者对发布的数据已经仔细地作了匿名处理,但德克萨斯大学奥斯汀分校的研究者通过把Netflix的评分数据与另一著名的电影评分网站IMDB的用户署名的评分数据进行交叉对比,仍然成功地破译了Netflix用户的身份信息[4]. 这一事件掀起了轩然大波,Netflix公司一夜之间被诉诸法庭,其后续的竞赛计划也随之流产.

Netflix事件表明,以个体数据作为数据发布粒度的做法对用户隐私安全的威胁是巨大的,因此,Dwork[5]提出的差分隐私机制(Differential privacy, DP)一面世,便引起了广泛的重视. 差分隐私机制充分考虑了数据的可用性与用户的隐私安全性两者间的折衷,提出了对用户的隐私保护应构建在聚合数据的基础之上,这里的"聚合数据"既可以是原始数据的某个统计量(如评分的平均值),也可以是基于原始数据学习获得的结果(如支持向量机的分割平面[6]). 如在Netflix竞赛问题背景下,差分隐私机制可以使用匿名用户提交的所有评分的平均值这一"聚合"的数据代替他们对具体每一项目的评分,进一步地,为了杜绝用户对个体对象评分泄露的可能,对于上述"聚合"的结果,差分隐私机制还将施以若干统计噪音干扰,干扰噪音的选择一般需遵循如下两个原则: 1) 能为用户的隐私安全提供理论保证; 2) 能在实际应用中保持数据的可用性.

对于差分隐私的研究在近十年取得了丰硕的成果,Dwork等[7]在综述中对这些结果进行了全面的介绍. 尤为可喜的是,近年来,差分隐私机制在实际应用中也开始取得突破(2014年,谷歌公司宣布,为提升用户的上网体验,他们已经启动了一个名为"RAPPOR"(Randomized aggregatable privacy-preserving ordinal response)的项目,采用差分隐私保护技术从Chrome浏览器采集用户的上网行为. 2016年6月,苹果公司软件工程高级副总裁Craig Federighi也在苹果全球开发者大会上宣布,苹果公司将自iOS 10开始,在不侵犯用户个人隐私的前提下,通过差分隐私技术获取用户对苹果产品的使用模式,以改进其产品设计.

本文主要研究协同过滤中的隐私保护问题. 自Netflix用户身份攻击事件[4]以来,该问题已经吸引了较多的关注,相关工作包括文献[8]提出的对协方差矩阵注入噪音的保护策略,文献[9]提出的对评分矩阵作低秩近似的保护策略,文献[10]提出的基于原始评分数据加噪的策略和文献[11]提出的对优化目标进行加扰的策略等. 文献[12]提出了一个具有社会意识的差分隐私保护协同过滤算法; 文献[13]提出了将差分隐私应用于矩阵分解的各个步骤之中,并对原始评分矩阵作相关预处理. 另一方面,众所周知,由Koren等[14]提出的SVD++模型由于融合了用户特征、评价对象特征以及用户反馈等因素,在包括Netflix竞赛在内的诸多协同过滤任务中都取得了杰出的表现,但在现有的研究中,与SVD++模型的安全性相关的工作尚不多见,因此,本文将致力于基于SVD++的差分隐私机制的研究. 需要强调的是,由于差分隐私机制是与算法的具体实现相关的,而对于SVD++模型的求解,当前有梯度下降策略和交替最小二乘法迭代策略这两种不同的主要实现方式,所以本文的研究也将围绕着两种不同的实现展开.

本文首先介绍相关的知识背景; 然后分别从梯度扰动、目标函数扰动、输出结果扰动3个角度出发,提出SVD++模型的3种差分隐私机制设计,同时进行相关实验研究并对结果进行分析; 最后对全文工作进行总结和展望.

1 知识背景

1.1 SVD++模型

SVD++是协同过滤研究中最为常用的算法模型之一. 记$R\in \boldsymbol{R}^{n\times m}$为由$n$名用户对$m$个项目给出的评分矩阵,其中用户$u$对项目$i$给出的分数记为$r_{u,i}$,特别地,若$r_{u,i}$未被采集获得(以下称为"缺失"),则令$r_{u,i}=0$. 协同过滤研究的主要任务是如何根据$R$中非0的评分数据,对缺失的数据给出合理的估计. 针对上述问题,Koren等[15]提出了如下SVD++模型:

$$\begin{aligned}\min_{\mu,U,I,P,Q,Y} F=&\Big\{\sum_u\sum_i\Big(r_{u,i}-\mu-b_u-b_i-\\&q_i^{\mathrm{T}}\Big(p_u+|X(u)|^{-0.5}\sum_{j\in X(u)}y_j\Big)\Big)^2\Big\}+\\&\lambda\Big(\mu^2+\sum_u b_u^2+\sum_i b_i^2+\sum_u||p_u||^2+\\&\sum_i||q_i||^2+\sum_u\sum_{j\in X(u)}||y_j||^2\Big).\end{aligned}\tag{1}$$

其中: $P \in \boldsymbol{R}^{n\times d}$ 是用户特征矩阵,记 P 的第 u 个行向量为 p_u,它对应用户 u 的一个 d 维的特征描述;类似地, $Q \in \boldsymbol{R}^{d\times m}$ 是被评价对象的特征矩阵,它的第 i 个列向量 q_i 对应了项目 i 的 d 维特征表示,这里一般取 $d \ll \min\{n,m\}$. $U=[b_1,b_2,\cdots,b_n]'$ 是用户评分的偏置向量,其分量 b_u 对应用户 u 的评分偏好相对于所有用户对所有项目给出的评分的平均分数 μ 的偏置; $I=[b_1,b_2,\cdots,b_m]'$ 是项目得分的偏置向量,其分量 b_i 也对应了项目 i 的得分相对于 μ 的偏置;对于每个 $u\in\{1,2,\cdots,n\}$,式(1)使用 $X(u)$ 记录 u,给出了评分的项目的集合;对于每个 $j\in X(u)$, d 维向量 y_j 对应了基于 u 对 j 的评分行为而得到的隐式的特征反馈; $Y=X(1)\bigcup X(2)\bigcup\cdots X(n)$ 是所有隐式反馈向量集合之并. 在式(1)中仅考虑有效评分.

对于目标(1)的求解,可以使用随机梯度下降(Stochastic gradient descent, SGD)和交替最小二乘(Alternating least squares, ALS)两种常见优化算法[15]. 针对这两种算法,本文将分别提出与之相对应的SVD++差分隐私保护策略.

1.2　差分隐私保护

差分隐私保护机制是Dwork等于2006年提出的一种面向聚合型数据(Aggregate data)的保护机制,它期望能通过对待发布的聚合数据(如某个单位所有职工的平均年收入)施以适当的(随机)噪音扰动,使之既能保护个体用户的隐私信息,又能保持对数据分析的可用性.

差分隐私力图实现在“所有可能被公开的数据都已公开”的情况下,仍能实现对目标对象的隐私的严格保护,具体到本文的研究,有如下定义[16-20].

定义1　相邻评分矩阵. 记 R^1 和 R^2 是由相同的 n 名用户对 m 个项目给出的(可能是不完全的)评分矩阵,当且仅当它们两者间至多只有一个相异的元素时, R^1 和 R^2 是“相邻”的.

例1　考虑由两名用户 $(u_1$、$u_2)$ 对3部电影 $(i_1\sim i_3)$ 给出的评分矩阵 R^1 和 R^2,其中 R^1 包含了 u_1 对 i_1、u_2 对 i_2 的评分, R^2 除了这两个评分外,还包含了 u_1 对 i_3 的评分,则 R^1 和 R^2 是相邻的.

定义2　函数的全局敏感度. 给定以评分矩阵 R 为输入的预测函数 $f:\boldsymbol{R}^{n\times m}\to\boldsymbol{R}^{n\times m}$,则 f 的 L_n 全局敏感度定义为

$$Z_f=\max_{R^1,R^2}||f(R^1)-f(R^2)||_n. \tag{2}$$

其中: L_n 是 n 范数, R^1 和 R^2 是任意两个相邻的评分矩阵.

定义3　差分隐私. 记随机算法 $G:\boldsymbol{R}^{n\times m}\to\boldsymbol{R}^{n\times m}$ 的值域为Range(G),对任意输入给定两个相邻的评分矩阵 R^1、R^2 和非负实数 ϵ,若对于任意 $S\subset\mathrm{Range}(G)$,都有

$$\prod_{u,i}\frac{\Pr[G(R^1)_{u,i}\in S]}{\Pr[G(R^2)_{u,i}\in S]}\leqslant\exp(\epsilon), \tag{3}$$

则称 G 满足 ϵ-差分隐私.

定义3的本质是刻画基于 G 的输出的两个相邻矩阵的不可区分程度. 设 G 满足 ϵ-差分隐私且令 $\epsilon\to0^+$,对于离散的Range(G),令 $|S|=1$,对于连续的Range(G),令 $S=(t,t')$,这里 $|t-t'|\to0$. 上述定义显示, $G(R^1)$ 与 $G(R^2)$ 的每一对相同位置上的元素都以近似1的概率彼此相等或充分接近. 因此,对于攻击者而言,若仅以 G 的输出结果为依据,则难以定位确切的 R^1 的取值.

拉普拉斯机制(Laplace mechanism)是最常见的差分隐私保护机制之一,其基本的策略是通过向聚合数据中注入服从Laplace分布的噪音变量,实现对个体数据的保护.本文的研究依赖于如下结论.

定理1　Laplace机制满足差分隐私. 给定函数 $f:\boldsymbol{R}^{n\times m}\to\boldsymbol{R}^{n\times m}$,对于任意矩阵 $R\in\boldsymbol{R}^{n\times m}$,定义Laplace机制 $G:\boldsymbol{R}^{n\times m}\to\boldsymbol{R}^{n\times m}$ 如下:

$$G(f(R))_{u,i}=f(R)_{u,i}+\zeta_{u,i}. \tag{4}$$

其中: $\zeta_{u,i}$ 服从均值为0、方差为 $2(Z_f/\epsilon)^2$ 的Laplace分布,即 $\Pr(\zeta_{u,i})=\frac{\epsilon}{2Z_f}\exp\Big(-\frac{\epsilon\zeta_{u,i}}{Z_f}\Big)$(以下记为 $\zeta_{u,i}\sim\mathrm{Lap}(Z_f/\epsilon)$),则 G 满足 ϵ-差分隐私.

证明　记 $g_{u,i}=G(f(R))_{u,i}$,为方便,记 $g_{u,i}$ 为 g,由

$$\Pr(g=r)=\frac{\epsilon}{2Z_f}\exp\Big(-\frac{\epsilon|g-r|}{Z_f}\Big),$$

$$\Pr(g=r')=\frac{\epsilon}{2Z_f}\exp\Big(-\frac{\epsilon|g-r'|}{Z_f}\Big),$$

可得

$$\frac{\Pr(g=r)}{\Pr(g=r')}=\exp\Big(\epsilon\frac{|g-r'|-|g-r|}{Z_f}\Big)\leqslant\exp\Big(\epsilon\frac{|r'-r|}{Z_f}\Big).$$

由此有

$$\prod_{u,i}\frac{\Pr(g=r)}{\Pr(g=r')}\leqslant\exp\Big(\epsilon\frac{\sum|r'-r|}{Z_f}\Big)\leqslant\exp(\epsilon).\quad\square$$

一个复杂的隐私保护问题通常会多次应用差分隐私保护技术,此时,为保证整个过程的隐私保护水平是控制在给定的隐私保护预算(ϵ)之内,文献[21]给出了差分隐私保护的两个重要特性:序列组合特性和并行组合特性. 序列组合特性即为多个随机算法分别分配隐私保护预算且满足相应的差分隐私,那么这些算法的组合算法在同一数据集上将满足隐私

保护预算之和的差分隐私,而并行组合体特性则是这些算法的组合算法在不相交的数据集上将满足最大隐私保护预算的差分隐私.

2 SVD++的差分隐私保护

由式(1)可知,SVD++模型求解的任务是根据已有的评分数据,获得对参数μ, b_u, b_i, P, Q, $\sum y_j$的估计,从而对于任意目标用户$u \in [n]$和目标项目$i \in [m]$,都可以使用下式获得u对i的评分估计:

$$\tilde{r}_{u,i} = \mu + b_u + b_i + q_i^{\mathrm{T}}\Big(p_u + |X(u)|^{-0.5}\sum_{j\in X(u)} y_j\Big). \tag{5}$$

注意SVD++的计算结果是对外公开发布的,对于任一用户u,若u更新了其评分记录(如新增了对某个项目的评价),则参数μ, b_u, b_i, P, Q, $\sum y_j$亦应随之修改. 另一方面,由于协同过滤应用场景具有开放性,这些修改也可以被公众用户感知,对于攻击者而言,他们有可能根据这些参数细微的变化实现对u更新的评分记录的破译. 因此,有必要对SVD++计算的结果(即参数)施以一定的扰动措施,以保证用户的评分行为的安全性.

SGD和ALS是SVD++模型求解中最常见的两种求解策略,本节将分别基于这两种策略,提出与之相对应的差分隐私保护算法. 其中,对于SGD,将提出一种基于梯度加扰的保护算法(算法1),对于ALS,将分别提出一种基于目标函数扰动的保护算法(算法2)和一种基于输出函数加扰的算法(算法3).

2.1 基于梯度扰动的SVD++隐私保护策略

SGD是最常用的SVD++求解策略之一[15],其基本思路是通过对模型参数作反向的梯度更新以获得局部最优解. 为实现对SVD++模型参数的保护,一个自然的基本思路是对SGD求解过程中获得的每个梯度都施以一定的噪音扰动,从而使攻击者无法定位准确的结果信息,具体如表1 ~ 表3所示.

表1 程序1

Procedure 1: dev($r, \hat{\mu}, \hat{b}_u, \hat{b}_i, \hat{q}, \hat{p}, \hat{y}$)	
/* 功能: 计算真实评分值r与预测值$\tilde{r}$两者之差.*/	
1	$\tilde{r} = \hat{\mu} + \hat{b}_u + \hat{b}_i + \hat{q}^{\mathrm{T}}(\hat{p} + \hat{y})$;
2	return $r - \tilde{r}$.

表2 程序2

Procedure 2: param Updatel()	
/* 功能: 更新参数μ, b_u, b_i, p, q, y.*/	
1	$\mu \leftarrow \mu + \gamma(e_{u,i} - \lambda\mu)$;
2	$b_u \leftarrow b_u + \gamma(e_{u,i} - \lambda b_u)$;
3	$b_i \leftarrow b_i + \gamma(e_{u,i} - \lambda b_i)$;
4	$p_u \leftarrow p_u + \gamma(e_{u,i}q_i - \lambda p_u)$;
5	$q_i \leftarrow q_i + \gamma\Big(e_{u,i}\Big(p_u + \sum_{j\in X(u)} y_i\Big)\Big)$;
6	for each $j \in X(u)$ do:
7	$y_j \leftarrow y_j + \gamma(e_{u,i}\|X(u)\|^{-0.5}q_i - \lambda y_j)$;
8	end

表3 算法1

Algorithm 1: 基于梯度扰动的SVD++隐私保持(DPSS++)	
/* 输入: $R \in \boldsymbol{R}^{n\times m}$: "$n$用户-$m$项目"评分矩阵(若$r_{u,i}$值缺失,则令$r_{u,i} = 0$);*/	
/*	d: 矩阵分解隐含特征矩阵的特征个数;*/
/*	λ: SVD++目标函数中的正则化参数;*/
/*	γ: 学习率;*/
/*	k: 梯度下降迭代次数;*/
/*	ϵ: 差分隐私保持预算参数;*/
/* 输出: $\mu, b_u, b_i, P, Q, \sum_{j\in X(u)} y_i$.*/	
1	初始化μ为所有有效评分的均值,b_u为所有用户的平均评分,b_i为所有项目的平均评分,P、Q为随机高斯矩阵,诸y_j为随机高斯向量;
2	for $l = 1:\ k-1$ do
3	for each $r_{u,i}$ do
4	$e_{u,i} = \mathrm{dev}\Big(r_{u,i}, \mu, b_u, b_i, \|X(u)\|^{-0.5}\sum_{j\in X(u)} y_j, P_u, Q_i\Big)$;
5	调用param Updatel()更新参数;
6	end
7	end
8	for each $r_{u,i}$ do
9	$e_{u,i} = \mathrm{dev}\Big(r_{u,i}, \mu, b_u, b_i, \|X(u)\|^{-0.5}\sum_{j\in X(u)} y_j, P_u, Q_i\Big)$;
10	根据$a \propto \exp(-\epsilon\|a\|_1/\Delta r)$生成Laplace随机噪声$a$;
11	令$e'_{u,i} = e_{u,i} + a$;
12	采用截断法把$e'_{u,i} = e_{u,i} + a$控制在$[-2, 2]$中;
13	令$e_{u,i} = e'_{u,i}$;
14	调用param Updatel()更新参数;
15	end
16	返回$\mu, b_u, b_i, P, Q, \sum_{j\in X(u)} y_j$.

对于算法1,有以下几点说明:

1) 输入参数k是预设的总迭代次数,算法1的2 ~ 7行对应前$k-1$次迭代,属于常规的基于交替梯度下降的SVD++求解;8 ~ 15行是最后一次迭代,是通过对梯度施加干扰噪音来实现对目标参数的隐私保护. 因此,由于隐私保护而造成的额外计算代价为$\theta(N)$,这里N是评分矩阵N中非零元素的总个数.

2) 第10行中的参数$\Delta r = r_{\max} - r_{\min}$,这里$r_{\max}$和$r_{\min}$分别对应"用户-项目"评分矩阵$R$中的最大值和最小值.

3) $e_{u,i} = r_{u,i} - \tilde{r}_{u,i}$表示原始评分与预测评分之间的误差. 第11行是对参数$e_{u,i}$施以Laplace噪音干扰,在实际应用中,为保证扰动结果的可用性,需把它限制在一个较合理的范围. 因此,本文通过实验可得:下界为-2,上界为2,即当$e_{u,i} \leqslant -2$时,$e_{u,i} = -2$;当当$e_{u,i} \geqslant 2$时,$e_{u,i} = 2$.

对于算法1的隐私保护性能,有如下结论.

定理2　算法1满足ϵ-差分隐私.

证明　首先,对于两个相邻评分矩阵R^1和R^2,由$e_{u,i} = r_{u,i} - \tilde{r}_{u,i}$和$\Delta r = r_{\max} - r_{\min}$可得$e_{u,i}$的$L_1$敏感度为

$$Z_{e_{u,i}} = \max ||e_{u,i}(R^1) - e_{u,i}(R^2)||_1 \leqslant \max ||(r_{u,i} - \tilde{r}_{u,i}) - (r'_{u,i} - \tilde{r}_{u,i})||_1 \leqslant \Delta r.$$

其次,根据Laplace机制(定理1)为误差$e_{u,i}$添加噪声向量$a \propto \exp(-\epsilon||a||_1/\Delta r)$,即有

$$e'_{u,i} = e_{u,i} + \mathrm{Lap}(Z_{e_{u,i}}/\epsilon) = e_{u,i} + \mathrm{Lap}(\Delta r/\epsilon). \quad \square$$

2.2　基于目标扰动的SVD++隐私保护策略

文献[22]中提出了目标函数加扰和输出结果加扰两种差分隐私保护算法,并分别应用于逻辑回归和支持向量机,其中的目标函数加扰算法已被证明在权衡隐私保护和预测准确率上效果更优. 本节将这两种加扰算法的思想应用于SVD++的另一常用求解策略ALS中.

对于一阶可导的具有c维自变量的凸函数$f: \boldsymbol{R}^c \to \boldsymbol{R}$,为求得$\min\limits_{x_1,x_2,\cdots,x_c} f(x_1,x_2,\cdots,x_c)$,ALS采用的基本思路是对于$l = 1,2,\cdots,c$,顺次令$\partial f/\partial x_l = 0$而固定其他元素进行求解. 由SVD++的目标函数(式(1))可知,采用ALS求解μ, b_u, b_i, P, Q和$\sum\limits_{j\in X(u)} y_i$的结果为

$$J(\mu,R) = \sum e_{u,i}^2 + \lambda\mu^2, \tag{6}$$

$$J(b_u,R) = \sum e_{u,i}^2 + \lambda b_u^2, \tag{7}$$

$$J(b_i,R) = \sum e_{u,i}^2 + \lambda b_i^2, \tag{8}$$

$$J(p_u,R) = \sum e_{u,i}^2 + n_u\lambda||p_u||_2^2, \tag{9}$$

$$J(q_i,R) = \sum e_{u,i}^2 + n_i\lambda||q_i||_2^2, \tag{10}$$

$$J\Big(\sum y_j,R\Big) = \sum (e_{u,i})^2 + \lambda||\sum y_j||_2^2. \tag{11}$$

其中,对于式(9)和(10),定义$n_u = |\{r_{v,i} \in R|v = u\}|$,$n_i = |\{r_{u,v} \in R|v = i\}|$.

算法2给出了基于ALS目标函数扰动的SVD++隐私保护策略.

表4　程序3

Procedure 3: param Update2(bool is Private)

/* 功能: 交替最小二乘法求解SVD++.*/
/* 说明: 若is Private为false,则为ALS实现的常规版本,对所有参数都作更新;*/
/* 若is Private为true,则为ALS的隐私保护版本,只更新$\mu, b_u, b_i, Q, \sum\limits_{j\in X(u)} y_j$.*/

1　$\mu = (\lambda+1)^{-1}\Big(r_{u,i} - b_u - b_i - q_i^{\mathrm{T}}\Big(p_u + |X(u)|^{-0.5}\sum\limits_{j\in X(u)} y_j\Big)\Big)$;
2　for $i = 1 : m$ do
3　　$b_i = (\lambda+1)^{-1}\sum\limits_{r_{u,i}\in r_{\cdot i}}\Big(r_{u,i} - \mu - b_u - q_i^{\mathrm{T}}\Big(p_u + |X(u)|^{-0.5}\sum\limits_{j\in X(u)} y_j\Big)\Big)$;
4　　for each $r_{u,i} \in r_{\cdot i}$ do
5　　　$v_u = \sum\limits_{r_{u,i}\in r_{\cdot i}}\Big(p_u + |X(u)|^{-0.5}\sum\limits_{j\in X(u)} y_j\Big)$;
6　　end
7　　$q_i = \Big(\sum\limits_{r_{u,i}\in r_{\cdot i}} v_u v_u^{\mathrm{T}} + \lambda n_i I\Big)^{-1} \times \Big(\sum\limits_{r_{u,i}\in r_{\cdot i}} (r_{u,i} - \mu - b_i - b_u)v_u\Big)$;
8　end
9　for $u = 1 : n$ do
10　　$b_u = \dfrac{1}{1+\lambda}\sum\limits_{r_{u,i}\in r_{u\cdot}}\Big(r_{u,i} - \mu - b_i - q_i^{\mathrm{T}}\Big(p_u + |X(u)|^{-0.5}\sum\limits_{j\in X(u)} y_j\Big)\Big)$;
11　　if !is Private then
12　　　$p_u = \Big(\sum\limits_{r_{u,i}\in r_{u\cdot}} q_i q_i^{\mathrm{T}} + \lambda n_u I\Big)^{-1}\Big(\sum\limits_{r_{u,i}\in r_{u\cdot}} (r_{u,i} - \mu - b_i - b_u)q_i - \sum\limits_{r_{u,i}\in r_{u\cdot}} q_i q_i^{\mathrm{T}}|X(u)|^{-0.5}\sum\limits_{j\in X(u)} y_j\Big)$;
13　　end
14　　$\sum\limits_{j\in X(u)} y_j = \Big(\sum\limits_{r_{u,i}\in r_{u\cdot}} |X(u)|^{-1} q_i q_i^{\mathrm{T}} + \lambda n_u I\Big)^{-1}\Big(\sum\limits_{r_{u,i}\in r_{u\cdot}} |X(u)|^{-0.5} q_i (r_{u,i} - \mu - b_i - b_u - q_i^{\mathrm{T}} p_u)\Big)$
15　end

表5 算法2

Algorithm 2: 基于目标函数扰动的SVD++隐私保持(DPSAObj++)

/* 输入: $R \in \boldsymbol{R}^{n\times m}$: "$n$用户-$m$项目"评分矩阵(若$r_{u,i}$值缺失,则令$r_{u,i}=0$);*/

/* d: 矩阵分解隐含特征矩阵的特征个数;*/

/* λ: SVD++目标函数中的正则化参数;*/

/* ϵ: 差分隐私保护预算参数;*/

/* k: 梯度下降迭代次数;*/

/* c: 计算松驰项的参数;*/

/* 输出: $\mu, b_u, b_i, P, Q, \sum\limits_{j\in X(u)} y_i$.*/

初始化μ为所有有效评分的均值,b_u为所有用户的平均评分,b_i为所有项目的平均评分,P、Q为随机高斯矩阵,诸y_j为随机高斯向量;

for $l=1:\ k-1$ do

  param Update2(false);

end

$\mu=(\lambda+1)^{-1}\Big(r_{u,i}-b_u-b_i-q_i^{\mathrm{T}}\Big(p_u+|X(u)|^{-0.5}\sum\limits_{j\in X(u)}y_j\Big)\Big)$;

for $i=1:m$ do

  $b_i=(\lambda+1)^{-1}\Big(r_{u,i}-\mu-b_u-q_i^{\mathrm{T}}\Big(p_u|X(u)|^{-0.5}\sum\limits_{j\in X(u)}y_j\Big)\Big)$;

  for each $r_{u,i}\in r_{\cdot i}$ do

    $v_u=\sum\limits_{r_{u,i}\in r_{\cdot i}}\Big(p_u+|X(u)|^{-0.5}\sum\limits_{j\in X(u)}y_j\Big)$;

end

$q_i=\Big(\sum\limits_{r_{u,i}\in r_{\cdot i}}v_uv_u^{\mathrm{T}}+\lambda n_iI\Big)^{-1}\times\Big(\sum\limits_{r_{u,i}\in r_{\cdot i}}(r_{u,i}-\mu-b_i-b_u)v_u\Big)$;

end

for $u=1:n$ do

  $\epsilon'=\epsilon-\log\Big(1+\dfrac{2c}{Nn_u\lambda}+\dfrac{c^2}{N^2(n_u\lambda)^2}\Big)$;

  if $\epsilon'>0$ then

    $t=0$;

  end

  else

    $t=\dfrac{c}{N(\mathrm{e}^{\epsilon/4}-1)}-n_u\lambda$;

    $\epsilon'=\epsilon/2$;

  end

  根据$a\propto\exp(-\epsilon'||a||_2/2)$生成Laplace随机噪声$a$;

  $p_u=\sum\limits_{r_{u,i}\in r_{u\cdot}}\Big(q_iq_i^{\mathrm{T}}+\lambda n_u+\dfrac{1}{2}tI\Big)^{-1}\Big(\sum\limits_{r_{u,i}\in r_{u\cdot}}(r_{u,i}-\mu-b_i-b_u)q_i-\sum\limits_{r_{u,i}\in r_{u\cdot}}q_iq_i^{\mathrm{T}}|X(u)|^{-0.5}\sum\limits_{j\in X(u)}y_j-\dfrac{1}{N}a\Big)$;

end

paramUpdate2(true);

返回$\mu, b_u, b_i, P, Q, \sum\limits_{j\in X(u)} y_j$.

对于算法2,有以下几点说明:

1) 算法2的设计策略是先执行一般的ALS求解过程以拟合SVD++的模型参数$\mu, b_u, b_i, P, Q, \sum\limits_{j\in X(u)} y_j$,继而对拟合的结果施行扰动以实现用户隐私保护. 其中,算法的2 ~ 4行是ALS求解过程:记式(1)中的和式为F,则过程Procedure 3中的参数更新策略顺次由以下6式求解获得:

$$\frac{\partial F}{\partial \mu}=0,\ \frac{\partial F}{\partial b_i}=0,\ \frac{\partial F}{\partial q_i}=0,$$
$$\frac{\partial F}{\partial b_u}=0,\ \frac{\partial F}{\partial p_u}=0,\ \frac{\partial F}{\partial \sum\limits_{j\in X(u)}y_j}=0.$$

2) 算法2的5 ~ 24行是隐私保护功能的实现:首先,对每一个用户u,生成一个Laplace随机变量a,并把式(9)改写为

$$J^{\mathrm{priv}}(p_u,R)=J(p_u,R)+\frac{1}{n}a^{\mathrm{T}}p_u. \tag{12}$$

考虑如下优化目标:

$$p_u^{\mathrm{priv}}=\arg\min_{p_u}J^{\mathrm{priv}}(p_u,R)+\frac{1}{2}t||p_u||_2^2. \tag{13}$$

这里的加项$\frac{1}{2}t||p_u||_2^2$是正则化因子,t的取值是由隐私保护参数ϵ与松弛项参数c共同决定的. 注意式(13)等号右端的和式对p_u为凸函数,所以令右端和式对p_u的偏导为0,可以求得p_u^{priv}的解. 进一步地,固定$p_1^{\mathrm{priv}},p_2^{\mathrm{priv}},\cdots,p_n^{\mathrm{priv}}$,顺次令$\frac{\partial F}{\partial b_i}=0,\frac{\partial F}{\partial q_i}=0,\frac{\partial F}{\partial b_u}=0$和$\frac{\partial F}{\partial\sum_{j\in X(u)}y_j}=0$,得到SVD++的扰动解.

3) 关于算法2中参数c的确定: 根据文献[22]的推论5和推论6,对式(13)中的损失函数$\ell(e_{u,i})=(e_{u,i})^2$求二阶导数,有$\ell'(e_{u,i})=\frac{\partial\ell(e_{u,i})}{\partial e_{u,i}}=2e_{u,i}$和$\ell''(e_{u,i})=\frac{\partial\ell'(e_{u,i})}{\partial e_{u,i}}=2$,令$|\ell''(e_{u,i})|\leqslant c$,得到$c=2$.

对于算法2的隐私保护性能,有如下定理.

定理3　算法2满足ϵ-差分隐私.

证明　对每一个$u\in[n]$,根据式(1),把式(9)中等式展开,改写为如下形式:

$$J(p_u,R)=\Big\{\sum_i\Big(r_{u,i}-\mu-b_u-b_i-q_i^{\mathrm{T}}\Big(p_u+|X(u)|^{-0.5}\sum_{j\in X(u)}y_j\Big)\Big)^2\Big\}+\lambda n_u||p_u||_2^2.$$

容易看出,上式中的经验误差因子$\sum_i\Big(r_{u,i}-\mu-b_u-b_i-q_i^{\mathrm{T}}\Big(p_u+|X(u)|^{-0.5}\sum_{j\in X(u)}y_j\Big)\Big)^2$二阶可导,正则化因子$||p_u||_2^2$可导且1-强凸,因而由文献[22]的定理6可知,由式(13)获得的p_u满足ϵ-差分隐私.

进一步地,注意到在算法2的隐私保护步骤(第25行),对参数$\mu,b_u,b_i,Q,\sum_{j\in X(u)}y_j$的更新都以目标(13)的求解结果为依据,所以更新后得到的预测误差必然不大于更新之前的预测误差,由此可以断言算法2满足ϵ-差分隐私. □

2.3　基于输出结果扰动的SVD++隐私保护策略

容易看出,算法1和算法2都是基于SVD++求解过程的隐私保护算法,因而也可以考虑通过对SVD++求解获得的结果进行扰动以实现隐私保护的目的,由此提出算法3.

表 6　算法3

Algorithm 3: 基于结果扰动的SVD++隐私保护(DPSAOut++)

/* 输入: $R\in\boldsymbol{R}^{n\times m}$: "$n$用户-$m$项目"评分矩阵(若$r_{u,i}$值缺失,则令$r_{u,i}=0$);*/
/*　　d: 矩阵分解隐含特征矩阵的特征个数;*/
/*　　λ: SVD++目标函数中的正则化参数;*/
/*　　ϵ: 差分隐私保护预算参数;*/
/*　　k: 梯度下降迭代次数;*/
/* 输出: $\mu,b_u,b_i,P,Q,\sum_{j\in X(u)}y_j$.*/

1　初始化μ为所有有效评分的均值,b_u为所有用户的平均评分,b_i为所有项目的平均评分,P、Q为随机高斯矩阵,诸y_j为随机高斯向量;
2　for $l=1:\ k$ do
3　　paramUpdate2(false);
4　end
5　$\Delta r=r_{\max}-r_{\min}$;
6　根据$a\propto\exp\Big(-\frac{\epsilon/5||a||_1}{\Delta r}\Big)$生成Laplace噪声$a$;
7　$\mu=\mu+a$;
8　for each $u\in[n]$ do
9　　根据$a\propto\exp\Big(-\frac{\epsilon/5||a||_1}{\Delta r}\Big)$生成Laplace噪声$a$;
10　　$b_u=b_u+a$;
11　　根据$a\propto\exp\Big(-\frac{\epsilon/5||a||_2}{2}\Big)\frac{n_u\lambda}{2q_{\max}\Delta r}$生成Laplace噪声$a$;
12　　$p_u=p_u+a$;
13　end
14　for each $i\in[m]$ do
15　　根据$a\propto\exp\Big(-\frac{\epsilon/5||a||_1}{\Delta r}\Big)$生成Laplace噪声$a$;
16　　$b_i=b_i+a$;
17　　根据$a\propto\exp\Big(-\frac{\epsilon/5||a||_2}{2}\Big)\frac{n_i\lambda}{2p_{\max}\Delta r}$生成Laplace噪声$a$;
18　　$q_i=q_i+a$;
19　end
20　返回$\mu,b_u,b_i,P,Q,\sum_{j\in X(u)}y_j$.

对算法3有以下几点补充说明:

1) 算法3的2 ∼ 4行是采用标准的ALS求解SVD++的过程.

2) 第5 ∼ 19行是隐私保护功能的实现: 分别对SVD++的输出结果μ, b_u, b_i, P, Q进行隐私保护处理.

3) 具体到算法3的隐私保护策略,它是对ALS求解SVD++的输出结果(μ, b_u, b_i, P, Q)进行加扰. 为了保证算法整个过程满足ϵ-差分隐私,将隐私保护参数ϵ平均分成5份(每一份为$\epsilon/5$).

4) 对于标量μ, b_u, b_i,由定义2可知,它们的L_1敏感度与L_2敏感度是相等的. 设有两个相邻评分矩阵R^1和R^2,3个函数从它们中得到的输出结果的最大区别为评分的最大差值. 因此,这些函数的L_1敏感度为$Z_\mu = Z_{b_u} = Z_{b_i} = \Delta r$.

5) 记$p_{\max}$和$q_{\max}$分别为$||p_u||_2$、$||q_i||_2$的上界,根据求解P和Q的ALS目标函数(式(9)和(10))可以断言,它们的解的L_2敏感度分别满足

$$Z_{p_u} \leqslant \frac{2q_{\max}\Delta r}{n_u\lambda},\ Z_{q_i} \leqslant \frac{2p_{\max}\Delta r}{n_i\lambda}.$$

下面给出定理4.

定理4 算法3满足ϵ-差分隐私.

证明 首先证明由算法3得到的p_u满足$\epsilon/5$-差分隐私. 对两个相邻的评分矩阵R^1和R^2,记根据式(9)得到的p_u分别为p_u^1和p_u^2,又记算法3中p_u^1和p_u^2生成的噪音向量分别为a_1和a_2,则有

$$a_1 - a_2 = p_u^1 - p_u^2.$$

从而有

$$||a_1||_2 - ||a_2||_2 \leqslant ||a_1 - a_2||_2 =$$
$$||p_u^1 - p_u^2||_2 \leqslant \frac{2q_{\max}\Delta r}{n_u\lambda}.$$

另一方面,注意

$$\frac{\Pr\{p_u^1|R^1\}}{\Pr\{p_u^2|R^2\}} = \exp\Big(-\frac{n_u\lambda\epsilon/5}{2q_{\max}\Delta r}(||p_u^1 - p_u^2||_2)\Big).$$

由此可知,由算法3得到的p_u满足$\epsilon/5$-差分隐私. 与上述过程完全类似,还可分别证明算法3得到的μ, b_i, b_u, q_i也各自满足$\epsilon/5$-差分隐私,因而算法3满足$\epsilon-$差分隐私. □

3 实验结果及分析

本文提出对基于SVD++的协同过滤算法进行差分隐私保护处理,具体是基于梯度扰动的SVD++隐私保护(算法1)、基于目标函数扰动的SVD++隐私保护(算法2)和基于输出结果扰动的SVD++隐私保护(算法3),通过两个真实数据集上的实验对算法进行验证,并与基于SVD的协同过滤和文献[13]的相关算法进行比较.

3.1 实验数据和设置

3.1.1 实验数据

为防止算法对某个数据集拟合,本文实验数据选用两个公开数据集,一是Movielens-1M(源自网址 http://grouplens.org/datasets/movielens/),另一个是从Netflix(源自网址http://www.netflixprize.com,比前者更稀疏)中截选的部分数据(本文称Netflix-1M),它们的统计属性分别如表7所示.

表7 实验数据集的统计属性

属性名	Movielens-1M	Netflix-1M
用户数	6 040	4 996
电影数	3 952	3 999
密度/%	4.19	0.19
平均评分	3.581 6	3.595 6
评分的方差	1.247 9	1.220 8

3.1.2 实验算法与评估指标

本文实验用到的算法汇总如表8所示.

表8 实验算法汇总

算法名称	描述
SGDBase++	无预处理,无差分隐私处理,SGD求解SVD++
ALSBase++	无预处理,无差分隐私处理,ALS求解SVD++
PSGD	文献[13]的算法4 (Differentially private SGD),有预处理, SGD 求解传统矩阵分解
PALS	文献[13]的算法5 (Differentially private ALS with output perturbation),有预处理,ALS求解传统矩阵分解
DPSS	无预处理, 基于梯度扰动的SVD隐私保护(参考本文算法1)
DPSAObj	无预处理, 基于目标函数扰动的SVD隐私保护(参考本文算法2)
DPSAOut	无预处理, 基于输出结果扰动的SVD隐私保护(参考本文算法3)
DPSS++	本文算法1, 无预处理, 基于梯度扰动的SVD++隐私保护
DPSAObj++	本文算法2, 无预处理, 基于目标函数扰动的SVD++隐私保护
DPSAOut++	本文算法3, 无预处理, 基于输出结果扰动的SVD++隐私保护

本文实验采用10-折交叉验证、训练和预测(训练集:验证集 = 9 : 1,且循环进行10次实验,最后取这10次实验结果的平均值). 为了评估做差分隐私保护处理之后SVD++的预测准确率,本文选用根均方误差(RMSE)来评估预测评分,RMSE越小意味着预测越准确,其计算公式为

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{r\in R}(r_{u,i}-\tilde{r}_{u,i})^2}{|R|}}.$$

3.1.3　参数设置及调节

本节阐述论文算法中几个重要参数的设置,包括隐含特征矩阵的特征个数d的确定、算法1中的截断区间的选取以及各算法中差分隐私保护参数ϵ的选择.

隐含矩阵的特征个数d的确定: 图1给出了在Movielens-1M数据集上,SVD和SVD++分别做梯度加扰和目标函数扰动两种差分隐私保护之后预测准确率与"隐含特征矩阵的特征个数"的关系. 注意这里其他参数取固定值$\lambda=0.125,\gamma=0.001,\epsilon=1,k=20$.

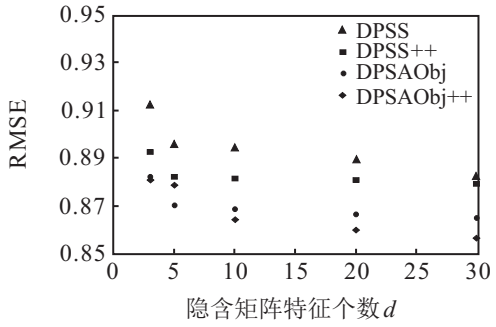


图1　矩阵隐含特征数(d)调节对预测结果的影响

从图1可看出,分别经过梯度扰动、目标函数扰动后的SVD和SVD++的预测准确率会随着隐含矩阵的特征个数稍加改善,但影响的幅度并不大,这是因为本文算法有Laplace随机噪声的作用. 除了$d=3$时各算法的RMES较差,当d取值分别为5, 10, 20, 30时,几个算法的RMSE的变化幅度($d\neq5$时的RMSE除以$d=5$时的RMSE)不会超过15%. 考虑到矩阵分解中特征个数较大,通常会导致过度拟合输入的矩阵,而且也会导致计算跟存储的复杂度过大[1],本文实验选择$d=5$.

参数截断区间选择: 算法1的第12行采用截断法把$e'_{u,i}=e_{u,i}+a$控制在合适的区间之内(本文把参数截断区间固定为$[-2,2]$). 为得到此区间,首先把它定义为0点对称区间$[-x,x]$,对于$n=0,1,2,3$,顺次令$x=0.5\times2^n$,并对于$\epsilon=0.1,\ 1,\ 2,\ 4$,分别把每一个(x,ϵ)组合应用于算法1来对Movielens-1M和Netflix-1M两个数据集进行预测,预测结果得到的RMSE分别如表9、表10所示.

表9　截断区间调节对RMSE的影响(Movielens-1M)

ϵ	$x=0.5$	$x=1$	$x=2$	$x=4$
0.1	0.907 6	0.897 5	0.891 1	0.905 4
1	0.906 3	0.894 5	0.882 0	0.897 3
2	0.891 1	0.882 0	0.880 1	0.890 0
4	0.888 9	0.881 2	0.879 1	0.884 6

表10　截断区间调节对RMSE的影响(Netflix-1M)

ϵ	$x=0.5$	$x=1$	$x=2$	$x=4$
0.1	0.995 6	0.994 5	0.981 3	0.986 9
1	0.993 3	0.986 5	0.972 6	0.981 5
2	0.985 7	0.978 2	0.963 6	0.975 1
4	0.983 2	0.971 2	0.960 2	0.972 5

从表9和表10可以看出,改变ϵ取值时,算法1对于不同x值产生的变化趋势大致相同,但均在$x=2$时得到的RMSE最小. 因此,把算法1中的截断区间取为$[-2,2]$.

差分隐私保护参数ϵ的选择: 本文算法的另一个重要调节参数为ϵ. 一方面,ϵ直接决定了隐私保护的力度;另一方面,过高的ϵ又影响了预测的准确率. 为了在隐私保护力度和预测准确率间取得平衡,本文采取如下处理策略:

1) 确定被推荐的用户对象;

2) 分别计算未做差分隐私处理和做某种差分隐私保护处理之后给1)中确定的用户对象推荐的物品集(本文实验为推荐电影集);

3) 求两个(做差分隐私和未做差分隐私)推荐物品集的交集;

4) 用该交集除以推荐物品集的总个数,得到一个百分比,该百分比的值越大意味着ϵ在这个取值时能够获得的预测准确率越高,此时的ϵ选择就应该相对合理.

该方案可以给出关于ϵ的一个较为合理的范围: 如果步骤4)得到的百分比小于20%,则认为虽然ϵ的隐私保护力度很强,但是已经严重影响了预测结果;如果百分比大于80%,则认为虽然有了良好的预测结果,但是隐私保护力度却过弱. 因此,使该百分比在20%～80%之间的ϵ取值是比较合理的.

图2给出了文献[13]的ALS输出结果加扰、SVD和SVD++分别按照目标函数扰动策略(算法2)在Movielens-1M上应用该选择方案的比较结果. 该实

验中,实验推荐电影集的个数设为30,随机任选一个用户作为推荐的对象. 为消除Laplace噪声的随机性的影响,重复10次实验,并取它们的结果的平均值作为最终的计算结果.

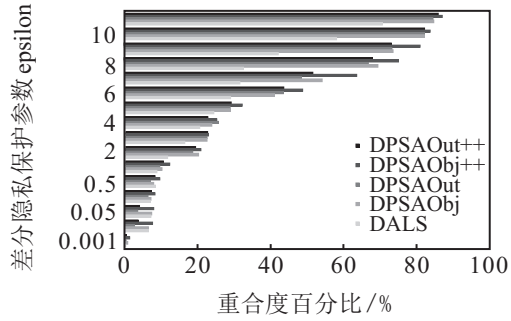


图 2 隐私保护参数(ϵ)调节对预测结果的影响

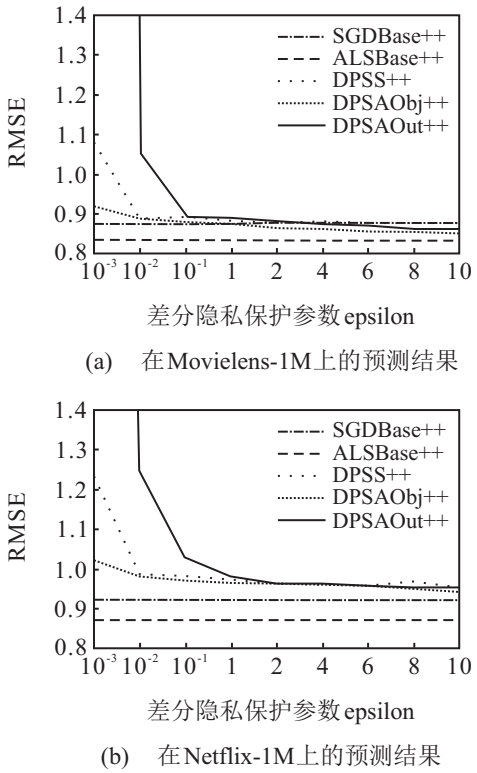
从图2可以看出,本文提出的基于目标函数扰动的SVD++隐私保护算法(DPSAObj++)中,ϵ对预测准确率影响是最小的. 特别需要指出的是,实验结果还显示,本文算法的预测准确率对ϵ并不敏感,事实上,由图2可见,对于$\epsilon \in [2, 11]$,本文提出的DPSAObj++算法得到的推荐电影集的重合度百分比取值都落在[20 %, 80 %]之间.

除了上述参数之外,本文还通过交叉验证实验,得到了其他一些算法参数的设置:矩阵分解目标函数的正则化参数$\lambda = 0.125$,学习率$\gamma = 0.001$,迭代次数$k = 20$(此时误差变化小于0.000 1);另外,由于实验结果将与文献[13]做比较,算法3中的$p_{\max}$和$q_{\max}$取值与文献[13]相同,分别取0.4和0.5.

3.2 实验结果

图3给出了本文提出的3个关于SVD++的差分隐私保护算法与各自未做任何隐私保护处理的原始算法在不同数据集上的RMSE结果.

由图3可以看出,在两个数据集上,本文提出的关于SVD++的差分隐私保护算法的RMSE结果均是可接受的,即都没有很大程度上偏离各自的无隐私保护的版本. 总体而言,3个算法在Movielen-1M数据集上的效果更好,这是因为截取的Netflix数据集的训练样本比Movielens-1M要少,且更加稀疏. 特别地,在图3(b)中,当$\epsilon < 0.01$时,输出结果扰动算法的预测准确率变差,这是因为ϵ越小,Laplace噪声越大,对分解后的两个隐含矩阵加扰再做内积,必然更加偏离真实值. 当$\epsilon > 0.01$时,图3(a)和图3(b)中的目标函数扰动算法获得了比其他算法更优的结果,这是因为SGD的每一次迭代更新与误差有关,而ALS的每一次迭代与训练的数据集有关,即ALS方法自身优于



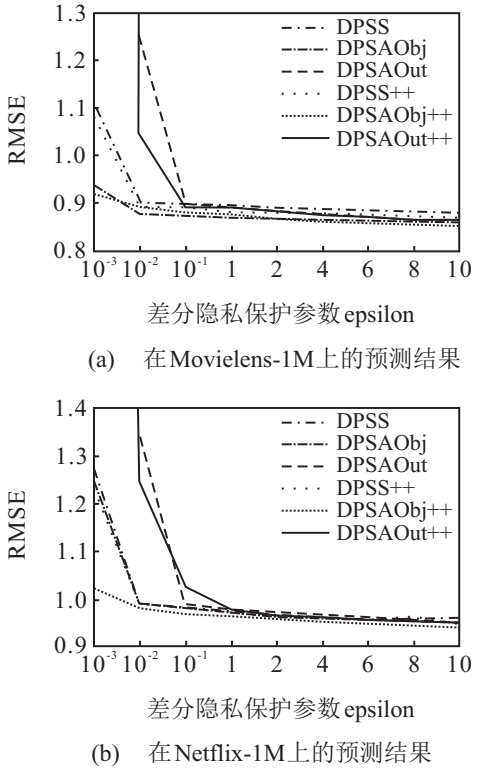
(a) 在Movielens-1M上的预测结果

(b) 在Netflix-1M上的预测结果

图 3 本文算法与无隐私保护算法的预测结果比较

SGD[15],所以做了相关差分隐私保护处理后最终的结果还是凸显优势.

为提高预测准确率,SVD++自身是在SVD基础上引入了像“用户是否参评过”这类隐式反馈信息. 图4给出了SVD++与SVD分别做3种差分隐私保护后的RMSE比较.



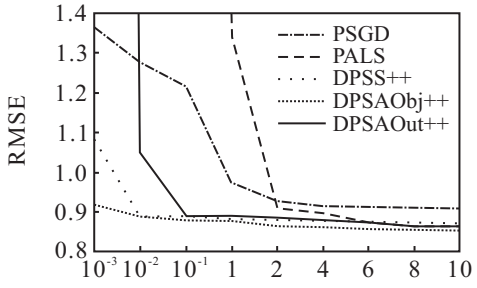
(a) 在Movielens-1M上的预测结果

(b) 在Netflix-1M上的预测结果

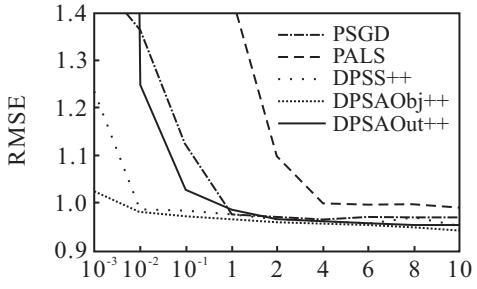
图 4 SVD++与SVD的3种差分隐私保护比较

由图4可以看出,即使是按照同样的差分隐私保护策略,SVD++的优势还是略高于SVD. 总体而言,目标函数扰动的RMSE都是最优的,特别是当$\epsilon > 0.01$时,上述趋势更为明显.

图5给出了本文的3个算法与文献[13]的算法4(SGD梯度扰动,PSGD)和算法5(ALS输出结果加扰PALS)在两个数据集上的结果比较.



(a)　在Movielens-1M上的预测结果



(b)　在Netflix-1M上的预测结果

图5　本文算法与文献[13]相关算法的比较

由图5可以看出,本文的3个算法在两个数据集上的RMSE均优于文献[13]的算法. 更加值得注意的是,当差分隐私保护参数epsilon < 1时,文献[13]的两个算法几乎不可用于做预测. 原因分析如下: 1) 文献[13]的两个算法均对原始数据做了差分隐私保护的数据预处理, 即对原始数据做了加扰处理,这样必定会给矩阵分解模型的训练带来影响. 2) 文献[13]的算法5(PALS)是对ALS输出结果加扰,自身也不如ALS目标函数扰动的算法占优势(文献[22]已证明). 因此,本文对SVD++分解进行差分隐私保护研究,与文献[13]相比,虽然都是针对矩阵分解的优化算法做加扰处理,但是本文算法一方面为保证原始数据的可用性,省去了对原始数据做预处理;二是在隐私保护参数ϵ较小的范围内,均获得了更好的预测准确率. 所以,在实际的应用中,本文算法将更具优势.

3.3　运行时间分析

为了验证本文提出的3个差分隐私保护算法的时间效率,表11给出了不做差分隐私保护和做差分隐私保护的算法在运行时间上的比较.

表11　有无DP的算法运行时间比较

算法	Movielens-1M	Netflix-1M
SGDBase++(无DP)	97	78
DPSS++(算法1)	100	78
ALSBase++(无DP)	806	640
DPSAObj++(算法2)	808	641
DPSAOut++(算法3)	809	642

从表11可以得出如下结论: 1) 本文提出的3个做差分隐私保护的算法在运行时间上与他们对应的无隐私保护的算法版本的运行时间没有显著的差异; 2) 由于ALS优化自身的运行时间比SGD优化算法要长[15],关于ALS扰动的算法2和算法3的运行时间也自然比算法1长; 3) 算法在Netflix-1M数据集比Movielens-1M数据集上耗时少,这是因为Netflix数据集的稀疏性所致.

综合本节的实验结果可知,本文针对SVD++提出的3个具有差分隐私保护能力的算法在确保预测准确率的基础上较好地保护了原始数据隐私, 特别是目标函数扰动算法(算法2),更好地平衡了隐私和推荐两个方面的效能.

4　结　论

改善服务质量是推荐系统发展的目标之一[23]. 目前协同过滤作为推荐系统的主要技术实现之一,在近年已经得到了广泛的应用. 因为协同过滤以用户对商品项目给出的历史评分为依据,而这些评价有可能反映了用户某些特殊的偏好,因此,如何为协同过滤算法引入隐私保护能力成为了当前研究者普遍关注的一个问题.

本文的工作以SVD++模型为目标,研究了基于SVD++的差分隐私保护机制. 基于SVD++的求解策略提出了3个SVD++隐私保护算法,分别是基于梯度扰动的差分隐私保护策略、基于优化目标扰动的差分隐私保护策略和基于结果扰动的隐私保护策略. 对于论文提出的所有算法,都分析了其隐私保护的理论性能,并在真实数据集上检验了其在协同过滤应用中的预测表现. 实验结果显示,本文提出的算法与无隐私保护的SVD++实现具有相近的运行效率和预测精度.

已有研究表明,预测算法的稳定性与差分隐私保护机制的设计有密切的联系,但对于SVD++算法的稳定性,目前尚未有确切的结论,因而也未能自稳定性这一起点出发建立SVD++与差分隐私保护机制两者间的联系,所以,如何建立起上述的联系,将是未来研究的重点.

参考文献(References)

[1] Ricci F, Rokach L, Shapira B. Recommender systems handbook[M]. Berlin: Springer, 2011: 1-35.

[2] Linden G, Smith B, York J. Amazon.com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.

[3] Zhou Yun-Hong, Wilkinson D, Schreiber R, et al. Large-scale parallel collaborative filtering for the netflix prize[C]. Proc of the 2008 Int Conf on Algorithmic Applications in Management. Binlin: Springer, 2008: 337-348.

[4] Narayanan A, Shmatikov V. Robust de-anonymization of large sparse datasets[C]. Proc of the 2008 IEEE Symposium on Security and Privacy. Washington: IEEE, 2008: 111-125.

[5] Dwork C. Differential privacy[C]. Proc of the 33rd Int Colloquium on Automata, Languages and Programming. Binlin: Springer, 2006: 1-12.

[6] Furey T S, Cristianini N, Duffy N, et al. Support vector machine classification and validation of cancer tissue samples using microarray expression data[J]. Bioinformatics, 2000, 16(10): 906-914.

[7] Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Foundations & Trends in Theoretical Computer Science, 2014, 9(3/4): 211-407.

[8] Mcsherry F, Mironov I. Differentially private recommender systems: Building privacy into the net[C]. Proc of the 2009 ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2009: 627-636.

[9] Hardt M, Roth A. Beating randomized response on incoherent matrices[C]. Proc of the Annual Acm Symposium on the Theory of computing. New York: ACM, 2012: 1255-1268.

[10] 鲜征征, 李启良. 差分隐私在推荐系统中的应用研究[J]. 计算机应用研究, 2016, 33(5): 1549-1553.
(Xian Z Z, Li Q L. Research on application of differential privacy in recommender system[J]. Application Research of Computers, 2016, 33(5): 1549-1553.)

[11] Hua Jing-yu, Xia Chang, Zhong Sheng. Differential private matrix factorization[J]. Proc of the 24th Int Conf on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 1763-1770.

[12] Yan Shen, Pan Shi-Ran, Zhu Wen-Tao, et al. DynaEgo: Privacy-reserving collaborative filtering recommender system based on social-Aware differential privacy[C]. Proc of the Int Conf on Information and Communications Security. Binlin: Springer, 2016: 347-357.

[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] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model[C]. Proc of the 2008 Acm Sigkdd Int Conf on Knowledg Discovery and Data Mining. New York: ACM, 2008: 426-434.

[15] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[16] Dwork C. Differential privacy: A survey of results[C]. Proc of the 5th Int Conf on Theory and Applications of Models of Computation. Binlin: Springer, 2008: 1-19.

[17] McSherry F. Privacy integrated queries: An extensible platform for privacy-preserving data analysis[C]. Proc of the 2009 ACM SIGMOD Int Conf on Management of Data(SIGMOD). New York: ACM, 2009: 19-30.

[18] Nissim K, Raskhodnikova S, Smith A. Smooth sensitivity and sampling in private data analysis[C]. Proc of the 39th Annual ACM Symposium on Theory of Computing. New York: ACM, 2007: 75-84.

[19] Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis[C]. Proc of the 3th Theory of Cryptography Conference(TCC). Binlin: Springer, 2006: 363-385.

[20] 熊平, 朱天清, 王晓峰. 差分隐私保护及其应用[J]. 计算机学报, 2014, 37(1): 101-122.
(Xiong P, Zhu T Q, Wang X F. A survey on differential privacy and applications[J]. Chinese J of Computers, 2014, 37(1): 101-122.)

[21] McSherry F. Privacy integrated queries: an extensible platform for privacy-preserving data analysis[C]. Proc of the 2009 ACM SIGMOD Int Conf on Management of Data(SIGMOD). New York: ACM, 2009: 19-30.

[22] Chaudhuri K, Monteleoni C, Sarwate A. Differentially private empirical risk minimization[J]. J of Machine Learning Research, 2011, 12(2): 1069-1109.

[23] Su Kai, Ma Liang-Li, Xiao Bin, et al. Web service QoS prediction by neighbor information combined non-negative matrix factorization[J]. J of Intelligent & Fuzzy Systems, 2016, 30(6): 3593-3604.

(责任编辑：闫 妍)