文章编号:1007-5321(2021)02-0081-08 **DOI**:10.13190/j.jbupt.2020-174

基于 BC 聚类的差分隐私保护推荐算法

王 永[1,2], 尹恩民[1], 冉 珣[2]
(1. 重庆邮电大学 计算机科学与技术学院, 重庆 400065;
2. 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: 为提高差分隐私保护下推荐算法的准确性,提出了一种考虑差分隐私保护的基于 Bhattacharyya 系数(BC)的聚类推荐算法. 以 BC 作为项目相似性度量的标准,根据 BC 相似性对项目进行 *K*-medoids 聚类,并在聚类簇中进行私有项目邻居选择. 最后,根据最近邻居集信息,对用户的评分进行预测和 Top-*n* 推荐. 提出的方案有效地克服了已有方法中存在的相似性度量依赖于共同评分的问题,提高了相似性度量的准确性,有效避免了因隐私保护而造成的最近邻居集质量下降的问题. 理论分析和实验测试的结果表明,该方法在实现隐私保护的同时还能有效保证推荐的高质量,较好地实现了隐私保护和数据效用之间的平衡,具有良好的应用潜力.

关 键 词: 协同过滤; Bhattacharyya 系数; 差分隐私保护; *K*-medoids 聚类; 推荐系统
中图分类号: TP309.2 **文献标志码**: A

Differential Privacy-Preserving Recommendation Algorithm Based on Bhattacharyya Coefficient Clustering

WANG Yong[1,2], YIN En-min[1], RAN Xun[2]
(1. College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
2. Key Laboratory of E-Commerce and Modern Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: To improve the accuracy of recommendation algorithm under differential privacy protection, a privacy preservation recommendation algorithm is proposed based on a clustering method with Bhattacharyya coefficient(BC). In the proposed algorithm, the Bhattacharyya coefficient is used as the standard of measuring item similarity. Based on the BC similarity, the items are clustered by *K*-medoids, and the private neighbors of the items are selected from the clusters. Finally, according to the selected nearest neighbor set, the user's rating is predicted and the Top-*n* recommendations are output. The proposed algorithm effectively overcomes the problem that the calculation of similarity must depend on the common rated ratings, improves the accuracy of the similarity measurement, and also avoid the problem of quality degradation of the nearest neighbor set due to privacy protection. It is shown that the proposed algorithm not only achieves privacy preservation but also guarantees the high quality of recommendation. Therefore, the proposed algorithm effectively balances the privacy preservation and the data utility, which has good application potential in the recommendation system.

Key words: collaborative filtering; Bhattacharyya coefficient; differential privacy preservation; *K*-medoids clustering; recommendation system

收稿日期: 2020-09-09
基金项目: 国家自然科学基金项目(71901045); 教育部人文社科规划项目(20YJAZH102)
作者简介: 王 永(1977—), 男, 教授, E-mail:wangyong_cqupt@163.com.

推荐系统作为解决“信息过载”问题的一种有效方法,在互联网领域发挥着不可替代的作用[1]. 与此同时,人们越来越重视推荐系统中的数据隐私保护问题[2-4]. 差分隐私是近年来较热门的一种隐私保护技术. Mcsherry 等[5]首次将差分隐私引入协同过滤推荐算法中. Zhu 等[6]针对不同的攻击场景为基于用户和基于项目的协同过滤算法分别设计了差分隐私保护方案. 当前,研究者利用推荐感知敏感度有效地降低了引入的噪声量[7],提高了数据的可用性. 首先,Yang 等[8]引入了 Johnson Lindenstrauss 变换,利用拉普拉斯机制设计了一种考虑用户隐私偏好的个性化协同过滤方案. 另一方面,差分隐私会引入大量噪声,破坏数据的可用性. 如何达到隐私性和数据可用性的平衡是应用差分隐私技术的一个难点;其次,现有的差分隐私协同过滤算法在计算相似性时往往受限于共同评分项,无法利用所有的评分数据,从而造成一定的推荐偏差. 当数据稀疏时,用户或项目可能不存在共同评分项,这将导致无法计算其相似性,在部分方案中,这种情况可能引发潜在的隐私风险. 此外,在全局范围内进行隐私邻居的搜索和选择一方面会影响算法的效率;另一方面容易受到指数机制引入噪声的影响,无法选出高质量的邻居. 为解决上述问题,提出了一种基于 Bhattacharyya 系数(BC, Bhattacharyya coefficient)聚类的差分隐私协同过滤(BCDPCF, Bhattacharyya coefficient clustering based differential privacy collaborative filtering)算法,主要贡献包括以下 3 点.

1) 利用聚类的方式降低因指数机制引入的噪声对基于项目协同过滤算法的影响,从而保证算法在低隐私预算的情况下也能有一个好的推荐结果.

2) 将 BC 应用于项目相似性的计算中,从项目品质的角度为用户进行个性化推荐,克服了已有相似性度量方法对共同评分的依赖,提高了评分数据的利用率.

3) 将 BC 和 *K*-medoids 聚类相结合,从评分概率分布的角度为项目进行聚类,解决了传统聚类中几何距离在推荐系统高维高稀疏环境下度量不准确的问题,保证了聚类的效果.

1 基于 BC 聚类的差分隐私保护推荐算法

基于邻居项目的协同过滤方法具有相似性关系稳定且推荐准确性高的特点,文献[2]中的结果表明,攻击者通过推测相关项目列表可以窥探到用户偏好的项目,因此存在用户隐私泄露的问题. 为了阻止相关项目列表的推测攻击,提高算法的性能,将 BC 作为项目 *K*-medoids 聚类的标准,提出了一种新的实施差分隐私保护的协同过滤算法. BCDPCF 算法的主要框架如图 1 所示.

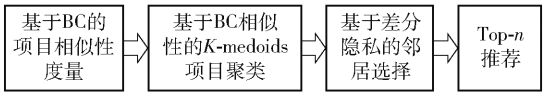

图 1 算法框架

1.1 基于 BC 的项目相似性度量

BC 被广泛应用于信号处理、模式识别等领域. 假定 2 个信号源的概率密度分布分别为 p_1 和 p_2,那么它们在离散域 X 上的 BC 定义为

$$B_{p_1,p_2}=\sum_{x\in X}\sqrt{p_1(x)p_2(x)} \tag{1}$$

将 BC 引入基于邻居项目的协同过滤算法中,并用于度量项目之间的相似性. 将所有用户已有的对项目 i 和 j 的评分视作 2 个数据源,$p_i(h)$ 和 $p_j(h)$ 分别表示项目 i 和 j 中评分值 h 的概率分布,那么基于 BC 计算项目 i 和项目 j 之间的相似性为

$$B_{p_i,p_j}=\sum_{h\in m}\sqrt{p_i(h)p_j(h)} \tag{2}$$

其中:m 为评分值的范围,$p_i(h)=\frac{\#h}{\#R_i}$,$p_j(h)=\frac{\#h}{\#R_j}$. $\#h$ 表示所有评分中评分值为 h 的评分数量,$\#R_i$ 和 $\#R_j$ 分别表示项目 i 和 j 的所有评分数量. 根据式(2)可知,采用 BC 度量项目间的相似性,取值范围为[0,1],其值越大,表明 2 个项目越相似.

1.2 基于 BC 相似性的 *K*-medoids 聚类

进行差分隐私邻居选择前,采用 *K*-medoids 算法对项目进行聚类,然后在聚类范围内采用指数机制进行隐私保护下的邻居选择,其主要工作过程如下.

1) 首先,从项目集合中随机选择 K 个项目作为初始中心集合 medoids;其次,计算每个项目与 K 个中心之间的 BC 相似性,并将项目分配给相似性最大的中心所在的簇; 再次,计算簇中每个项目与簇内其他项目之间 BC 相似性的和,有

$$S_{\kappa,i}=\sum_{j\in C_\kappa}B_{p_i,p_j} \tag{3}$$

2) 更新中心点,要求新中心点与簇内其他项目的相似性之和最大. 重复上述过程,直到中心点不再变化为止. 上述算法对应的伪代码如算法 1 所示.

算法 1　基于 BC 的 K-medoids 项目聚类

输入：聚类数 K

输出：$C=\{C_1,C_2,\cdots,C_\kappa\}$

随机选择 K 个中心点
将每个项目分配到最近的簇中
for $\kappa=1:K$ do:
　计算 $S_{\kappa,\text{medoids}[\kappa]}$
　for each item i in C_κ do:
　　计算 $S_{\kappa,i}$
　　if $S_{\kappa,i}>S_{\kappa,\text{medoids}[\kappa]}$:
　　$S_{\kappa,\text{medoids}[\kappa]}=S_{\kappa,i}$
　　medoids$[\kappa]=i$
　　end if
　end for
end for
if 中心点未改变:
　return $C=\{C_1,C_2,\cdots,C_\kappa\}$
else:
　返回步骤 3
end if

1.3　差分隐私邻居选择

在邻居选择阶段，将指数机制运用于邻居项目的选择过程中，对其实施隐私保护. 假设有目标项目 i，采用 BC 作为相似性度量的标准，则项目 j 被选为其邻居的效用函数为

$$q_{i,j}=B_{p_i,p_j} \tag{4}$$

由于 BC 相似性的取值范围为[0, 1]，所以效用函数的全局敏感度为

$$G_q=\max_{D,D'}\| q_{i,j}(D)-q_{i,j}(D')\| =1$$

通过算法 1 得到了聚类项目集 C 后，将目标项目 i 所在簇内的项目作为候选邻居. 同时，为了安全性，必须保证候选邻居数大于所需邻居数 N. 当簇内项目数少于 N 时，需计算目标项目 i 与其他簇中心的距离，将距离最小的那个簇中的项目加入候选邻居集中. 重复此过程，直到候选邻居数大于 N. 将最终得到的候选邻居集表示为 I，设计 I 中每个项目被选入最近邻居集的概率为

$$\Pr[j|j\in I]=\frac{\exp\left(\dfrac{\varepsilon B_{p_i,p_j}}{2NG_q}\right)}{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B_{p_i,p_{j'}}}{2NG_q}\right)} \tag{5}$$

依据式(5)的概率以无放回的方式从 I 中选择邻居项目，重复 N 次，得到目标项目 i 的最近邻居集 $N_s(i)$. 综合上述过程，所提出的考虑隐私保护的邻居项目选择方案如算法 2 所示.

算法 2　基于差分隐私保护的邻居项目选择

输入：目标项目 i，最近邻居数 N，聚类的项目集 C，候选邻居集 I

输出：项目 i 的最近邻居集 $N_s(i)$

初始化候选邻居集 I 为空
从 C 中找到 i 所在的簇 C_i，将 C_i 中的项目添加到 I 中
while I 中项目数 $<N$:
　在 C 中找到与 i 差异最小且不在 I 中的簇 C_x
　簇 C_x 内的项目加入 I 中
end while
for $t=1:N$ do:
　根据式(5)选出项目 j 并添加到 $N_s(i)$ 中
end for

1.4　Top-n 推荐

根据算法 2 中得到项目 i 的最近邻居集 $N_s(i)$，预测用户 u 对项目 i 的评分为

$$P_{u,i}=\frac{\sum\limits_{j\in Nb(i)}S(i,j)r_{u,j}}{\sum\limits_{j\in Nb(i)}|S(i,j)|} \tag{6}$$

其中：$r_{u,j}$ 为用户 u 对项目 j 的真实评分，$S(i,j)$ 为项目 i 和项目 j 的相似性，根据用户 u 对所有未评分项目的预测值，挑选预测分数最高的 n 个项目推荐给用户.

2　算法分析

2.1　效用分析

1）BCDPCF 算法采用 BC 度量项目之间的相似性，以评分值的概率密度分布作为计算标准，受数据稀疏性的影响小. 常用的相似性度量方法主要基于共同评分进行计算，受数据稀疏性的影响大. 以 MovieLens 数据集为例，两用户间的共同评分数仅仅约占所有评分数据的 4%，这意味着基于共同评分的相似性度量方法其信息的利用率极低，容易造成偏失. 由于 BC 相似性不受共同评分的限制，对评分数据的利用率可以达到 100%，所以计算得到的相似性更加全面和准确.

2）在进行邻居选择前对项目进行聚类. 聚类可以有效降低指数机制引入的噪声，提高算法的效用. 设数据集中项目的总数为 z，目标项目为 i，预测计算需要的邻居数为 N，高质量邻居数为 y，显然存

在 $N \leqslant y < z$. 在不聚类的条件下,当隐私保护程度较高时,数据集中每个项目被选作邻居的概率是较为均匀的. 假定采用均匀选择,那么选到 N 个高质量邻居的概率为

$$\frac{C_y^n}{C_z^n} = \frac{\dfrac{y(y-1)(y-2)\cdots(y-N+1)}{N!}}{\dfrac{z(z-1)(z-2)\cdots(z-N+1)}{N!}} = \frac{y(y-1)(y-2)\cdots(y-N+1)}{z(z-1)(z-2)\cdots(z-N+1)}$$

在考虑聚类的条件下,假设项目被聚为 k 类,且目标项目 i 的高质量邻居被分到各个簇中的概率相同,则高质量邻居分到与 i 同簇的概率为 $1/k$. 由于 i 的高质量邻居占比为 y/z,则 i 所在的簇内选到 N 个高质量邻居的概率为

$$\frac{y}{z}\underbrace{\frac{1}{k}\frac{1}{k}\cdots\frac{1}{k}}_{N} = \frac{y}{zk^N}$$

若 $$\frac{y}{zk^N} \geqslant \frac{y(y-1)(y-2)\cdots(y-N+1)}{z(z-1)(z-2)\cdots(z-N+1)}$$

则说明聚类能提高算法的效用,即

$$k \leqslant \sqrt[N]{\frac{(z-1)(z-2)\cdots(z-N+1)}{(y-1)(y-2)\cdots(y-N+1)}}$$

时,聚类能提高算法的效用. 在推荐场景中,由于 $N \leqslant y < z$,则 $(z-1)(z-2)\cdots(z-N+1)$ 远大于

$$(y-1)(y-2)\cdots(y-N+1)$$

上述条件在推荐场景中很容易满足,所以采用的聚类方式能有效提高算法的效用.

2.2 隐私性分析

定理 1 BCDPCF 算法满足 ε-差分隐私保护.

证明 数据集 D 为候选邻居集 I 所对应的评分数据集,G_q 为 BC 相似性的全局敏感度,给定邻居项目数 N 和隐私预算 ε,将邻居选择步骤视作随机算法 $\mathscr{K}$. 因为

$$\frac{\exp\left(\dfrac{\varepsilon B_{p_i,p_j}}{2NG_q}\right)}{\exp\left(\dfrac{\varepsilon B'_{p_i,p_j}}{2NG_q}\right)} = \exp\left(\frac{\varepsilon(B_{p_i,p_j}-B'_{p_i,p_j})}{2NG_q}\right) \leqslant \exp\left(\frac{\varepsilon}{2N}\right)$$

类似地,采用同样的推导方法可得

$$\exp\left(\frac{\varepsilon B'_{p_i,p_j}}{2NG_q}\right) \leqslant \exp\left(\frac{\varepsilon}{2N}\right)\exp\left(\frac{\varepsilon B_{p_i,p_j}}{2NG_q}\right)$$

根据差分隐私的定义,对每轮的邻居选择,有

$$\frac{\Pr[\mathscr{K}_{\mathrm{BC}}^{\varepsilon}(D)=j]}{\Pr[\mathscr{K}_{\mathrm{BC}}^{\varepsilon}(D')=j]} = \frac{\dfrac{\exp\left(\dfrac{\varepsilon B_{i,j}}{2NG_q}\right)}{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B_{i,j'}}{2NG_q}\right)}}{\dfrac{\exp\left(\dfrac{\varepsilon B'_{i,j}}{2NG_q}\right)}{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B'_{i,j'}}{2NG_q}\right)}} =$$

$$\left(\frac{\exp\left(\dfrac{\varepsilon B_{i,j}}{2NG_q}\right)}{\exp\left(\dfrac{\varepsilon B'_{i,j}}{2NG_q}\right)}\right)\left(\frac{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B'_{i,j'}}{2NG_q}\right)}{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B_{i,j'}}{2NG_q}\right)}\right) \leqslant$$

$$\exp\left(\frac{\varepsilon}{2N}\right)\left(\frac{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon}{2N}\right)\exp\left(\dfrac{\varepsilon B_{i,j'}}{2NG_q}\right)}{\sum\limits_{j'\in I}\exp\left(\dfrac{\varepsilon B_{i,j'}}{2NG_q}\right)}\right) \leqslant$$

$$\exp\left(\frac{\varepsilon}{N}\right)$$

因此,算法对每轮的邻居选择满足 $\dfrac{\varepsilon}{N}$-差分隐私. 在 BCDPCF 算法中需要进行 N 轮邻居的选择,根据差分隐私的组合性质可得 BCDPCF 算法满足 ε-差分隐私,证毕.

3 实验结果与分析

采用公开的数据集 MovieLens(ml-latest-small 和 MovieLens 1M)和 Yahoo Music 作为实验的数据集. 数据集 ml-latest-small 中被评次数超过 20 次的项目形成子集 M,将数据集以 8∶2的比例分为训练集和测试集. 采用平均绝对误差(MAE, mean absolute error)和根均方误差(RMSE, root mean square error)对 BCDPCF 算法进行评测.

为减小差分隐私的随机性对实验结果的影响,采用 10 次实验的平均值作为 BCDPCF 算法的实验结果. 为确定聚类的个数,将项目的聚类数作为变量,其值依次设置为 3,4,5,…,15. 在 M 数据集中得到的结果如图 2 所示. 可以看出,当聚类数为 7 时,BCDPCF 算法的 MAE 值最小;当聚类数为 11 时,RMSE 的值最低,在综合考虑簇中数量差异的情况下,将聚类数设置为 7. 在 Yahoo Music 数据集和 MovieLens 1M 数据集中进行同样的实验,根据实验结果设置的聚类数分别为 12、30. 将 BCDPCF 算法中涉及的其他参数设置为 $\varepsilon = 1$,M 和 Yahoo Music 数据集中的 $N = 60$,MovieLens 1M 中的 $N = 100$. 为了进一步说明 BCDPCF 算法性能,选取如下 3 个算法与之进行对比.

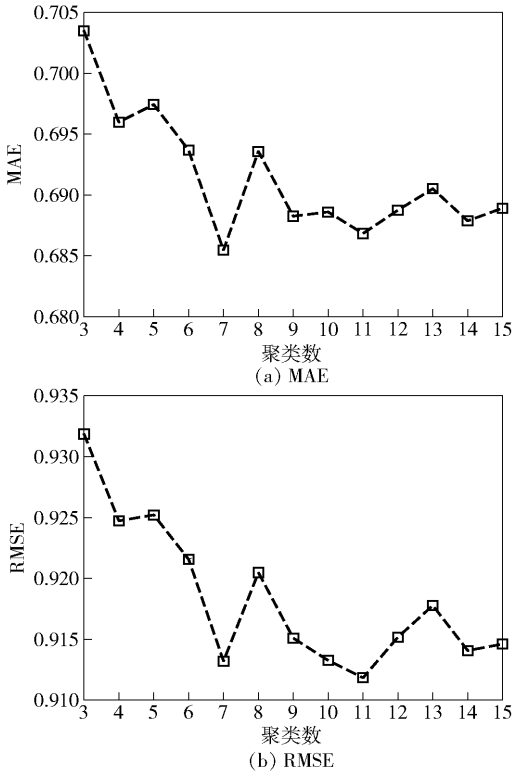

图 2　不同聚类数下 BCDPCF 算法的 MAE 和 RMSE

1）PNCF（private neighbor collabora tive filtering）算法[6]. 采用推荐感知敏感度和截断相似度，同时运用指数机制和拉普拉斯机制对基于项目的协同过滤算法实施隐私保护.

2）IBCF（item based collaborative filtering）算法[9]. IBCF 算法是经典的基于项目的协同过滤算法，相似性计算方式为修正余弦相似性，且未对推荐过程实施任何隐私保护.

3）PSGD（private stochastic gradient descent）算法[10]. 在矩阵分解的每轮梯度下降过程中加入拉普拉斯噪声，以实现差分隐私保护.

为了说明聚类操作和差分隐私对算法性能的影响，以 BCDPCF 算法为基础，删除其中的步骤 3，即去除隐私保护操作，得到未考虑隐私保护的算法. 这是一个 *K*-medoids 聚类与 BC 相结合的协同过滤（KBCF，*K*-medoids Bhattacharyya collaborative filtering）算法. 同样地，以 BCDPCF 算法为基础，删除其中的步骤 2，即去除聚类操作，得到一个考虑隐私保护的基于 BC 的协同过滤（PBCF，private Bhattacharyya collaborative filtering）算法.

3.1　差分隐私对预测准确性的影响

为探究差分隐私措施的引入对预测准确性的影响，选取 IBCF 算法和 KBCF 算法与 BCDPCF 算法进行对比实验. 在 *N* 取不同值的条件下，得到各算法在 3 个数据集中的 MAE 和 RMSE，如表 1 所示.

表 1　测试差分隐私对预测准确性影响的结果

数据集	算法名称	指标	邻居数				
			20	40	60	80	100
M	IBCF	MAE	0.811	0.764	0.736	0.715	0.700
		RMSE	1.090	1.023	0.983	0.955	0.935
	KBCF	MAE	0.688	0.671	0.665	0.660	0.658
		RMSE	0.927	0.895	0.883	0.874	0.871
	BCDPCF	MAE	0.732	0.703	0.694	0.686	0.679
		RMSE	0.986	0.940	0.921	0.910	0.897
Yahoo Music	IBCF	MAE	1.029	1.044	1.037	1.033	1.017
		RMSE	1.558	1.542	1.517	1.494	1.460
	KBCF	MAE	1.016	1.003	0.987	0.977	0.970
		RMSE	1.498	1.445	1.402	1.372	1.348
	BCDPCF	MAE	1.035	1.014	1.008	0.993	0.993
		RMSE	1.528	1.468	1.440	1.403	1.392
MovieLens 1M	IBCF	MAE	0.997	0.985	0.975	0.955	0.922
		RMSE	1.273	1.257	1.246	1.227	1.194
	KBCF	MAE	0.777	0.754	0.742	0.735	0.731
		RMSE	1.034	0.989	0.966	0.953	0.944
	BCDPCF	MAE	0.803	0.777	0.765	0.760	0.751
		RMSE	1.074	1.025	1.001	0.989	0.975

从表1可以看出,KBCF算法的性能远远优于IBCF算法. BCDPCF算法的测试结果介于KBCF和IBCF算法之间,2种方法采用的结构和过程类似,说明差分隐私措施的引入使所选择邻居项目的质量下降,因此BCDPCF算法的MAE和RMSE值大于KBCF算法. 另一方面,由于在BCDPCF算法中选取了较高质量的相似度量方法,通过聚类方式有效避免了选到质量差的项目邻居,所以BCDPCF算法的预测准确性虽有下降,但下降的幅度较小. 由于M和Yahoo Music数据集的稀疏性不同,其结果表明BCDPCF算法具有更稳定的性能. 在MovieLens 1M上的测试结果同样验证了BCDPCF算法所采用的BC相似性和聚类策略能够有效保证预测的准确性,为实施隐私保护,保证数据的质量奠定了良好的基础.

3.2 不同隐私保护算法的对比

将PNCF,PSGD, PBCF和BCDPCF算法在3个数据集中进行测试与对比,得到的MAE和RMSE如表2所示. 可见,随着邻居数的增加,对应的MAE和RMSE逐渐下降. 此外,由于PSGD并不是依据最近邻居集进行预测的,所以其测试结果没有变化. 从整体效果看,BCDPCF算法明显优于其他方案,而PSGD算法的预测性能最差. PNCF,PBCF和BCDPCF算法属于同类型, BCDPCF算法相比于PNCF和PBCF算法,在MAE和RMSE上均有较高的提升,这主要是因为BCDPCF算法能够更好地限制指数机制所带来的随机性. 相较于PNCF和PBCF算法在全局范围内进行私有邻居选择,BCDPCF首先对相似项目进行聚类;然后在聚类的簇范围内进行私有邻居选择,有效减少了私有邻居选择中的偶然性,因此提高了算法的性能.

此外,BCDPCF算法与PBCF算法相比在M, Yahoo Music和MovieLens 1M三个数据集上用每个参数下提升的均值计算,MAE和RMSE分别提升了

表2 各隐私保护算法的MAE和RMSE对比

数据集	算法名称	指标	邻居数				
			20	40	60	80	100
M	PNCF	MAE	0.801	0.774	0.760	0.750	0.745
		RMSE	1.066	1.026	1.003	0.986	0.977
	PBCF	MAE	0.808	0.774	0.758	0.748	0.740
		RMSE	1.076	1.025	1.002	0.985	0.969
	PSGD	MAE	0.848	0.848	0.848	0.848	0.848
		RMSE	1.096	1.096	1.096	1.096	1.096
	BCDPCF	MAE	0.732	0.703	0.694	0.686	0.679
		RMSE	0.986	0.940	0.921	0.910	0.897
Yahoo Music	PNCF	MAE	1.070	1.060	1.052	1.039	1.034
		RMSE	1.512	1.495	1.479	1.450	1.429
	PBCF	MAE	1.149	1.141	1.130	1.115	1.105
		RMSE	1.667	1.620	1.577	1.535	1.502
	PSGD	MAE	1.195	1.195	1.195	1.195	1.195
		RMSE	1.477	1.477	1.477	1.477	1.477
	BCDPCF	MAE	1.035	1.014	1.008	0.993	0.993
		RMSE	1.528	1.468	1.440	1.403	1.392
MovieLens 1M	PNCF	MAE	0.960	0.929	0.911	0.896	0.887
		RMSE	1.247	1.204	1.179	1.160	1.148
	PBCF	MAE	0.991	0.943	0.921	0.903	0.889
		RMSE	1.295	1.220	1.185	1.156	1.137
	PSGD	MAE	0.782	0.782	0.782	0.782	0.782
		RMSE	0.997	0.997	0.997	0.997	0.997
	BCDPCF	MAE	0.803	0.777	0.765	0.760	0.751
		RMSE	1.074	1.025	1.001	0.989	0.974

8.70%/7.96%、10.60%/8.47%、16.97%/15.47%. 由于两者在相似性度量方法、隐私保护手段和预测方法上完全相同，唯一的差距在于是否对相似项目进行聚类，所以两者在预测性能上的差异充分验证了聚类对 BCDPCF 算法的提升作用.

3.3　隐私预算对实验结果的影响

在不同隐私预算的情况下，算法的性能在区间[0,1]上以 0.1 为间隔依次设置隐私预算 ε 的值，并计算 PSGD，PNCF，PBCF 和 BCDPCF 算法下的 MAE 和 RMSE，结果如图 3 所示.

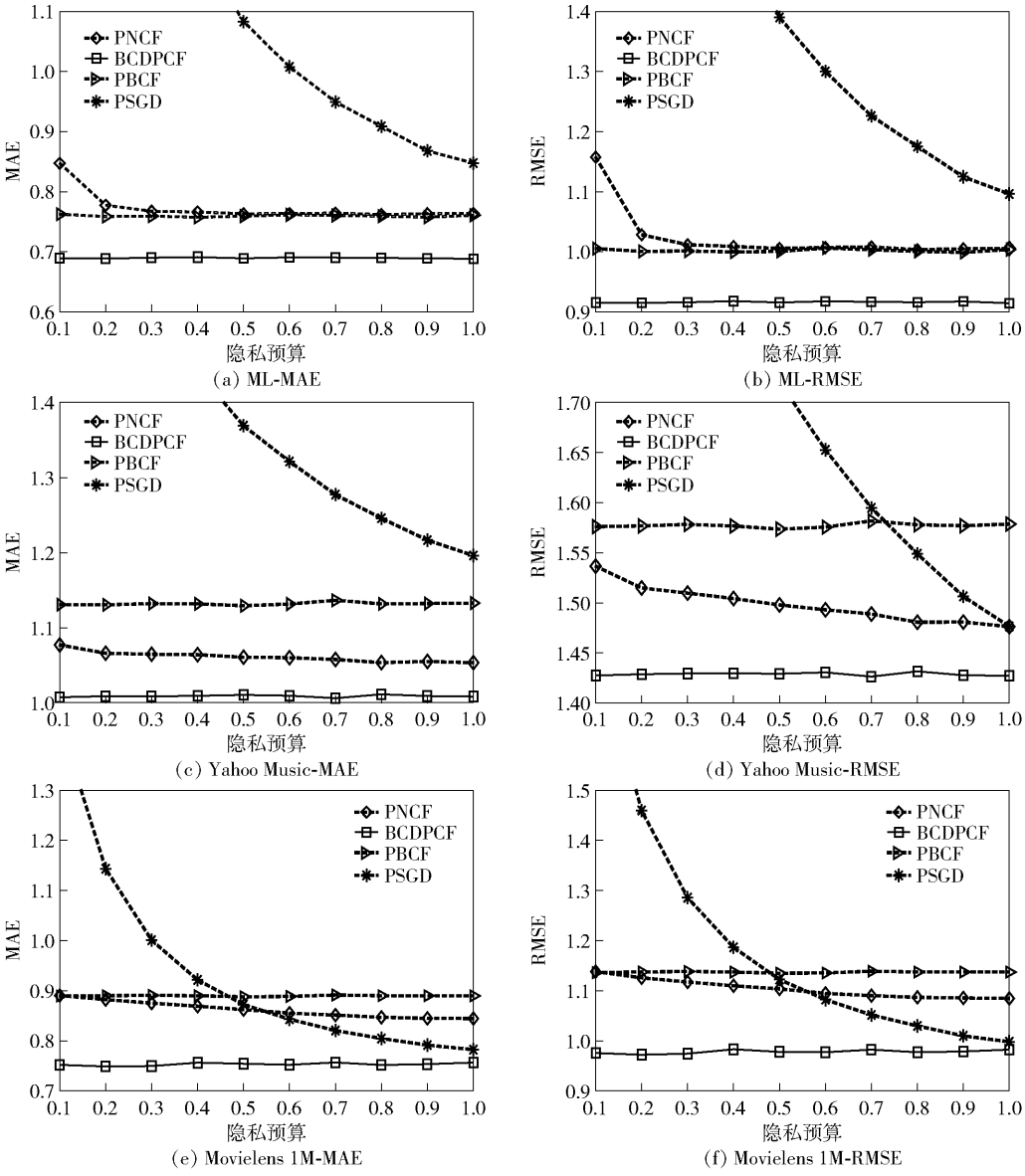

图 3　不同数据集中隐私预算值对算法性能的影响

可以看出，随着 ε 的增大，PSGD 和 PNCF 算法的 MAE 和 RMSE 逐渐变小，这是由差分隐私的性质决定的. 随着 ε 的增大，数据的可用性越高，预测的结果越好. 随着 ε 的增大，BCDPCF 和 PBCF 算法的 MAE 和 RMSE 几乎没有明显变化. 这是因为这 2 个算法没有在数据中直接引入拉普拉斯噪声，而是通过指数机制引入邻居选择的随机性. 同时，作为指数机制效用函数的 BC 相似性能够充分利用评分数据信息，因此所选出的邻居虽然不同，但是其效用却能够在预测值的计算中按照权重被合理利用，从而获得相对稳定的预测结果. 从图 3 还可以看出，当 ε 值较小时，PSGD 算法为了达到隐私保护的效果，引入的拉普拉斯噪声值较大，从而破坏了数据的可用性. 由于 BCDPCF 算法对隐私预算值不敏感，所以

在隐私预算较小时,也能获得良好的预测结果.

此外,由于 Yahoo Music 和 MovieLens 1M 数据集比 M 数据集更稀疏,相对于其他算法,PBCF 算法的预测性能受稀疏性的影响更大. 总体来说,BCD-PCF 算法受稀疏性和隐私预算变化的影响都很小,具有稳定的预测性能,且优于其他算法.

4 结束语

针对推荐系统中的相关项列表推断攻击问题,提出了一种考虑差分隐私保护的推荐算法. 利用 BC 计算项目间的相似性,然后以此相似性从概率分布的角度对项目进行聚类. 由于基于 BC 的相似性能够有效弥补现有相似性计算依赖于共同评分项的不足,能够更充分地利用评分数据,为提高算法的效用奠定了良好的基础. 同时,因为在聚类结果中应用指数机制进行差分隐私邻居选择,所以能够有效保证所选最近邻居集合的质量,从而保证了预测准确性. 在 3 个不同稀疏程度数据集上的测试结果均表明,新方案能更好地平衡隐私性和推荐性能,具有很好的应用潜力. 未来的工作中,可以考虑将标签信息、文本评论等多维信息结合到一起,建立更为综合的考虑隐私保护的高质量推荐模型.

参考文献:

[1] 孟祥武, 纪威宇, 张玉洁. 大数据环境下的推荐系统[J]. 北京邮电大学学报, 2015, 38(2): 1-15.
Meng Xiangwu, Ji Weiyu, Zhang Yujie. A survey of recommendation systems in big data [J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(2): 1-15.

[2] Calandrino J A, Kilzer A, Narayanan A, et al. "You Might Also Like:" privacy risks of collaborative filtering [C]// IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2011: 231-246.

[3] Wadhwa S, Agrawal S, Chaudhari H, et al. Data poisoning attacks against differentially private recommender systems[C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 1617-1620.

[4] Ermis B, Cemgil A T. Data sharing via differentially private coupled matrix factorization[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2020, 14(3): 1-27.

[5] Mcsherry F, Mironov I. Differentially private recommender systems: building privacy into the netflix prize contenders[C]// Knowledge Discovery and Data Mining. New York: ACM, 2009: 627-636.

[6] Zhu X, Sun Y. Differential privacy for collaborati-ve filtering recommender algorithm[C]// International Workshop on Security. New York: ACM, 2016: 9-16.

[7] Zhu T, Ren Y, Zhou W, et al. An effective privacy preserving algorithm for neighborhood-based collaborative filtering[J]. Future Generation Computer Systems, 2014, 36: 142-155.

[8] Yang M, Zhu T, Xiang Y, et al. Personalized privacy preserving collaborative filtering[C]// International Conference on Green, Pervasive, and Cloud Computing. Switzerland: Springer, 2017: 371-385.

[9] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. New York: ACM, 2001: 285-295.

[10] Friedman A, Berkovsky S, Kaafar M A, et al. A differential privacy framework for matrix factorizeation recommender systems[J]. User Modeling and User-Adapted Interaction, 2016, 26(5): 425-458.