

文章编号:1671-4229(2023)01-0009-08

基于用户评分一致性的协同过滤个性化推荐算法

白源¹, 马浚¹, 刘松华², 李泽鹏^{1*}

(1. 兰州大学 信息科学与工程学院, 甘肃 兰州 730000; 2. 太原理工大学 信息与计算机学院, 山西 晋中 030600)

摘要: 在目前信息高速发展的时代,个性化推荐作为信息过滤的重要手段,是解决信息过载的最有效方法之一。协同过滤一直是解决个性化推荐比较热门的技术,其主要思想是计算用户之间的相似性或计算项目之间的相似性,然后根据用户或项目之间的相似性对目标用户进行推荐。文章基于协同过滤的思想,提出了一种结合用户评分一致性的单模投影算法,首先在用户与项目的关系二部图中计算用户之间的评分一致性,然后把一致性赋值作为压缩之后的单模投影权值,最后用K近邻找到相似用户并做出个性化推荐。在MovieLens、FilmTrust和Jester等真实数据集上的实验表明,基于评分一致性的推荐算法达到了较好的效果。

关键词: 个性化推荐; 协同过滤; 单模投影; 一致性; K近邻

中图分类号: TP 311 **文献标志码:** A

Collaborative filtering personalized recommendation algorithm based on user scoring consistency

BAI Yuan¹, MA Jun¹, LIU Song-hua², LI Ze-peng^{1*}

(1. School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China;
2. College of Information and Computer, Taiyuan University of Technology, Jinzhong 030600, China)

Abstract: In the era of rapid development of information, personalized recommendation, as an important means of information filtering, is one of the most effective ways to solve information overload. Collaborative filtering has always been a popular technology to solve personalized recommendation. Its main idea is to calculate the similarity between users or calculate the similarity between projects, and then recommend to the target users according to the similarity between users or projects. Based on the idea of collaborative filtering, this paper proposes a single-mode projection algorithm combined with user scoring consistency. Firstly, the scoring consistency between users is calculated in the relationship bipartite graph formed by users and projects. Then the consistency assignment is taken as the weight of the compressed single-mode projection. Finally, the method of K-nearest neighbor is used to find similar users and to make personalized recommendations. Experiments on real datasets such as MovieLens, FilmTrust and Jester show that the recommendation algorithm based on score consistency achieves better results.

Key words: personalized recommendation; collaborative filtering; one-mode projection; consistency; K-nearest neighbor

收稿日期: 2022-06-06; 修回日期: 2022-09-29

基金项目: 国家自然科学基金资助项目(61802158, 61762047); 甘肃省自然科学基金资助项目(20JR10RA605); 甘肃省重点研发计划资助项目(20YF3FA024)

作者简介: 白源(1994—), 男, 硕士研究生. E-mail: baiy2018@lzu.edu.cn

*通信作者. E-mail: lizp@lzu.edu.cn

引文格式: 白源, 马浚, 刘松华, 等. 基于用户评分一致性的协同过滤个性化推荐算法[J]. 广州大学学报(自然科学版), 2023, 22(1): 9-16.

近年来,随着互联网的发展和计算机性能的不断提高,Facebook、腾讯、淘宝和新浪微博等许多大型的社交网络得到迅速发展,从而导致了信息过载问题,如何从海量信息里筛选有用的信息变得越来越重要。特别是在一些大型的社交网站或购物平台上进行选择时,用户往往会不知所措。为解决这些问题,个性化推荐系统逐渐成为研究热点。个性化推荐主要根据每个用户的历史浏览和评分记录,分析用户个人的喜好,进而对每个用户进行个性化推荐。到目前为止,大部分社交软件和购物平台都进行了个性化推荐的研究,从而提高了用户的使用效率,为用户节省了大量的时间。

目前提出的推荐算法主要有:基于协同过滤推荐算法^[1-7]、基于内容的推荐算法^[8-9]和基于图结构的推荐算法^[10-14]等。

协同过滤一直是解决个性化推荐比较热门的技术^[1-5],其主要思想是基于用户之间或项目之间的相似性来进行个性化推荐。Choi等^[6]提出了结合用户相似性和项目相似性的协同过滤。计算相似性的方法主要有:基于余弦的相似性、基于相关性的相似性和基于余弦变形的相似性。针对协同过滤方法中数据稀疏的问题,Phuong等^[7]考虑到用户之间的间接关系,提出了矩阵乘法的方法,从而降低了数据稀疏性产生的负面影响。

基于内容的推荐方法^[8-9]一般可分为以下3个步骤:①通过描述可能被推荐的项目;②构建一个用户文档,该用户文档用来记录用户喜欢的项目类型;③根据构建的用户文档类型的信息,用项目与文档记录的喜好类型作对比,最终为用户推荐出可能的项目。值得注意的是,由于用户在不断购买新项目,所以需要不断地维护和更新这个用户文档。

基于图的推荐算法首先构建一个图模型,然后根据图的特性挖掘用户之间的关系,进而做出推荐。Shams等^[10]利用用户、项目及评分等级对构建了一个三部图,然后通过PageRank方法得到了对应的推荐列表。Zhou等^[11]提出了二部图资源分配推荐算法。Wang等^[12]结合相似函数和二部图提出了一个提高资源配置的相似网络。

Shams等^[13]提出了基于图的信息网络去模拟用户的优先权,然后推断出一个推荐列表。赵红等^[14]在基于加权二部图个性化推荐算法的基础上又提出了融合初始资源与协同过滤的二部图推荐算法。

为了提高推荐的效率和准确率,新的推荐方法一直被不断地更新。其他的推荐算法还有在线推荐算法^[15]、基于PageRank推荐算法^[16]和基于机器学习的推荐算法^[17-18]等。

本文在基于协同过滤的基础上,根据不同用户对同一项目评分的差异,引入了用户评分一致性的概念。首先构建用户和项目的二部图,计算用户之间的评分一致性,然后将用户项目二部图向用户节点进行投影,得到用户节点之间的关系图,其中,图的权值为评分一致性。此外,借助Phuong等^[7]提出的矩阵乘法思想计算用户之间的间接关系,进而得到用户之间的相似性,最后使用K近邻算法给目标用户进行推荐。

1 基于用户评分一致性的协同过滤推荐方法

1.1 相关概念

二部图:设 $G=(V,E)$ 是一个无向图,若节点集 V 可划分为两个互不相交的子集 X 和 Y ,使得 G 的每条边的两个端点分别属于 X 和 Y 中,则称 G 为二部图。一个二部图 $G(X,Y,E)$, X 和 Y 分别表示两个节点集, E 表示边的集合。令 $X=\{x_1,\dots,x_n\}$, $Y=\{y_1,\dots,y_m\}$ 。图 G 用邻接矩阵可以表示为

$$a_{ij} = \begin{cases} 1, & \text{若 } x_i \text{ 与 } y_j \text{ 相邻} \\ 0, & \text{若 } x_i \text{ 与 } y_j \text{ 不相邻} \end{cases} \quad (1)$$

评分一致性:假设有两个用户 u 和 v ,两个用户都对一个项目 p 进行了评分,用 $t(u,p)$ 表示用户 u 对项目 p 的归一化评分,用 $t(v,p)$ 表示用户 v 对项目 p 的归一化评分,那么定义 u 和 v 对项目 p 的评分一致性 $Con_p(u,v)$ 可定义为式(2)。

$$Con_p(u,v) = 1 - |t(u,p) - t(v,p)| \quad (2)$$

评分一致性是对用户评分差异细粒度的表示。若两个用户 u 和 v 对同一项目 p 的评分差异越小,即 $|t(u,p) - t(v,p)|$ 越小,说明用户对该项目的兴趣

程度越相近,从而评分一致性 $Con_p(u, v)$ 越大;反之,若两个用户 u 和 v 对同一项目 p 的评分差异越大,即 $|t(u, p) - t(v, p)|$ 越大,说明用户对该项目的兴趣程度相差越大,从而评分一致性 $Con_p(u, v)$ 越小。

进一步,如果用户对一个项目进行评分,则说明这个用户对该项目有某种程度的兴趣,故引入参数 a 表示用户的初始兴趣程度。结合评分一致性公式(2),可以得出带初始兴趣程度的一致性为

$$Con_p(u, v) = a + (1 - a)(1 - |t(u, p) - t(v, p)|) \quad (3)$$

二部图一致性计算方法:设定两个用户的一致性为对共同节点一致性的累加,然后除以用户节点在二部图中的度计算,见图 1。

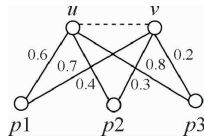


图 1 一致性差异

Fig. 1 The difference of consistencies

图 1 中,上面两个节点分别表示用户 u 和 v ,下面 3 个节点表示项目 $p1, p2, p3$, 用户 u 和 v 对项目 $p1, p2, p3$ 的评分一致性分别记为 $Con_{p1}(u, v)$ 、 $Con_{p2}(u, v)$ 和 $Con_{p3}(u, v)$ 。用 $deg(u)$ 和 $deg(v)$ 表示用户 u 和 v 在二部图中的度数。最后用户 u 和 v 的一致性可表示为

$$Con(u, v) = \frac{Con_{p1}(u, v) + Con_{p2}(u, v) + Con_{p3}(u, v)}{\sqrt{deg(u) * deg(v)}} \quad (4)$$

如果两个用户节点 u 和 v ,对多个项目 $p1, p2, \dots, pn$ 进行了评分,根据公式(4),进行如下推广:

$$Con(u, v) = \frac{\sum_{i=1}^n Con_{p_i}(u, v)}{\sqrt{deg(u) * deg(v)}} \quad (5)$$

1.2 二部图向用户节点单模投影

单模投影^[11]就是通过对二部图进行只向一个节点集合的压缩,压缩后的两个节点相邻当且仅当它们在原图中至少有一个共同的邻居节点,见图 2。

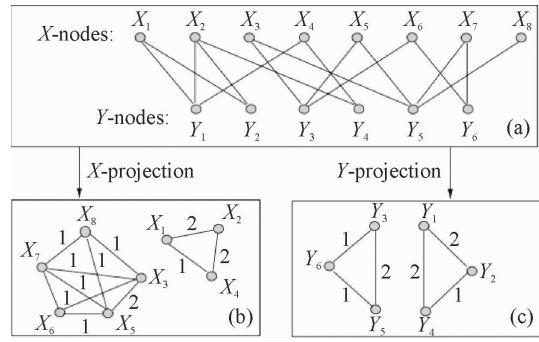


图 2 单模投影

Fig. 2 One-mode projection

但是这个投影的过程并没有考虑到权值。这里有一个简单的例子,两个人分别搜集了 100 首音乐,但是只有一首音乐是两个人共同挑选的,那么就可以推断出这两个人音乐兴趣不相同,但是如果有许多音乐是两个人共同挑选的,那么推断他们两个人的音乐喜好大致相同。但是在二部图压缩到没有权值的单模后,会丢失掉部分信息。一个比较简单的单模分配权值的方式是根据两个指定节点在另外集合的邻居节点数给定权值,但是该方法有偏差。怎样分配权值没有一个固定的方法,目前还在研究过程中。

文献[11]还提出了一个给单模投影赋权值的方法,在图 2 中,用户和项目的二部图见图 2(a), X 节点表示项目, Y 节点表示用户,在这里,首先对图 2(a)的二部图做 Y -projection 投影,也就是向用户节点投影,见图 2(c)。单模投影后得到用户之间的关系图,然后通过式(5)计算用户之间的一致性,并给投影后用户节点之间的边赋权值为一致性值。

2 矩阵乘法计算用户之间的相似性

前面通过单模投影得到了用户的一致性单模投影图,下面根据 Phuong 等^[7]的方法使用矩阵乘法挖掘用户之间的间接关系,进而得到用户之间的相似性。

首先对一致性关系图中的每一对用户 u 和 v ,定义用户 u 和用户 v 的相似性为这两个用户之间的所有不同路径一致性相乘,然后再相加得到两个用户的相似性。在这里,首先计算用户之间的一致性矩阵 $W_{n \times n}$, n 表示用户的总数。用 Z 表示

用户相似性矩阵:

$$\mathbf{Z} = \mathbf{W}^q, q = 2, 3, 4, 5 \dots \quad (6)$$

3 K 近邻推荐

K 近邻算法是首先计算用户之间的相似性, 然后根据相似性的大小选取与目标用户最相似的前 K 个用户, 最后按照相似性和相似用户的评分对目标用户进行有效的推荐。K 近邻算法的推荐步骤如下: 给定一个目标用户 u , 选择 K 个与 u 最相似的用户, 用 U 来表示与 u 最为相似的用户集合, 对于 u 未评分的项目 p , u 对 p 的评分 $Rate_{up}$ 可以被估算为

$$Rate_{up} = \sum_{v \in U} S_{uv} * r_{vp} \quad (7)$$

在这里 $Rate_{up}$ 表示用户 u 对未评分项目 p 的评分预测值, S_{uv} 表示用户 u 和用户 v 的相似性, r_{vp} 表示用户 v 对项目 p 的评分。

一致性算法描述的伪代码为

输入: 用户 - 项目评分矩阵, 目标用户 ut , 最相似邻居数 K , 初始兴趣度 a 。

输出: 给目标用户 ut 推荐的 N 个项目。

- (1) Step1: 将每个用户对项目的评分除以最大值进行归一化;
- (2) Step2: 将用户和项目的二部图向用户节点做单向投影, 用户之间的权值为一致性值;
- (3) $S = \emptyset$
- (4) for each user $u \in U$ do
- (5) $S = S \cup \{u\}$
- (6) for each user $v \in U \setminus S$ do
- (7) $weight(u, v) = 0$
- (8) while (u and v have common rating item p)
- (9) create node u and v
- (10) $Con_p(u, v) = 1 - |t(u, p) - t(v, p)|$
- (11) $weight(u, v) := weight(u, v) + Con(u, v)$
- (12) end while
- (13) end for

(14) end for

(15) Step3: 通过评分一致性图计算目标用户与其他用户的相似性

(16) let $\mathbf{W}_{n \times n} = [weight(u, v)]_{n \times n}$

(17) compute $\mathbf{Z} = \mathbf{W}^q, q = 2, 3, 4, 5 \dots$

(18) denote by $S_{uv} = \mathbf{Z}(u, v)$

(19) Step4: 生成用户 u 的推荐列表

(20) for u , select K users with the highest similarity scores to u and denote by U the set of these K users

(21) estimate each missing rating r_{up} of u for item p

(22) $Rate_{up} = \sum_{v \in U} S_{uv} * r_{vp}$

(23) Return K items for u with the highest rating

4 实验分析

本节在不同的数据集上, 对一致性单模投影算法和传统的一些算法进行了分析。对比算法包括: 基于图的喜好匹配协同排名算法 (PMCR)^[19]、资源分配算法 (Reource-Allocation)^[11]、基于随机游走的算法 (PersonalRank)、基于用户的协同过滤算法 (User-CF) 和评分一致性协同过滤算法 (Consis-CF), 对以上算法进行实验结果对比, 进一步分析一致性单模投影算法的效率和准确率。

4.1 数据集

本文采用 3 个数据集 MovieLens, FilmTrust 和 Jester 进行实验分析。数据集 MovieLens^① 包含 973 个用户对 1 682 部电影产生 100 000 条评分记录。评分的分值为 1 ~ 5 分。数据集 FilmTrust^② 包含 1 508 个用户对 2 071 个项目产生的 35 497 条评分记录, 评分的分值为 0.5 ~ 4 分。数据集 Jester^③ 包含 7 699 个用户对 158 个玩笑产生的 100 000 条评分记录, 评分的分值为 -10 ~ 10 分。为了计算方便, 对分值进行归一化处理, 把所有的分值都归一化到区间 [0 - 1]。本实验把数据集分为训练集和测试集, 并且将用户对项目的评分数

① <https://grouplens.org>

② <https://www.librec.net>

③ <http://eigentaste.berkeley.edu>

据按照均匀分布随机分成8份,挑选一份作为测试集,剩下的7份作为训练集,该实验在训练集上建立模型,并在测试集上对用户的评分进行预测。为保证评测指标不过拟合,需要多次实验,每次使用不同的测试集,然后取评测指标的平均值作为最终评测指标。

4.2 评价指标

准确率和召回率是推荐系统最常用的两种评价指标,经常被用于评价各种推荐算法的合理性。文献[20]详细叙述了这两种指标,下面将详细阐述准确率和召回率。

准确率:表示用户对系统推荐商品感兴趣的概率。在计算准确率的时候,常用的做法是设定推荐列表长度 L ,根据预测评分对所有待预测商品排序,系统认为排在前 L 位的商品是用户最可能喜欢的,因此,将它们推荐给用户。于是,对于一个未曾被用户选择或评分的商品,最终可能的结果有4种(表1),即系统推荐给用户且用户很喜欢、系统推荐给用户但是用户不喜欢、用户喜欢但是系统没有推荐和用户不喜欢且系统没有推荐。

表1 评价指标

Table 1 Evaluation index

用户喜好	系统推荐	系统不推荐
喜欢	True-Positive N_{tp}	False-Negative N_{fn}
不喜欢	False-Positive N_{fp}	True-Negative N_{tn}

将系统中所有用户的准确率求平均得到系统整体的推荐准确率,表示为

$$P(L) = \frac{1}{M} \sum_u P_u(L) \quad (8)$$

召回率:表示一个用户喜欢的商品被推荐的概率,定义为推荐列表中用户喜欢的商品与系统中用户喜欢的所有商品的比率:

$$R_u(L) = \frac{N_{tp}}{B_u} = \frac{N_{tp}}{N_{tp} + N_{fn}} \quad (9)$$

4.3 参数值 a 对准确率的影响

为了更好地挖掘用户对项目的兴趣程度,一致性协同过滤算法对用户的评分进行了更加详细的实验。假设用户对项目评过 a 分,那么用户对该项目具有一定的兴趣度。参数 a 就用来调节初始的兴趣和一致性协同过滤算法表示的兴趣程度。如图3所示,在不同的数据集中,对不同的参数 a 进行实验分析。

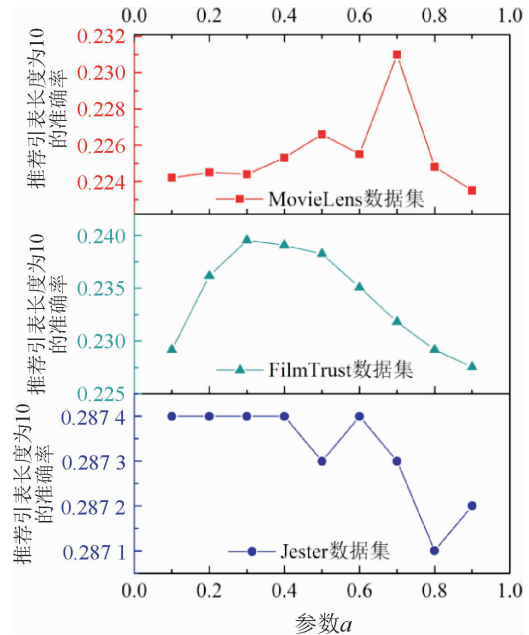


图3 不同数据集下参数 a 对准确率的影响

Fig. 3 Influence of parameter a to accuracy under different data sets

由图3可以看出,针对不同的数据集,参数 a 的取值会有明显的不同。在MovieLens数据集中,当参数 a 从0.1增加到0.5时,准确率也随之上升;当参数 a 从0.5增加到0.6时,准确率在下降;当参数 a 从0.6增加到0.7时,准确率又继续上升;当参数 $a=0.7$ 时,准确率达到峰值0.23;当参数 a 超过0.7时,准确率呈下降趋势。在FilmTrust数据集中,当参数 a 从0.1增加到0.3时,准确率也相应地随之增加;当参数 $a=0.3$ 时,准确率达到其峰值;当参数 a 超过0.3时,准确率开始呈现下降趋势。在Jester数据集中,该数据集对参数 a 的变化并不敏感,随着参数 a 的变化,准确率其实增幅不算太大,一直保持比较平稳的趋势。

4.4 矩阵乘法中 L 值的选取

本文利用矩阵乘法近似计算用户之间的间接影响,通过把用户对项目评分的二部图进行向用户节点的单模投影,得到用户之间一致性的关系矩阵,通过矩阵乘法,可以进一步得到用户之间的间接联系,从而提高推荐的准确率。

从图4中可以看到,对不同的数据集,矩阵乘法中参数 $L=3$ 时,数据集Jester和FilmTrust的准确率Precision10达到最大值,因此,矩阵乘法中取 $L=3$ 。

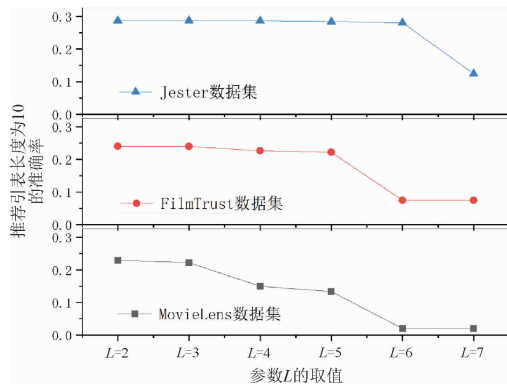


图 4 不同数据集下矩阵乘法中 L 对准确率的影响

Fig. 4 Influence of L in matrix multiplication to accuracy under different data sets

4.5 KNN 的 K 值对结果的影响

通过分别对不同数据集下面的 KNN 的 K 值进行实验,见图 5,当 K 取不同的值时,最后在不同的数据集上的实验结果达到了不同的结果,cos - 表示基于余弦相似性的协同过滤算法,consis - 表示一致性协同过滤算法,随着 K 值不断增大,一致性协同过滤算法比协同过滤算法推荐更准确。在 MovieLens 数据集和 FilmTrust 数据集中,cos 协同过滤在将近 $K = 60$ 时就已经达到极值,而一致性还有稳定的上升趋势。在 Jester 数据集中,一致性算法比协同过滤算法有更好的收敛性,在 K 值比较小的时候,一致性算法比协同过滤有更好的收敛性。

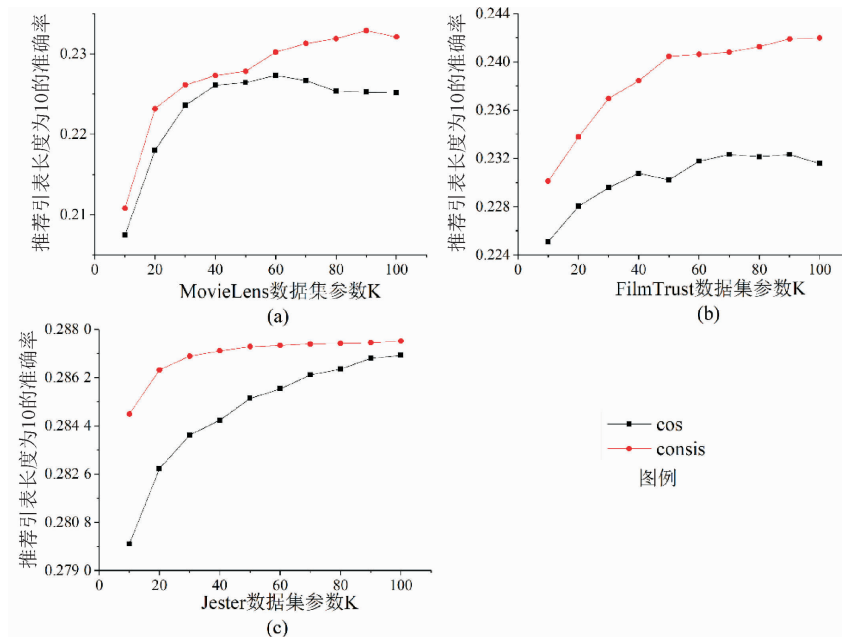


图 5 不同数据集下 K 对准确率的影响

Fig. 5 Influence of K to accuracy under different data sets

从图 5 中可以观察到,在后面的研究中,为了方便研究,针对不同数据集,选取一个特定的 K 值进行实验,在 MovieLens 数据集上,选择 $K = 40$ 进行实验,在 FilmTrust 数据集上,选择 $K = 160$ 进行实验,在 Jester 数据集上,选择 $K = 70$ 进行实验。

4.6 计算时间比较

由于基于随机游走的算法(PersonalRank)、资源分配算法(Reource-Allocation)和基于图的喜好匹配协同排名算法(PMCR)的时间不会随着 K 值的变化而改变,所以其运行时间保持不变。基于用户的协同过滤算法(User-CF)的计算时间分为相似性计算时间和推荐时间,评分一致性协同过滤算法(Consis-CF)分为评分一致性计算时间和推

荐时间。以 MovieLens 数据集为例,图 6(a)表示在 MovieLens 数据集上,User-CF 对 943 个用户相似性计算时间为 1.13 min,每个用户的相似性计算时间就为 0.134 s,每个用户在 $K = 10$ 到 $K = 100$ 的推荐时间分别为:0.08 s、0.17 s、0.26 s、0.35 s、0.44 s、0.52 s、0.61 s、0.69 s、0.78 s 和 0.86 s,则不同 K 值的计算时间为相似性计算时间加推荐时间为:0.15 s、0.24 s、0.33 s、0.42 s、0.51 s、0.59 s、0.68 s、0.77 s、0.85 s 和 0.93 s。Consis-CF 对每个用户的评分一致性计算时间为 0.205 s,每个用户在 $K = 10$ 到 $K = 100$ 的总推荐时间分别为:0.30 s、0.39 s、0.49 s、0.58 s、0.68 s、0.77 s、0.87 s、0.96 s、1.06 s 和 1.15 s。

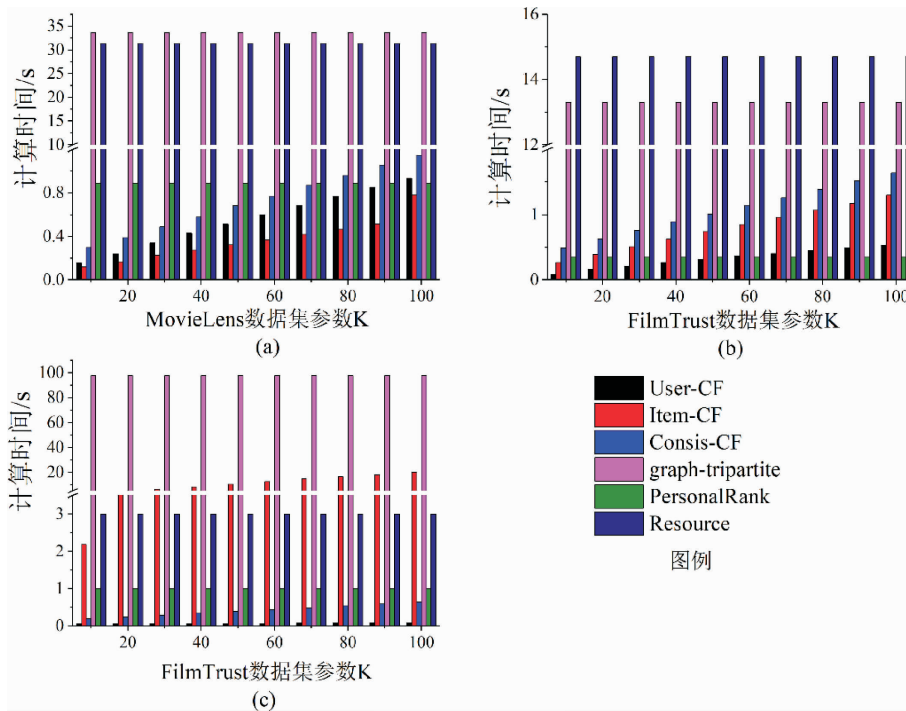


图 6 不同数据集下不同 K 值的计算时间

Fig. 6 Computing time of different K values under different data sets

4.7 结果

通过把一致性协同过滤算法 (Consis-CF) 和不同的算法进行比较,结果表明该算法在某种程度上达到了比较令人满意的推荐准确率。下面实验中推荐项目的个数都为 10。

从表 2 中可以看出,在 MovieLens 的数据集下,通过对比传统的基于用户的协同过滤算法 User-CF、PersonalRank 算法、Recurse-Allocation 算法和当前比较新的基于图的喜好匹配协同排名算法 (PMCR),可以看到评分一致性协同过滤算法 (Consis-CF) 的准确率(23.29%)和覆盖率(19.27%)相对较高。

表 2 MovieLens 数据集的实验结果

Table 2 Experimental results on MovieLens %

算法	评价指标	
	Precision	Recall
PMCR	16.09	13.31
PersonalRank	17.03	15.36
Recurse-Allocation	18.56	18.57
User-CF	22.73	18.81
Consis-CF	23.29	19.27

从表 3 中,可以看到评分一致性协同过滤算法 (Consis-CF) 在 FilmTrust 数据上,其准确率

(24.41%) 和召回率(69.45%)也比对比算法高。

表 3 FilmTrust 数据集的实验结果

Table 3 Experimental results on FilmTrust %

算法	评价指标	
	Precision	Recall
PMCR	23.70	67.42
PersonalRank	23.92	68.04
Recurse-Allocation	24.11	68.59
User-CF	23.29	66.25
Consis-CF	24.41	69.45

从表 4 中可以看到评分一致性协同过滤算法 (Consis-CF) 在 Jester 数据集上,其效果和其他算法相差不多。

表 4 Jester 数据集的实验结果

Table 4 Experimental results on Jester %

算法	评价指标	
	Precision	Recall
PMCR	28.75	95.18
PersonalRank	28.45	95.16
Recurse-Allocation	28.75	95.19
User-CF	28.74	93.12
Consis-CF	28.76	95.19

5 结 论

该文提出了评分一致性协同过滤算法,通过数据集中用户对项目的评分,考虑了两个用户对共同项目评分的差异,进而得到两个用户的一致性。该文首先把二部图向用户节点做单向投影,然后把一致性赋值为投影后边的权值,之后使用

矩阵乘法计算用户相似性,最后再用 K 近邻方法找到相似用户进行个性化推荐。在 MovieLens 数据集上,一致性协同过滤算法和传统的协同过滤算法计算时间大致相同,准确率有明显的提高,在 FilmTrust 和 Jester 数据集上,评分一致性协同过滤算法比传统的协同过滤算法计算时间稍多,但准确率有明显的提高。

参考文献:

- [1] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1):76-80.
- [2] Deng A L, Zhu Y Y, Shi B L. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of Software, 2003, 14(9):54-65.
- [3] Gao M, Ling B, Yang L D, et al. From similarity perspective: A robust collaborative filtering approach for service recommendations[J]. Frontiers of Computer Science, 2019, 13(2): 231-246.
- [4] Ortega F, Rojo D, Valdiviezo-diaz P, et al. Hybrid collaborative filtering based on users rating behavior[J]. IEEE Access, 2018, 6: 69582-69591.
- [5] 赵明, 闫寒, 曹高峰, 等. 融合用户信任度和相似度的基于核心用户抽取的鲁棒性推荐算法[J]. 电子与信息学报, 2019, 41(1): 180-186.
- [6] Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering[J]. Knowledge-Based Systems, 2013, 37(1):146-153.
- [7] Phuong T M, Lien D T, Phuong N D. Graph- based context-aware collaborative filtering[J]. Expert Systems with Applications, 2019, 126: 9-19.
- [8] Martinez L, Perez L G, Barrabco M. A multigranular linguistic content-based recommendation model[J]. International Journal of Intelligent Systems, 2007, 22(5):419-434.
- [9] Liu L, Lecue F, Mehandjiev N. Semantic content-based recommendation of software services using context[J]. ACM Transactions on the Web, 2013, 7(3):1-20.
- [10] Shams B, Haratizadeh S. Graph-based collaborative ranking[J]. Expert Systems with Applications, 2017, 67: 59-70.
- [11] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 2007, 76(4):1-7.
- [12] Wang X M, Liu Y, Xiong F. Improved personalized recommendation based on a similarity network[J]. Physica A: Statistical Mechanics and its Applications, 2016, 456: 271-280.
- [13] Shams B, Haratizadeh S. Reliable graph-based collaborative ranking[J]. Information Sciences, 2018, 432: 116-132.
- [14] 赵红, 郑骏. 融合初始资源与协同过滤的二部图推荐算法[J]. 计算机应用与软件, 2019, 36(1):297-301.
- [15] Liu D R, Chen K Y, Chou Y C, et al. Online recommendations based on dynamic adjustment of recommendation lists[J]. Knowledge-Based Systems, 2018, 161: 375-389.
- [16] Haveliwala T H. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4):784-796.
- [17] 冯浩, 黄坤, 李晶, 等. 基于深度学习的混合兴趣点推荐算法[J]. 电子与信息学报, 2019, 41(4): 880-887.
- [18] 刘帅, 刘婵娟. 音乐推荐系统综述[J]. 广州大学学报(自然科学版), 2020, 19(5):36-46, 77.
- [19] Zhou W, Han W B. Personalized recommendation via user preference matching[J]. Information Processing & Management, 2019, 56(3): 955-968.
- [20] 朱郁筱, 吕琳媛. 推荐系统评价指标综述[J]. 电子科技大学学报, 2012, 41(2):163-175.