基于社区划分的社交推荐隐私保护方法

汪玉洁[1],刘 涛[1],包象琳[1],潘正高[2]
1. 安徽工程大学 计算机与信息学院,安徽 芜湖 241000
2. 宿州学院 信息工程学院,安徽 宿州 234000

摘 要:目的 社交推荐是在传统推荐的基础上引入用户的社交信息以更好地生成推荐结果。由于社交推荐不仅涉及用户本身的信息,还涉及用户的社交关系信息,因此对用户的隐私保护变得更加重要。然而,目前的社交推荐方法大多只注重提高推荐准确性,而忽视了对用户个人信息隐私保护的问题。因此针对社交推荐中用户的评分数据和社交关系数据的隐私保护问题,提出了一种基于社区划分的社交推荐隐私保护方法(SRCD)。**方法** 首先,考虑评分值的分布范围对用户相似度的影响,并结合用户之间的社交关系,来给社交网络中的用户划分社区,并计算每个社区中用户对所看过项目的评分的均值;然后,根据社区划分的结果,寻找与目标用户所在社区相似的其他社区。从而可以构造出一个社区-项目评分均值矩阵。并且针对实际场景中评分均值矩阵稀疏的情况,采用了中位数填补矩阵的缺失元素。最后,用矩阵分解的结果来预测用户对项目的评分,从而评估算法的性能。**结果** 通过仿真实验验证,所提方法相比于现有的社交推荐算法不仅在隐私保护方面提供了保障,而且在推荐准确度方面具有相近的预测准确率。**结论** 提出的方法不仅在一定程度上保护了用户的隐私信息,还为用户提供了令人满意的推荐结果。

关键词:隐私保护;社交推荐;社区划分;分组聚合;矩阵分解
中图分类号:TH122 **文献标识码:**A doi:10.16055/j.issn.1672-058X.2024.0006.004

A Privacy Protection Method for Social Recommendation Based on Community Division

WANG Yujie[1], LIU Tao[1], BAO Xianglin[1], PAN Zhenggao[2]
1. School of Computer and Information, Anhui University of Engineering, Anhui Wuhu 241000, China
2. School of Information Engineering, Suzhou University, Anhui Suzhou 234000, China

Abstract: Objective Social recommendation introduces users' social information based on traditional recommendation to generate better recommendation results. As social recommendation involves not only the user's information but also the user's social relationship information, privacy protection for users becomes more important. However, most current social recommendation methods focus only on improving recommendation accuracy and overlook the issue of privacy protection for users' personal information. Therefore, a privacy protection method for the social recommendation based on community division (SRCD) was proposed to address the privacy protection issues of users' rating data and social relationship data in the social recommendation. **Methods** First, considering the impact of rating value distribution on user similarity and combined with the social relationships between users, communities in the social network were divided, and the average ratings of users for items they have viewed in each community were calculated. Then, based on the results of community

收稿日期:2023-07-31 **修回日期:**2023-09-18 **文章编号:**1672-058X(2024)06-0030-09
基金项目:安徽省自然科学基金(2108085QF264);安徽工程大学-鸠江区产业协同创新专项基金(2021CYXTB4);安徽工程大学科研基金(XJKY2020120);安徽省自然科学基金(2208085QD106).
作者简介:汪玉洁(1998—),女,安徽六安人,硕士研究生,从事计算机网络和信息安全研究.
通讯作者:刘涛(1973—),女,安徽六安人,教授,从事计算机网络和信息安全研究. Email:liutao@ahpu.edu.cm.
引用格式:汪玉洁,刘涛,包象琳,等. 基于社区划分的社交推荐隐私保护方法[J]. 重庆工商大学学报(自然科学版),2024,41(6):30-38.
WANG Yujie, LIU Tao, BAO Xianglin, et al. A privacy protection method for social recommendation based on community division[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, 41(6): 30-38.

division, similar communities to the community where the target users belong were identified. This allowed for the construction of a community-item rating average matrix. Additionally, for sparse rating average matrices in practical scenarios, median imputation was used to fill in missing elements in the matrix. Finally, matrix decomposition results were used to predict user ratings for items, thus evaluating the algorithm' s performance. **Results** Through simulation experiments, it was verified that the proposed method not only provides a guarantee for privacy protection but also has a similar prediction accuracy in recommendation accuracy compared with existing social recommendation algorithms. **Conclusion** The proposed method not only protects users' privacy information to a certain extent but also provides satisfactory recommendation results for users.

Keywords: privacy protection; social recommendation; community division; grouping aggregation; matrix decomposition

1 引　言

随着信息技术的飞速发展,数据量日益增加,在海量数据面前人们越来越觉得束手无策。推荐系统[1]的出现有助于人们决策,比如电影推荐[2]、饮食推荐等[3]。但是,为了实现准确的推荐,推荐系统需要收集大量的用户信息和交互数据进行分析。而这些信息和数据涉及大量隐私内容,给用户带来了隐私泄露的风险[4]。随着在线社交网络的发展,人们越来越关注以社交关系为核心的社交网络,并且人们开始萌发利用社交关系来生成更加准确的推荐的想法。社交推荐[5]是在传统推荐系统上的一种改进,在推荐过程中将用户的社交信息作为重要因素引入进去,从而提高推荐的准确性。虽然引入社交信息可以提高推荐的准确性,但同时会增加用户隐私泄露的风险。

如果社交推荐存在隐私风险会导致用户对其产生不信任,进而不愿意向系统提供自己的数据,最终影响推荐的质量。因此,为了提高推荐系统的性能,社交推荐系统必须尽全力提供强有力的隐私安全保证。更安全的隐私保护可以减轻用户对于分享数据的顾虑,从而让用户更愿意向系统提供真实数据。因此,为了促进推荐系统的健康发展,研究社交推荐中用户个人隐私的保护是非常必要的。

对数据进行匿名化处理是保护隐私的一种简单而直接的方法,其目的是防止数据使用方获知数据的具体来源,以此来保护用户隐私。通常采用假名化和分组聚合两种方法来实现匿名化[6]。假名化方法的核心思想是在提交数据时不提供个人属性信息。例如,文献[7]提供了一种"隐藏在人群中"的推荐服务模式,通过使用带有公共组身份的 Web 服务,用户能够在人群中隐藏自己,在获得一定程度的匿名性的同时,仍然可以获得个性化推荐。而分组聚合方法的核心思想是用集体数据代替个人数据,简单来说是在收集用户数据时将用户分组并以组为单位收集信息。通过将个人信息与分组的集体信息混淆,使得个人信息不被暴露出来,这种方法能够有效降低个别用户被识别的可能性,从而保护用户的隐私。在本文中,选择分组聚合的匿名化方法来保护用户数据的隐私。鲜英等[8]提出一种敏感属性匿名化方法,该方法首先分组敏感属性值,然后再进行匿名化处理,但未考虑动态信息的处理。王海艳等[9]提出一种应用于群组推荐的隐私保护模型。该模型首先收集群组里各用户的数据,然后通过随机扰动前 k 个用户来保护隐私。彭丽寻等[10]提出了一种基于 k 匿名的隐私资源推荐算法,首先将用户的隐私属性进行泛化,然后允许同一等价类中的用户可以相互转发数据来保护用户的隐私,但并未考虑多用户同时转发时的资源问题。Yi 等[11]利用联邦学习框架,首先将用户分组,然后将梯度信息按组收集,使得服务器无法区分用户与梯度之间的对应关系来保护用户隐私,其不足之处在于以集中的方式收集所有的数据可能会导致巨大的费用。Bassem 等[12]提出了一种协作群感知模型,在该模型中,对等方相互协作混淆其轨迹信息,接着通过协作推荐机制和行程分割算法,在不牺牲系统的推荐性能的同时还保护了轨迹信息。Li 等[13]提出了一种基于差分隐私的安全聚合联邦推荐框架,该框架不仅可以保证数据的隐私和模型的训练效率,还提升了集中式训练性能。Luo 等[14]通过聚类方法实现了推荐过程中的隐私保护。首先采用同态加密对用户数据进行保护,然后采用聚类技术对数据进行分区后再推荐。Sahoo 等[15]提出一种用于推荐的扰动 K-mode 隐私保护方法。在该方法中,用户评分是模糊的,然后确保分组在同一聚类中的用户相似,实验结果表明该方法的推荐效果优于其他 K-means 和 K-mode 聚类方法。

上述工作整合了匿名化方法、同态加密、联邦学习框架等隐私保护技术,为传统推荐系统的隐私保护做出了巨大贡献。然而,目前在社交推荐中还没有一个行之有效的方法能够兼顾用户评分数据的隐私性和可用性。无论是用户的评分数据还是用户的社交关系数据,都存在着泄露隐私的风险。另外,现有的社区划分算法如 Louvain 算法[16]、Fast Unfolding 算法[17]只考虑了社交网络结构,并没有考虑社交关系的影响。因此,为了在社交推荐中给用户提供满意的推荐结果的同时

保护用户的个人隐私,本文在现有研究成果的基础上,提出了一种基于社区划分的社交推荐隐私保护方法。该方法采用了匿名化中的分组聚合思想来保护用户的隐私。具体步骤如下:首先,通过同时考虑用户之间的社交关系和用户相似度将社交网络中的用户划分社区;接着,寻找与目标用户所在社区相似的其他社区,并使用这些社区的数据代替用户的个人数据进行推荐。这样,用户既能够获得满意的推荐结果,又能够隐藏自己的身份。主要优势如下:

(1) 推荐准确性。首先通过综合考虑社交关系和用户评分给用户划分社区,使得相似的用户被划分到同一个社区;然后寻找与目标用户所在社区相似的其他社区,利用这些社区数据进行推荐,给用户提供较好的推荐效果。

(2) 数据保护。用社区数据集合代替个人数据进行推荐,起到隐匿身份的目的。并且可以通过调整社区里的用户数目,实现不同程度的隐私保护。

(3) 实验验证。在公开数据集上,对SRCD算法进行实验分析并与其他算法对比,实验结果表明SRCD算法可以在推荐精确度损失较小的情况下同时保护用户的隐私。

2 问题定义

主要说明社交推荐中的隐私保护问题模型背景和要解决的问题。将社交关系和用户偏好,分别构建成一个社交图 $G_s=(U,E_S)$ 和一个偏好图 $G_P=(U,I,E_P,S)$,其中 $U=<u_1,u_2,\cdots,u_n>$ 表示用户集合,社交边 $(u,v)\in E_s$ 代表两个用户 $u,v\in U$ 之间的社交联系(例如朋友关系);$I=<i_1,i_2,\cdots,i_n>$ 表示项目集合,权重边 $(u,i)\in E_p$ 表示用户 $u\in U$ 对项目 $i\in I$ 进行了评分,$S=<s_1,s_2,\cdots,s_n>$ 表示用户对项目的评分值。如图1所示,这两个图构成了社交推荐的输入。

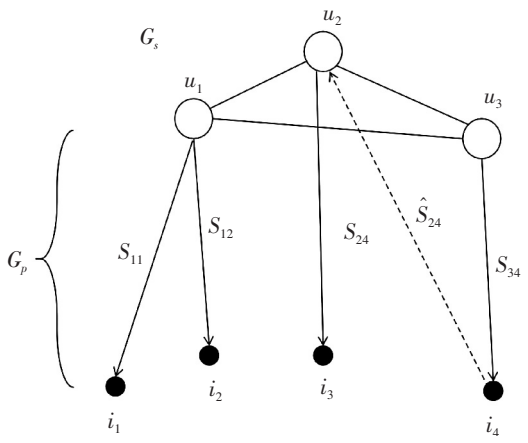


图1 由社交图和偏好图构成的图模型

Fig. 1 The graph model of social graph and preference graph

社交推荐利用社交关系来生成更加准确的推荐结果。然而,在进行社交推荐时,用户之间可能需要交换彼此的信息,这可能会导致潜在的恶意用户窃取其他用户的个人信息。因此,需要采取措施来避免用户之间的信息泄露,防止恶意用户窃取其他用户的隐私信息。

另外,在现实生活中,用户可能只给过少数物品评分,用来进行矩阵分解的用户-项目评分矩阵过于稀疏,分解出来的数据可信度较低。通常的做法是用缺失元素所在列的均值进行填补,但是由于均值填充会极大扭曲其他用户-项目的评分,并且可能推断出用户的偏好。因此采用中位数来填充。

3 SRCD算法

为了在社交推荐中更好地实现对用户个人信息的保护,提高数据的安全性和有效性,提出了一种基于社区划分的社交推荐隐私保护方法,将个人数据隐藏在社区数据集合中,用社区数据代替个人数据进行推荐,社区数据集合可以公开发布,不会泄露用户个人隐私,同时仍然可以用于给用户提供推荐服务。总体算法流程如图2所示。

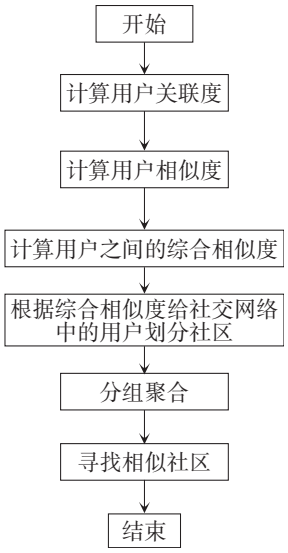


图2 SRCD算法流程图

Fig. 2 SRCD algorithm flow chart

SRCD算法主要包括3个步骤:

(1) 通过综合考虑社交关系和用户相似度来划分社交网络中的用户,不同社区间的用户可重叠;

(2) 计算每个社区的项目评分均值;

(3) 寻找 n 个与目标用户所在社区最相似的其他社区。

3.1 划分社区

步骤1 通过综合考虑社交关系和用户相似度来计算用户间的综合相似度,并利用综合相似度将社交网络中的用户进行社区划分,如算法1所述。

算法 1:划分社区

Input: 用户列表 *users*,项目列表 *items*,用户-项目评分字典 *user_scorings*,社区里的用户数 k

Output: 社区列表 *communities*

(1) 初始化空列表 *communities* 用于存放划分好的社区
(2) 初始化空字典 *user_similarities* 用于记录每对用户之间的综合相似度
(3) for each *user*1 in *users*:
(4) for each *user*2 in *users*:
(5) if *user*1 != *user*2:
(6) 利用式(1)计算关联度 *sim*(*user*1,*user*2)
(7) if *user*1,*user*2 没有相似项目:
(8) *user_similarity*=*sim*(*user*1,*user*2)
(9) else:
(10) 利用式(2)计算用户相似度 *sim_i*(*user*1,*user*2)
(11) *user_similarity*=*sim*(*user*1,*user*2)+ *sim_i*(*user*1,*user*2)
(12) *similarities*. *append*(*user*1,*user*2, *user_similarity*)
(13) *similarity*. *sort*(*reverse*=*True*,*key*=*lambdax*:*x*[2])
(14) for i=1 to k do
(15) *community*=*find_user_by_index*(*similarity*[i][2])
(16) *communities*. *append*(*community*)
(17) return *communities*

为了保护用户的隐私,采用社区数据集合代替个人数据进行推荐。算法 1 首先遍历每对用户节点:如果这两个用户没有看过相同的项目,则仅考虑他们之间的关联度作为他们之间的综合相似度;如果这两个用户看过相同的项目,则考虑他们之间的关联度和用户相似度作为他们之间的综合相似度。计算出每对用户之间的综合相似度后,找到与目标用户综合相似度最大的前(k-1)个用户,将目标用户与这(k-1)个用户划分在同一个社区。

具体做法如下:首先,根据式(1)计算社交网络中用户 u,v 之间的关联度。

$$sim(u,v)=\begin{cases}\dfrac{|t_{u,v}|}{|t_u+t_v-2|}, & \text{节点 } u,v \text{ 相连}\\ \dfrac{|t_{u,v}|}{|t_u+t_v|}, & \text{节点 } u,v \text{ 不相连}\end{cases} \tag{1}$$

其中,$t_{u,v}$ 表示用户 u 与用户 v 的共同邻居;t_u 为用户 u 的邻居。

Example 1　如图 3 所示,每个节点代表一个用户。假设用户 1 是目标用户,用户 1 和用户 2 之间的关联度为 $sim(1,2)=\dfrac{2}{3+3-2}=\dfrac{1}{2}$;用户 1 和用户 5 之间的关联度为 $sim(1,5)=\dfrac{1}{3+3}=\dfrac{1}{6}$。依次计算每个用户之间的关联度。

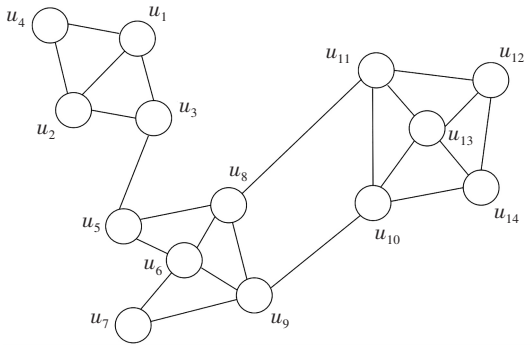


图 3　社交图模型

Fig. 3　Social graph model

然后,计算用户相似度。通常采用欧几里得距离[18]、余弦相似度[19]、皮尔逊相关系数[20]等方法来计算用户相似度。一般来说,两个用户看的相同项目数越多,这两个用户就越相似。但是考虑以下一种情况:虽然两个用户看过同一个项目,但是对这个项目的评分不同。

Example 2　假设用户 1 看了项目 1 并且评分为 1,用户 2 看了项目 1 并且评分为 5,那么项目 1 就不能纳入用户 1 与用户 2 的用户相似度计算。

因此,本文考虑评分的分布范围对用户相似度的影响来计算用户相似度,如式(2)所示:

$$sim\_i(u,v)=\frac{|d_{uv}|}{|d_u\cap d_v|} \tag{2}$$

其中,d_{uv} 表示用户 u 与用户 v 的相似项目;d_u 为用户 u 看过的项目。

Example 3　假设用户 1 看过的项目是项目 1、2、3、5、11,用户 2 看过的项目是项目 1、2、7、11。那么用户 1 和用户 2 看过的共同项目是项目 1、2、11。用户 1 和用户 2 的给项目的评分情况如表 1 所示。

表 1　用户 1 和用户 2 的项目评分数据

Table 1　Item score data for user 1 and user 2

用　户	项　目					
	1	2	3	5	7	11
1	1	1	3	2		4
2	2	5			3	5

用户 1 和用户 2 给项目 1 的评分在[1,2]区间,给项目 11 的评分在[3,5]区间,而给项目 2 的评分不在[1,2]区间也不在[3,5]区间,所以用户 1 和用户 2 的相似项目数目是 2 个,于是用户 1 与用户 2 之间的项目相似度是 $sim\_i(u,v)=\dfrac{2}{5+4-3}=\dfrac{1}{3}$。

最后,计算用户 1 与用户 2 间的综合相似度,他们之间的综合相似度为 $\dfrac{1}{2}+\dfrac{1}{3}=\dfrac{5}{6}$。

按照以上方法,计算每两个用户间的综合相似度。

通过比较目标用户与其他用户的综合相似度,来寻找(k-1)个与目标用户最相似的用户,然后将这 k 个用户划分在一个社区。

Example 4 假设给用户 1 进行推荐,并且设置社区划分的参数 k 为 5,即每个社区包含 5 个用户。为确保同一社区里的用户之间相似性最大,首先让用户 1 找到与其他综合相似度最大的一个用户,假设是用户 3。然后,将用户 1 和用户 3 划分在一个社区。继续寻找综合相似度与当前社区中所有用户之间相加值最大的第二个用户。即比较用户 1 与第二个用户的综合相似度加上用户 3 与第二个用户的综合相似度,并选择相加值最大的那个用户。按照相同的方法,逐步寻找。最终,将这五个用户划分到同一个社区。通过这样的方式,能够确保同一社区里的用户相似。

时间复杂度:算法 1 的时间复杂度为 $O(m^2)$。m 个用户节点,1-12 行计算用户之间的综合相似度,运行时间为 $O(m^2)$;14-16 行是给目标用户寻找(k-1)个最相似的用户,时间复杂度为 $O(mk)$。k 远远小于 m,所以算法 1 的时间复杂度为 $O(m^2)$。

3.2 分组聚合

步骤 2 计算每个社区里的项目评分均值,如算法 2 所述。

算法 2:分组聚合

Input:社区 *communities*,用户-项目评分字典 *user_scorings*

Output: 社区对项目的评分均值字典 *communities_avgs*

(1) 初始化空字典 *communities_avgs* 用于存放每个社区对项目的评分均值

(2) for each *community* in *communities*:

(3) 初始化空字典 *community_items* 用于存放该社区的用户看过的项目和评分

(4) for each *user* in *community*:

(5) for each *item*,*score in user_scorings*[*user*]:

(6) *community_item* = *user_scorings*[*user*]

(7) *community_items*. *append*(*community_item*)

(8) for each *item*,*scores* in *community_items*:

(9) 利用式(3)计算社区 *community* 对项目 *item* 的评分均值 *avg*

(10) *community_avgs*[*item*] = *avg*

(11) *communities*. *append*(*community_avgs*)

(12) return *communities_avgs*[*item*]

为了计算社区中每个项目的评分平均分,算法 2 首先遍历社区,找到该社区中所有用户看过的项目以及对应的评分;然后对该社区找到的项目进行合并,得到该社区中所有用户看过的项目的并集;最后为该并集中所有项目计算各自的评分平均分。项目的评分平均分计算如式(3):

$$avg_{ci} = \frac{sum_{ci}}{k} \tag{3}$$

其中,sum_{ci} 为社区 c 里的用户对项目 i 的评分总和;k 为社区 c 里的用户数。

Example 5 假设有一个社交网络共有 100 个用户,并且 k = 5 时用算法 1 进行社区划分的结果如表 2 所示。社区 1 里的用户给项目的评分情况如表 3 所示。则社区 1 对项目 1 的评分均值为 1/5 = 0.2,社区 1 对项目 3 的评分均值为(4+2)/5 = 1.2,…。依次计算,社区 1 的项目评分均值如表 4 所示。

表 2 社区划分结果

Table 2 Results of community division

社 区	用 户				
1	1	3	4	51	96
2	2	3	9	12	13
3	3	1	4	51	96
4	4	51	46	72	96
5	5	7	19	37	89
⋮	⋮	⋮	⋮	⋮	⋮

表 3 社区 1 里用户的项目评分数据

Table 3 Item score data of users in Community 1

用 户	项 目						
	1	3	6	7	11	13	…
1			1		5		
3		4	3	1	5		
4				4		5	
51	1			5		5	
96		2	1		2	3	

表 4 社区 1 的项目评分均值

Table 4 Average item scores for Community 1

社 区	项 目						
	1	3	6	7	11	13	…
1	0.2	1.2	1.0	2.0	2.4	2.6	

按照以上规则,给每个社区都计算其中包含的项目评分均值。

时间复杂度:算法 2 的时间复杂度为 $O(mks)$。算法需要遍历每个社区、每个用户以及每个用户的项目评分。因此,总的时间复杂度为 $O(mks)$,其中 m 是社区列表中的社区数量,k 是每个社区的用户数量,s 是平均每个用户评分的项目数量。

3.3 寻找相似社区

步骤 3 计算社区之间的相似度,然后寻找目标用

户所在社区的其他相似社区,如算法 3 所述。

算法 3:寻找相似社区

Input: 社区列表 *communities*,相似社区个数 *n*

Output: 相似社区列表 *sim_communities*

(1) 初始化空列表 *sim_communities* 用于存放相似社区

(2) 初始化空字典 *community_similarities* 用于记录两个社区之间的相似度

(3) for *community*1 in *communities*:

(4)　for *community*2 in *communities*:

(5)　　if *community*1! =*community*2 :

(6)　　利用式(4)计算社区之间的相似度

(7)　　*community_similarities. append*((*community*1, *community*2, *similarity*))

(8) *community_similarities. sort*(*reverse*=*True*, *key*=*lambdax*:*x*[2])

(9) for *i*=1 to *n* do

(10)　*sim_community*=*find_community_by_index*(*community_similarities*[*i*][2])

(11)　*sim_communities. append*(*sim_community*)

(12) return *sim_communities*

算法 3 首先遍历每个社区,然后计算每两个社区之间的相似度,两个社区里的相同用户数越多,那么这两个社区越相似。最后找与目标用户所在社区相似度最大的前 n 个社区。

寻找相似社区的具体做法如下:首先计算社区之间的共同用户数,然后根据式(4)计算社区 x,y 之间的相似度 $f_{(x,y)}$:

$$f_{(x,y)}=\begin{cases}0, node_x \cap node_y=0\\ \dfrac{|node_x \cap node_y|}{|2k|}, & \text{else}\\ 1, node_x \cap node_y=k\end{cases} \tag{4}$$

其中,*node* 代表社区里的用户节点。

Example 6　社区划分结果如表 2 所示,社区 1 和社区 5 没有共同用户数,则 $f_{(1,5)}$;社区 3 和社区 4 里共同用户是用户 4、用户 51 和用户 96,共同用户数是 3,则 $f_{(3,4)}=\dfrac{3}{10}$;社区 1 和社区 3 里共同用户数等于 k(社区规模的大小),则 $f_{(1,3)\,1,3}=1$。

按照上述方法,首先计算社区之间的相似度,然后通过比较目标用户所在的社区与其他社区之间的相似度来寻找与目标用户所在社区最相似的前 n 个社区。这样可以确定与目标用户有着类似兴趣和特征的其他社区。

Example 7　假设在算法 2 找到社区 1 的四个最相似的社区,是用户 3、用户 16、用户 57、用户 85。将社区 1 和这四个相似社区及其项目评分,构造成社区-项目评分均值矩阵,如表 5 所示。

表 5　社区-项目评分均值矩阵

Table 5　Matrix of community-item scoring means

社　区	项　目					
	1	2	3	5	6	…
1	$avg_{1,1}$	$avg_{1,2}$	$avg_{1,3}$		$avg_{1,6}$	
3		$avg_{3,2}$		$avg_{3,5}$		
16	$avg_{16,1}$		$avg_{16,3}$			
57				$avg_{57,5}$	$avg_{57,6}$	
85	$avg_{85,1}$		$avg_{85,3}$		$avg_{85,6}$	

时间复杂度:算法 3 的时间复杂度为 $O(m^2)$。m 个用户节点,共划分为 m 个社区,3-7 行计算社区之间的相似度,运行时间为 $O(m^2)$;9-11 行是寻找 n 个相似社区,时间复杂度为 $O(n^2)$。m 远远大于 n,所以算法 3 的时间复杂度为 $O(m^2)$。

由于用户的评分项目数 s 很大,远大于用户数 m,所以 SRCD 算法的时间复杂度为 $O(mks)$。

4　仿真实验与结果分析

4.1　实验设置

实验使用公开的 Film Trust 电影评分数据集,其中社交关系条数为 1 853,评分条数为 35 497,将评分数据随机划分成训练集和测试集,分别占比 80% 和 20%。数据集的详细统计数据如表 6 所示。

表 6　Film Trust 实验数据集统计分析

Table 6　Statistical information of Film Trust dataset

信　息	特　征	指标值
评分信息	用户数	1 508
	项目数	2 071
	评分数	35 497
	稀疏度	1.14%
社交信息	信任者数	609
	被信任者数	732
	信任数	1 853
	密度	0.42%

为验证所提算法的有效性,将用 SRCD 算法得到的社区-项目评分均值矩阵作为矩阵分解算法的输入,将矩阵分解的结果作为预测的用户-项目评分 $\hat{s}_{ui}$,实验步骤如图 4 所示。

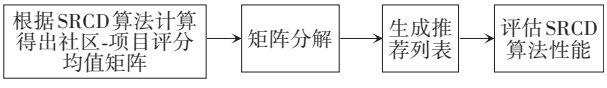


图 4　实验步骤

Fig. 4　Experimental procedure

SRCD 算法是用 Python 在 Pycharm 平台上实现的,与 3 种基于矩阵分解的社交推荐算法进行比较,这 3 种

社交推荐算法的介绍如表 7 所示。实验中的参数设置如表 8 所示。为了降低模型的复杂度,在本文的所有实验中都设置成 $\lambda_U=\lambda_I=10^{-3}$。

表 7 社交推荐算法和模型

Table 7 Social recommendation algorithms and models

	社交推荐算法模型	特点介绍
基于矩阵分解的社交推荐方法	RSTE[21]	一种信任感知的推荐方法,它将用户的评分建模为用户自身喜好和信任用户口味的平衡。
	SocialMF[22]	将信任传播机制纳入模型。
	SoReg[23]	一种社会正则化模型,该模型使用矩阵分解目标函数并加入了社会正则化的考虑。

表 8 实验参数设置

Table 8 Experimental parameter settings

变量名	描 述	默认值
λ_U	用户正则化参数	10^{-3}
λ_I	项目正则化参数	10^{-3}
λ_T	社交特征正则化参数	0.5
γ	偏置项计算学习速率	10^{-4}
q	冷启动用户对项目的评分数量	5
d	特征矩阵维度	10^1
steps	迭代次数	10^2

4.2 评估指标

为了评估 SRCD 算法的推荐和隐私保护的效果,这里采用两个常用的评估指标,平均绝对误差(MAE)和均方根误差(RMSE)。

平均绝对误差定义为

$$MAE_{ui}=\frac{1}{N}\sum_{u,i\in N}|s_{ui}-\hat{s}_{ui}|$$

均方根误差定义为

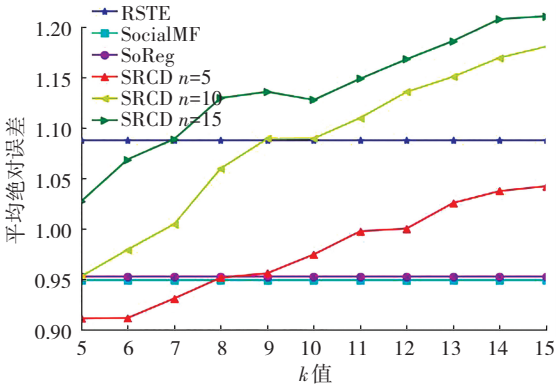
$$RMSE_{ui}=\sqrt{\frac{1}{N}\sum_{u,i\in N}|s_{ui}-\hat{s}_{ui}|^2}$$

其中,N 为测试集中的评分条数;s_{ui} 是用户 u 对项目 i 的真实评分;$\hat{s}_{ui}$ 是用户 u 对项目 i 的预测评分。MAE 和 RMSE 的值越小,意味着真实评分与预测评分之间的误差越小,推荐效果会越好。

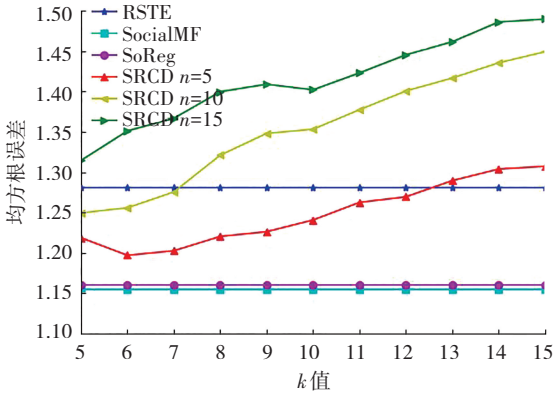
4.3 隐私保护对推荐结果的影响

实验的目的是考察所提出的 SRCD 算法在对用户隐私进行保护的同时对推荐效果的影响。社区里的用户数 k,相似社区个数 n 在不同取值的情况下对 MAE 和 RMSE 值的影响如图 5 所示。

从图 5 可以观察到,对于 SRCD 算法,不论 n 取 5、10 还是 15,在社区中的用户数 k 从 5 变化到 15 时,MAE 和 RMSE 的值都由小变大。这主要是因为随着社区中用户数量 k 的增加,用户个人数据会变得更加模糊不清。而相比之下,由于 RSTE 算法、SocialMF 算法和 SoReg 算法并未加入隐私保护机制,因此其 MAE 和 RMSE 值并不随社区中用户数 k 的变化而变化。



(a) 社区里的用户数 k 变化对 MAE 值的影响



(b) 社区里的用户数 k 变化对 RMSE 值的影响

图 5 隐私保护对推荐结果的影响

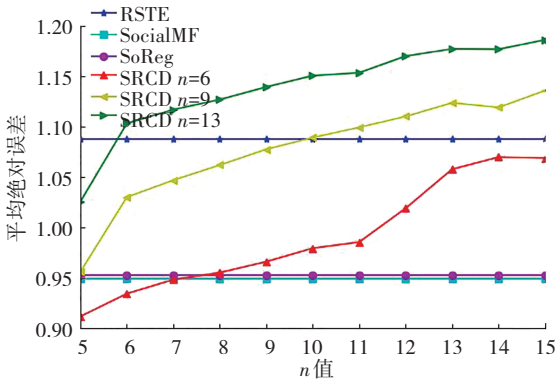
Fig. 5 The impact of privacy protection on recommendation results

根据图 5(a)的观察结果可以得知,当 n 取 5 时,不论社区中用户数 k 如何变化,SRCD 算法的 MAE 值均优于 RSTE 算法,这意味着 SRCD 算法提供了有效的推荐结果。在 n 取 5、k 小于 9 时,SRCD 算法的 MAE 值也比 SocialMF 算法和 SoReg 算法小;但是 k 大于等于 9 时,SRCD 算法的 MAE 值超过了 SocialMF 算法和 SoReg 算法,这说明在 n 取 5、k 取 9 时能够获得较好的推荐结果和一定程度的隐私保护。而当 n 取 10 或 15 时,除了 n 取 10、k 取 5 时,SRCD 算法的 MAE 值接近 SocialMF 算法和 SoReg 算法之外,在其他情况下 SRCD 算法的 MAE 值都超过了 SocialMF 算法和 SoReg 算法。但是,在 n 取 10、k 小于 10 或 n 取 15、k 小于 7 时,SRCD 算法的 MAE 值仍然优于 RSTE 算法,这说明在 n 取 10 或 15 时,SRCD 算法仍能获得相对满意的推荐效果。从图 5(b)可以看出,无论 n 取 5、10 还是 15,SRCD 算法的 RMSE 值都高于 SocialMF 算法和 SoReg 算法。但是,在 n 取 10、k 小于 13 时,SRCD 算法的 RMSE 值却优于 RSTE 算法,这说明在该情况下能够获得较好的

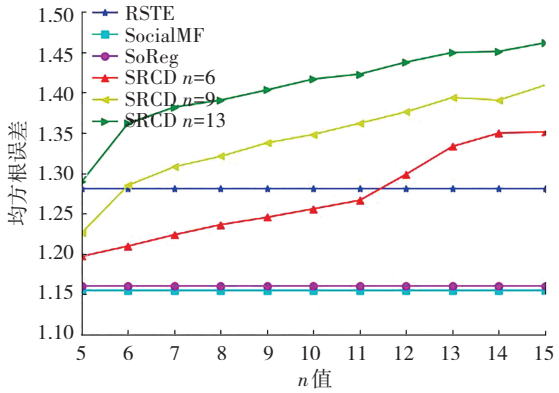
推荐效果。

4.4　推荐效果对比

实验的目的是考察所提出的 SRCD 算法的推荐有效性。实验根据 4.3 节的实验结果选取 k 的值,分别为 6、9 和 13,相似社区个数 n 在不同取值的情况下对 MAE 和 RMSE 值的影响如图 6 所示。



(a) 相似社区个数 n 变化对 MAE 值的影响



(b) 相似社区个数 n 变化对 RMSE 值的影响

图 6　隐私保护对推荐结果的影响

Fig. 6　The impact of privacy protection on recommendation results

根据图 6 的观察结果可以得知,对于 SRCD 算法而言,无论 k 取 6、9 还是 13,随着相似社区个数 n 从 5 变化到 15,MAE 和 RMSE 的值都逐渐增大。这主要是因为 SRCD 算法采用了用社区数据集合代替个人数据进行推荐的方式,由于社区数据集合模糊了个人数据,这种替代会导致推荐效果下降。而相比之下,RSTE 算法、SocialMF 算法和 SoReg 算法的 MAE 和 RMSE 的值并不随相似社区个数 k 的变化而变化。

根据图 6(a),当 k 等于 6 时,不论相似社区的数量如何变化,SRCD 算法的 MAE 值都比 RSTE 算法的小。这表明 SRCD 算法提供了有效的推荐结果。当 k 为 6 且 n 小于 8 时,SRCD 算法的 MAE 值比 SocialMF 算法和 SoReg 算法的小;而当 n 大于等于 8 时,SRCD 算法的 MAE 值则比 SocialMF 算法和 SoReg 算法大。在 k 取 9 或 13 时,除了 k 取 9、n 取 5 时,SRCD 算法的 MAE 值接近于 SocialMF 算法和 SoReg 算法的值外,其他情况下不论相似社区的数量如何变化,SRCD 算法的 MAE 值都比 SocialMF 算法和 SoReg 算法的 MAE 值大。但是,在 k 取 9 且 n 小于 10 或者 k 取 13 且 n 小于 6 时,SRCD 算法的 MAE 值比 RSTE 算法的小,这说明 SRCD 算法在 k 取 9 或 13 时也能够获得较满意的推荐效果。根据图 6(b),在 k 取 6、9 或 13 时,不论相似社区的数量是多少,SRCD 算法的 RMSE 值都比 SocialMF 算法和 SoReg 算法的大。然而,在 k 取 6 且 n 小于 12 时,SRCD 算法的 RMSE 值比 RSTE 算法的小,这说明此时可以获得较好的推荐结果。

综合以上两组实验结果可知:在 n 取 5、k 取 6 时,SRCD 算法的推荐效果最好;在 n 取 5、k 取 9 时能够在推荐和隐私保护之间取得较好地平衡;在 n 取 5、k 取 13 时既能够较好地保护用户隐私,又能够获得较好的推荐效果。

5　结　论

现有的社交推荐方法主要关注如何提高推荐准确性,而忽略了用户个人信息的隐私保护问题。因此,本文提出了一种基于社区划分的隐私保护社交推荐方法。该方法首先将社交网络中的用户进行社区划分,然后使用社区数据集合替代个人数据集合进行推荐,以保护用户的个人隐私。接下来,通过寻找与目标用户所在社区最相似的 n 个社区,来给目标用户提供满意的推荐结果。实验结果表明:相比与现有的基于矩阵分解的社交推荐方法,提出的方法能够同时提供用户满意的推荐结果和一定程度的个人隐私保护。未来的研究计划如何处理动态社交网络中用户隐私推荐问题,以进一步提高社交推荐系统的推荐准确性和用户隐私保护效果。

参考文献(References):

[1] 黄勃, 严非凡, 张昊, 等. 推荐系统研究进展与应用[J]. 武汉大学学报(理学版), 2021, 67(6): 503-516.
HUANG Bo, YAN Fei-fan, ZHANG Hao, et al. Progress and application of recommendation system[J]. Journal of Wuhan University (Natural Science Edition), 2021, 67(6): 503-516.

[2] KOTA N R, PADMANABHAN V, BHUKYA W N. Content based network representational learning for Movie recommendation (CNMovieRec)[C]//Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023: 112-123.

[3] 张悦琳, 王创剑. 基于人工智能知识库的营养膳食推荐系统研究[J]. 重庆工商大学学报(自然科学版): 1-13[2023-08-03].
ZHANG Yue-lin, WANG Chuang-jian. Research on nutrition diet recommendation system based on artificial intelligence knowledge base[J]. Journal of Chongqing Technology and Business University (Natural Science Edition): 1-13[2023-

08-03].

[4] 李文才. 大数据背景下基于个性化推荐的安全隐私问题综述[J]. 网络安全技术与应用, 2023(5): 67-71.
LI Wen-cai. Summary of security and privacy issues based on personalized recommendation in the context of big data[J]. Network Security Technology & Application, 2023(5): 67-71.

[5] 张岐山, 翁丽娟. 社会化推荐系统综述[J]. 计算机工程与应用, 2020, 56(1): 1-10.
ZHANG Qi-shan, WENG Li-juan. Review of social recommender systems[J]. Computer Engineering and Applications, 2020, 56(1): 1-10.

[6] 刘生昊, 吴国洋, 邓贤君, 等. 推荐系统与隐私保护研究综述[J]. 华中科技大学学报(自然科学版), 2023, 51(2): 1-9.
LIU Sheng-hao, WU Guo-yang, DENG Xian-jun, et al. Survey of recommender system and privacy protection[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2023, 51(2): 1-9.

[7] CHECCO A, BRACCIALE L, LEITH D J, et al. OpenNym: Privacy preserving recommending via pseudonymous group authentication[J]. Security and Privacy, 2022, 5(2): 1-23.

[8] 鲜英, 于炯, 薛朋强. 基于改进匿名模型的上下文推荐系统研究[J]. 计算机工程, 2018, 44(3): 214-219.
XIAN Ying, YU Jiong, XUE Peng-qiang. Research on context recommendation system based on improved anonymity model[J]. Computer Engineering, 2018, 44(3): 214-219.

[9] 王海艳, 陆金祥. 面向群组推荐的个性化隐私保护方法[J]. 通信学报, 2019, 40(9): 106-115.
WANG Hai-yan, LU Jin-xiang. Personalized privacy protection method for group recommendation[J]. Journal on Communications, 2019, 40(9): 106-115.

[10] 彭丽寻, 刘丰恺. 基于个性化 k 匿名隐私保护的资源推荐算法[J]. 电脑与电信, 2020(6): 66-73.
PENG Li-xun, LIU Feng-kai. Resource recommendation algorithm based on k-anonymity for generalizing user query requests[J]. Computer & Telecommunication, 2020(6): 66-73.

[11] YI J, WU F, WU C, et al. Efficient-FedRec: efficient federated learning framework for privacy-preserving news recommendation[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2021: 2814-2824.

[12] BASSEM C. Oncooperative obfuscation for privacy-preserving task recommendation in mobile CrowdSensing[C]//Proceedings of the 17th International Conference on Wireless and Mobile Computing, Networking and Communications. Piscataway: IEEE Press, 2021: 90-95.

[13] LI W, CHEN H, ZHAO R, et al. A federated recommendation system based on local differential privacy clustering[C]//Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation. Piscataway: IEEE Press, 2021: 364-369.

[14] LUO J, YI X, HAN F, et al. An efficient clustering-based privacy-preserving recommender system[C]//Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2022: 387-405.

[15] SAHOO A K, RAJ S, PRADHAN C, et al. Perturbation-based fuzzified K-mode clustering method for privacy preserving recommender system[J]. International Journal of Information Security and Privacy, 2022, 16(1): 1-20.

[16] QIE H, DOU Y, HUANG Z, et al. Isolate setsbased parallel Louvain method for community detection[J]. Journal of Computer Science and Technology, 2023, 38(2): 373-390.

[17] ZHANG Z, MARTIN A, LIU Z, et al. Fastunfolding of credal partitions in evidential clustering[C]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021: 3-12.

[18] ZHU X, WU X, WU B, et al. An improved fuzzy C-means clustering algorithm using Euclidean distance function[J]. Journal of Intelligent & Fuzzy Systems, 2023, 44(6): 9847-9862.

[19] MOHANA H, SURIAKALA M. Integrated cosine and tuned cosine similarity measure to alleviate data sparsity issues for personalized recommendation[C]//Proceedings of the 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions. Piscataway: IEEE Press, 2018: 41-49.

[20] ZHANG F, ZHOU W, SUN L, et al. Improvement of Pearson similarity coefficient based on item frequency[C]//Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition. Piscataway: IEEE Press, 2017: 248-253.

[21] MA H, KING I, LYU M R. Learning to recommend with social trust ensemble[C]//Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. New York: ACM, 2009: 203-210.

[22] JAMALI M, ESTER M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//Proceedings of the fourth ACM conference on Recommender systems. New York: ACM, 2010: 135-142.

[23] MA H, ZHOU D, LIU C, et al. Recommender systems with social regularization[C]//Proceedings of the fourth ACM international conference on Web search and data mining. New York: ACM, 2011: 287-296.

责任编辑:陈 芳