融合似然比相似度的协同过滤推荐算法研究

王嵘冰,徐红艳,冯 勇,郭 浩

(辽宁大学 信息学院,沈阳 110036)
E-mail: fengyong@lnu.edu.cn

摘 要: 在基于用户的协同过滤推荐算法中,用户相似度计算准确与否直接影响推荐系统的质量. 目前,传统的相似度计算方法虽广泛使用,但仍存在较大的局限性,尤其在数据稀疏的情况下很难准确计算出用户相似度,容易出现过分放大或缩小的歧变,从而影响推荐算法的运行. 因此,本文使用似然比相似度并结合欧几里得距离加以调整的方法计算用户的相似度,藉此解决推荐系统中在每个用户只有少量评分的情况下计算两个用户间相似度的问题. 最后,在 MovieLens 数据集上,将本文所提计算方法与其他传统计算方法应用到同一基于用户的协同过滤推荐算法中进行对比实验,结果表明,本文所提方法能够更加准确、有效地识别相似用户,从而提高了推荐的准确性.

关 键 词: 个性化推荐; 协同过滤; 相似度; 似然比; 数据稀疏
中图分类号: TP311 **文献标识码:** A **文 章 编 号:** 1000-1220(2018)07-1478-04

Research on Collaborative Filtering Recommendation Algorithm Fused with Likelihood Ratio Similarity

WANG Rong-bing, XU Hong-yan, FENG Yong, GUO Hao
(School of Information, Liaoning University, Shenyang 110036, China)

Abstract: Due to the user-based collaborative filtering recommendation algorithm, the accuracy of user similarity calculation has a direct impact on the quality of recommender system. So far, the traditional method of similarity computation has been widely used. But there are still many limitations, especially in the case of sparse data, the similarity of users is difficult to be calculated, even cannot be calculated, as the result is prone to be magnified or minified, thus affecting the accuracy of the recommendation algorithm. In this paper, we propose a method of using likelihood ratio similarity, adjusted with Euclidean distance, to solve the problem of similarity computation between two users in the case that only a small amount of scoring is available each user. Finally, the proposed method and other traditional methods are applied to the same user-based collaborative filtering algorithm for comparative tests with MovieLens data sets. The result shows that the proposed method can identify similar users more accurately and effectively, thus improving the accuracy of the recommendation results.

Key words: personalized recommendation; collaborative filtering; similarity; likelihood ratio; data sparseness

1 引 言

互联网技术的飞速发展,人们在享受丰富的网络资源和服务的同时也不得不忍受信息过载的困扰. 个性化推荐系统被认为是当前解决信息过载问题的有效方法[1]. 协同过滤推荐是目前应为最为广泛的推荐方法,分为基于用户的协同过滤推荐和基于项目的协同过滤推荐,该方法的核心步骤是通过计算用户与用户之间或者项目和项目之间的相似度得分来对未来用户的首选项进行预测[2].

在基于用户的协同过滤推荐系统中,推荐算法依赖于用户相似度的计算,因此对用户相似度计算方法的改进就成了提高推荐算法准确度的有效途径之一[3]. 皮尔逊相关系数、余弦相似度、Jaccard 系数及巴氏距离等都是目前广泛使用的计算相似度的方法,其实用性已得到验证,但局限性也逐渐暴露出来,特别是在数据稀疏的情况下无法准确衡量用户的相似度,这在很大程度上影响了推荐算法的准确度[4].

2 相关工作

2.1 传统相似度计算方法

对于用户相似度计算方法的研究,目前使用的计算方法均依据待测两个向量的距离,如果两个向量的差距很小,则意味着这两个向量很相似[5]. 对于用户相似度计算来说,基本思路是首先找到两个用户 u,v 共同评分项目的评分向量 V_u,V_v,然后通过计算向量间的相似度作为用户 u,v 间的相似度[6].

2.2 传统方法不足分析

虽然传统计算方法在协同过滤推荐算法中取得了很大的成功,但仍然存在着一些局限之处,其中最突出的就是在数据

收稿日期: 2017-09-19 收修改稿日期: 2017-09-12 基金项目: 辽宁省博士科研启动基金项目(201601099)资助; 2016 年省级本科教改立项一般项目(201607)资助. 作者简介: 王嵘冰,男,1979 年生,博士,讲师,CCF 会员,研究方向为云计算、大数据、个性化推荐等; 徐红艳,女,1972 年生,硕士,副教授,研究方向为数据挖掘、Deep Web; 冯 勇(通信作者),男,1973 年生,博士,教授,CCF 会员,研究方向为数据挖掘、个性化推荐; 郭 浩,男,1990 年生,硕士研究生,研究方向为个性化推荐.

稀疏的状况下其结果容易出现过分放大或缩小,甚至无法计算的情况[7,8]. 例如,矩阵的稀疏程度为 90%,这意味着可参与计算的评分数据只有 10%,假设每个用户平均评分数据有 5 个,则在这种情况下每个用户可用于相似度计算的评分数据只有 0.5 个,这就导致在很多情况下用户之间的相似度是无法计算的或者计算不准确[9]. 下面简要分析传统相似度计算方法的不足与局限之处:

1) 当所处理数据集的稀疏程度较高时,由于用户之间共同评过分的项目数量不足,这必然会导致在用户相似度计算上的不准确.

2) 在用户之间共同评过分项目的数量只有 1 个的情况下,Jaccard 相关系数尽管是可以计算出相关结果的,但结果很难令人信服; 而余弦相似度在该情况下的计算结果显示总为 1.

3) 当两个用户的评分向量在每个维度上的取值都是相同的情况下,例如{1,1,1},{2,2,2}和{5,5,5},皮尔逊相关系数由于减去评分平均值之后其计算公式的分母为 0,所以出现无法计算的情况; 而余弦相似度的计算结果显示总为 1[10].

4) 在不同取值的情况下,皮尔逊相关系数与余弦相似度计算的结果往往会出现很大偏差. 例如,当两个用户评分向量分别为{1,0,5}和{5,4,3}时,由皮尔逊相关系数计算出来代表相似度的值是很大的,但从现实情况来看两个用户的相似度很低; 相反地,当两个用户评分向量依次为{4,5,3}和{5,4,5}时,由皮尔逊相关系数计算出来的代表相似度的值是很小的,但从现实情况看来两个用户的相似度很高; 当用户 u、v、w 的评分向量分别为{2,2,2}、{5,5,5}、{1,2,2}时,肉眼观察显然用户 u 和 w 很相似,但根据余弦相似度计算显示,结果却表明 u 与 v 更相似[11].

3　似然比相似度

3.1　计算思想

融合似然比的思想是受到在遗传图谱计算中广泛使用的 LOD 值和社区检测中得出的模块化概念的启发[12]. 在这两种情况下,相似度的概念是基于假设存在某种潜在数据结构,在此假设之上现有的(即已评分的、未缺失的)数据在这样数据结构上的分布中取得某个值的概率与此数据在概率随机试验中出现的概率的比值. 在遗传图谱的计算中,LOD 值表示两个遗传位点连锁的概率与不连锁的概率的比值的常用对数值[13]. 纽曼[13]在社区结构领域引入了这种概念: 如果社交网络顶点之间的边是随机生成的,一个社区结构包含了比预期更多的边缘,也就是包括了更多的类簇. 这些想法延伸到推荐系统领域,可以有效弥补传统计算方法的不足,藉此提出了似然比相似度.

3.2　似然比相似度定义

根据上面所介绍的 LOD 值的相关思想,本文给出似然比相似度的定义如下:

定义 1. 对于两个分别独立赋值的用户评分向量 $x_u=\{x_{u1},x_{u2},\cdots,x_{ui}\}$ 和 $x_v=\{x_{v1},x_{v2},\cdots,x_{vi}\}$,$x_{ui}$和 x_{vi}分别表示用户 u 和用户 v 对项目的评分,它们的似然比相似度(Likelihood Ratio Similarity,LRS)按照公式(1)方式定义:

$$LRS(x_u,x_v)=\log_{10}\frac{p(differences\ in\ x_u\ and\ x_v\mid same\ cluster)}{p(differences\ in\ x_u\ and\ x_v\mid pure\ chance)} \quad (1)$$

式(1)中的分子表示假设评分向量 x_u 和 x_v 在所定义的簇模型中属于同一簇的条件下,评分向量 x_u 和 x_v 中的每一对对应评分值取值之差出现的条件概率; 分母表示评分向量 x_u 和 x_v 中的每一个值在随机产生的情况下,每一对评分对应值取值之差出现的概率.

评分向量中每个值的取值只能是离散值 $V=\{1,2,\cdots,d\}$ 中的一个数字. 那么,就可以简单的计算出 x_{ui}和 x_{vi}在纯粹的随机试验的条件下,而且都未缺失有值的情况下,x_{ui}和 x_{vi}差值出现的概率. 例如在此条件下 $x_{ui}=x_{vi}$,这个概率为 $1/d^2$. 因为,在上述条件下,在指定的项目 i 上,两个用户的评分差为 0 的概率为 $p(|x_{ui}-x_{vi}|=0)=d/d^2=1/d$. 同理,可以推出出现其他差值的概率($p|x_{ui}-x_{vi}|=\delta$),其中 δ 为 1,2,…,d-1.

综上所述,似然比相似度公式中分母的定义如公式(2)所示:

$$p(differences\ in\ x_u\ and\ x_v\mid pure\ chance)=\prod_{\delta=0}^{d-1}b_\delta^{\#\delta} \quad (2)$$

其中,$b_\delta=p(|x_{ui}-x_{vi}|=\delta)$,$x_{ui}$和 x_{vi}是随机、独立产生的. $\#\delta$ 表示差值 δ 出现的次数.

计算的难点在于如何定义在假设 x_u 和 x_v 属于同一簇的情况下,x_{ui}和 x_{vi}取值之差为 δ 的条件概率. 根据 LOD 值思想和社区检测中的模块化概念,在推荐系统中有以下两个可信的假设:

1) 在推荐系统数据中存在着一个潜在的簇结构模型: 在推荐系统数据中有很多簇 $C_1,C_2,\cdots,C_k$,并且每个用户 u 都至少属于一个簇 C_c.

2) 用户对同一项目评分差的概率分布是固定在一个簇上的.

将上述假设总结归纳就是相似用户的评分是相似的.

根据以上合理的假设,定义评分之差 $|x_{ui}-x_{vi}|$ 的概率分布如公式(3)所示:

$$c_\delta=p(|x_{ui}-x_{vi}|=\delta\mid same\ cluster)=\left(\frac{1}{2}\right)^{\delta+1} \quad (3)$$

为了保证一个合理的概率分布,所以评分差为 d-1 时计算如公式(4)所示:

$$c_{d-1}=p(|x_{ui}-x_{vi}|=d-1)=1-\sum_{\delta=0}^{d-1}c_\delta=\frac{1}{2^{\delta-1}} \quad (4)$$

因此,似然比相似度公式中分子的定义如公式(5)所示:

$$p(differences\ in\ x_u\ and\ x_v\mid same\ cluster)=\sum_{\delta=0}^{d-1}c_\delta^{\#\delta} \quad (5)$$

其中,c_δ 与$\#\delta$ 参照上文中的定义; 如果用户 u 对项目 i 有评分,则 $x_{ui}=r_{ui}$.

本文强调 x_u 和 x_v 可能会存在很多缺失值,这些缺失值在余弦相似度和皮尔逊相关系数计算时被简单的看成是 0,本文对这些缺失值在计算过程中是不考虑在内的. 另一方面,只要 $1/2>1/d$,LRS 值会随着共同评分项目数量的增多而增大,而且一般而言,评分差值对 LRS 值的作用取决于离散评分值的最大值 d. 例如,当 $d=5$ 时,$b_1>c_1$; 但是当 $d=10$ 时,$b_1<c_1$; 因此,评分范围在 1 至 5 之间时,评分差为 1 的出现会减小 LRS 值,反之评分范围在 1 至 10 之间时,评分差为 1 的出现会增大 LRS 值.

综上所述,可以把 LRS 值改写成为如公式(6):

$$LRS(x_u,x_v) = \sum_{\delta=0}^{d-1} \#\delta \log_{10}\left(\frac{c_\delta}{b_\delta}\right) \quad (6)$$

其中,$\log_{10}(c_\delta/b_\delta)$ 是在有共同评分的项目 i 下,评分 x_{ui} 和 x_{vi} 差的绝对值为 δ 时对 LRS 值贡献的数值.

特别注意的是,似然比相似度的最大值是在两个评分向量都不缺失数据且完全相同的条件下取得的. 但是相似度是按照 $O(n\log_{10}d)$ 增长的,n 代表的是输入向量的维度,d 是此离散评分值的数量.

似然比相似度是负数的情况代表着数据更可能是随机巧合的相似情况,而不是在本文所陈述的用户数据簇模型的基础上出自同一簇的可能性.

3.3 混合相似度

在日常使用中,一般习惯于将相似度与 1 比较,越接近 1,相似度就越高. 所以在此对似然比相似度进行归一化处理. 本文使用反正切函数进行归一化处理,处理如公式(7)所示:

$$Lsim(x_u,x_v) = \frac{arctan(LRS(x_u,x_v)) \times 2}{\pi} \quad (7)$$

由于 LRS 考虑的是一个概率上的相似度,没有把评分差异纳入相似度的计算中,故在此基础上考虑评分之间差异的相似度,需要加入欧几里得距离作为另一半相似度.

欧几里得距离(Euclidean Distance)是一个经常使用的距离上的定义,表示在多维坐标空间中两个点之间的真实距离,或者是所表示向量的自然长度(即该点到原点的距离). 公式定义如(8)所示:

$$d(x_u,x_v) = \sqrt{\sum_{i=1}^{n}(x_{u_i}-x_{v_i})^2} \quad (8)$$

计算出来的欧几里德距离是一个大于 0 的数,为了使其更明显地体现用户之间的相似度,可以把它规约到(0,1]之间,就形成了基于欧氏距离的相似度,所以归一化处理如公式(9)所示:

$$Esim(x_u,x_v) = \frac{1}{1+d(x_u,x_v)} \quad (9)$$

综合前文所述,本文最终使用用户混合相似度作为最终的用户相似度的计算公式,定义如公式(10)所示,所占权重比例是将由实验得到最优值.

$$LEsim(x_u,x_v) = \varphi \times Lsim(x_u,x_v) + (1-\varphi) \times Esim(x_u,x_v) \quad (10)$$

4 实 验

4.1 实验环境与评价标准

本文实验环境配置: Windows7 操作系统,CPU i5-4460、3.20GHz,内存 2G 或以上,可用硬盘空间 50G 以上. 算法采用 Java 语言编写,对数据集直接进行文本提取. 实验用到的对比算法为 Apache mahout 框架所封装的基本推荐算法.

本文使用数据集 MovieLens 来评估本文所使用的推荐算法的性能,该数据集包括 943 位用户,1682 部电影和 100000 条评分记录,其中每个注册用户必须至少对 20 部电影进行评分,评分范围{1,2,3,4,5},评分数值越大,则表示该用户对该项目越喜欢. 用户-项目评分矩阵的稀疏度为

$$1-100000/(943\times1682) = 0.93695$$ [14].

本文实验的评价指标: 平均绝对误差 *MAE*(Mean Absolute Error)和均方根误差 *RMSE*(Root Mean Squared Error)[15],根据它们的值来验证本文所提相似度计算方法所得的预测结果的优势.

MAE 计算如公式(11)所示:

$$MAE = \frac{\sum_{u,i\in T}|r_{ui}-pre_{ui}|}{|T|} \quad (11)$$

其中,r_{ui} 表示用户 u 对项目 i 的实际评分,pre_{ui} 表示用户 u 对项目 i 的预测评分. T 为测试集,$|T|$ 表示测试集中元素的个数. *MAE* 越小,说明预测值与实际值越接近,预测结果就越准确.

RMSE 计算如公式(12)所示:

$$RMSE = \sqrt{\frac{\sum_{u,i\in T}(r_{ui}-pre_{ui})^2}{|T|}} \quad (12)$$

同样,*RMSE* 值越小,表示预测值与评分真实值越接近,预测效果越好.

4.2 参数 φ 的最优值确定

参数 φ 用来表示似然比相似度和欧几里得距离相似度在最终混合相似度计算中所占比例,实验在基于用户的协同过滤算法中进行,选取 φ 为不同值,从 0 到 1.0,并调整近邻用户集大小 N 的值进行多次实验,以排除偶然性,得到使算法效果达到最好的参数值. 相似度比例参数 φ 对 *MAE* 值和 *RMSE* 值的影响如图 1 和图 2 所示.

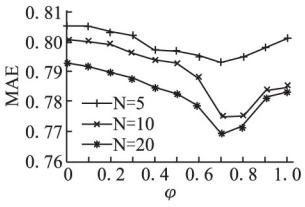

图 1 参数 φ 对 *MAE* 值的影响

Fig. 1 Effect of parameter φ on *MAE*

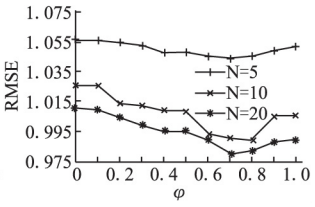

图 2 参数 φ 对 *RMSE* 值的影响

Fig. 2 Effect of parameter φ on *RMSE*

4.3 预测准确度对比实验

为了验证本文所提相似度计算方法优于现有的相似度计算方法,对比实验将三种传统相似度的计算方法皮尔逊相关系数(pearson)、余弦相似度(cosine)、巴氏距离(bhattacharyya)及本文算法(LEsim)应用到同一个基于用户的协同过滤算法中,通过该算法的预测准确度来衡量相似度计算方法的优劣. 在 MovieLens 数据集中,首先按照各种相似度算法进行近邻用户集的筛选,然后根据近邻用户集的评分数据进行预测评分,最后计算得出 *MAE* 和 *RMSE* 值并进行比较. 其

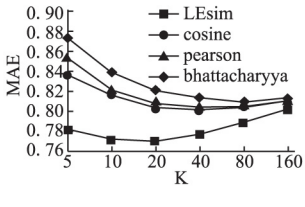

图 3 准确度在 *MAE* 值上的对比(原始数据集)

Fig. 3 Accuracy comparison on *MAE*(original data set)

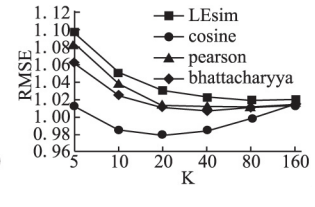

图 4 准确度在 *RMSE* 值上的对比(原始数据集)

Fig. 4 Accuracy comparison on *RMSE*(original data set)

中,对当前目标用户的近邻用户集中的用户个数分别选取为 5,10,20,…,160,进行多次实验,以排除偶然因素. 各种相似度预测准确度的比较如图 3 和图 4 所示.

4.4　不同稀疏性对比实验

为了检验在不同稀疏程度的数据集下本文所提出的相似度计算方法的预测性能,本文设置如下对比实验: 从原始 MovieLens 数据集删除部分数据,使其稀疏度达到 0.99,算法预测准确度的比较如图 5 和图 6 所示.

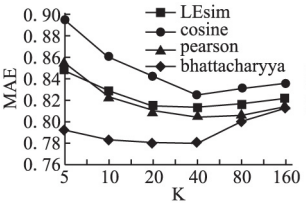

图 5　准确度在 *MAE* 值上的对比(稀疏数据集)
Fig. 5　Accuracy comparison on *MAE*(sparse data set)

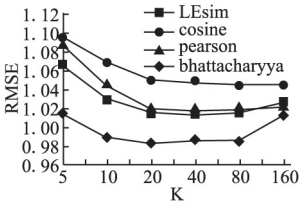

图 6　准确度在 *RMSE* 值上的对比(稀疏数据集)
Fig. 6　Accuracy comparison on *RMSE*(sparse data set)

4.5　实验结果分析

由图 1、图 2 可以看出,在近邻用户集大小不变的情况下,在参数取 0.7 时本文所提的相似度计算方法的效果达到最佳; 通过调整近邻用户集的大小进行多次实验排除偶然因素,得到同样的效果. 由图 3、图 4 可以看出,总体上由 *LEsim* 计算所得到的 *MAE* 和 *RMSE* 值都比其他三种相似度计算方法要小,也就是说 *LEsim* 的表现更优秀,预测更准确. 单独来看每条折线的情况,随着近邻用户集的逐渐增大,*MAE* 和 *RMSE* 值都趋于平稳,*LEsim* 的变化趋势较其他方法来说较为平缓,说明 *LEsim* 的表现相对稳定. 图 5、图 6 说明本文所提方法在数据稀疏的情况下表现依旧良好.

5　结束语

本文介绍了似然比相似度的计算方法,并将它应用于个性化推荐领域中. 该方法适合于离散的、稀疏的、高维的数据环境. 并且在真实的数据集中通过实验表明似然比相似度可以很好地衡量用户之间的相似度,它的性能优于传统计算方法. 未来的研究重点是探索在推荐系统数据中如何设计一个更好的聚类结构模型,以提高协同过滤算法预测准确度. 另一个可能的研究方向是开发快速聚类方法,使用似然比相似度,以提高基于用户的协同过滤推荐算法的可扩展性.

References:

[1] Meng Xiang-wu, Liu Shu-dong, Zhang Yu-jie, et al. Research on social recommender systems [J]. Journal of Software, 2015, 26(6): 1356-1372.

[2] Leng Ya-jun, Lu Qing, Liang Chang-yong. Survey of recommendation based on collaborative filtering [J]. Pattern Recognition and Artificial Intelligence, 2014, 27(8): 720-734.

[3] 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.

[4] Fan Yong-quan, Du Ya-jun. Improved user-based collaborative filtering method based on weighted similarity [J]. Computer Engineering and Applications, 2016, 52(22): 222-225.

[5] Desrosiers C, Karypis G. A comprehensive survey of neighborhood-based recommendation methods [M]. Recommender Systems Handbook, USA: Springer US, 2011.

[6] Herlocker J, Konstan J A, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms [J]. Information Retrieval Journal, 2002, 5(4): 287-310.

[7] Patra B K, Launonen R, Ollikainen V, et al. A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data [J]. Knowledge-Based Systems, 2015, 82(7): 163-177.

[8] Ahn H J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem [J]. Information Sciences, 2008, 178(1): 37-51.

[9] Wang Cheng, Zhu Zhi-gang, Zhang Yu-xia, et al. Improvement in recommendation efficiency and personalized of user-based collaborative filtering algorithm [J]. Journal of Chinese Computer Systems, 2016, 37(3): 428-432.

[10] Liu Qing-wen. Research on recommender systems based on collaborative filtering [D]. Hefei: University of Science and Technology of China, 2013.

[11] Ye Wei-gen. Research on personalized recommendation algorithms based on collaborative filtering [D]. Wuxi: Jiangnan University, 2016.

[12] Yu Jin-ming, Meng Jun, Wu Qiu-feng. Item collaborative filtering recommendation algorithm based on improved similarity measure [J]. Journal of Computer Applications, 2017, 37(5): 1387-1391, 1406.

[13] Newman M E J. Modularity and community structure in networks [J]. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577-8582.

[14] Lan Yan, Cao Fang-fang. Research of time weighted collaborative filtering algorithm in movie recommendation [J]. Computer Science, 2017, 44(4): 295-301, 322.

[15] Liu Zhu-song, Ou Shi-hua, Huang Shu-qiang. Collaborative filtering recommendation algorithm based on asymmetric weighted user similarity [J]. Journal of Chinese Computer Systems, 2017, 38(4): 721-725.

附中文参考文献:

[1] 孟祥武,刘树栋,张玉洁,等. 社会化推荐系统研究 [J]. 软件学报,2015,26(6): 1356-1372.

[2] 冷亚军,陆　青,梁昌勇. 协同过滤推荐技术综述 [J]. 模式识别与人工智能,2014,27(8): 720-734.

[4] 范永全,杜亚军. 基于加权相似度的用户协同过滤方法 [J]. 计算机工程与应用,2016,52(22): 222-225.

[9] 王　成,朱志刚,张玉侠,等. 基于用户的协同过滤算法的推荐效率和个性化改进 [J]. 小型微型计算机系统,2016,37(3): 428-432.

[10] 刘青文. 基于协同过滤的推荐算法研究 [D]. 合肥: 中国科学技术大学,2013.

[11] 叶卫根. 基于协同过滤的个性化推荐算法研究 [D]. 无锡: 江南大学,2016.

[12] 于金明,孟　军,吴秋峰. 基于改进相似性度量的项目协同过滤推荐算法 [J]. 计算机应用,2017,37(5): 1387-1391,1406.

[14] 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): 295-301,322.

[15] 刘竹松,欧仕华,黄书强. 基于非对称加权相似度的协同过滤推荐算法 [J]. 小型微型计算机系统,2017,38(4): 721-725.