山东农业大学学报(自然科学版),2016,47(6):940-944 VOL.47 NO.6 2016
Journal of Shandong Agricultural University (Natural Science Edition) doi:10.3969/j.issn.1000-2324.2016.06.026

改进 Pearson 相关系数的个性化推荐算法

陈功平,王 红

六安职业技术学院 信息与电子工程学院, 安徽 六安 237158

摘 要: 基于用户的协同过滤推荐算法(User CF)从用户的历史操作记录中分析用户的兴趣，找到每个用户的 k 个相似近邻，然后基于这 k 个近邻集合实施推荐。皮尔森相关系数能够根据用户的历史评分计算用户间的相似度。本文加入流行项目惩罚系数、共同评分项目惩罚系数δ和评分差异惩罚系数λ，对皮尔森相关系数实施了改进和修订。实验结果表明，改进后的皮尔森相似度的推荐效果好于原始皮尔森相似度。

关键词: 个性化推荐; 相似性计算; 皮尔森相关系数; 评分预测

中图法分类号: TP301 **文献标识码**: A **文章编号**: 1000-2324(2016)06-0940-05

A Personalized Recommendation Algorithm on Improving Pearson Correlation Coefficient

CHEN Gong-ping, WANG Hong

College of Information and Electronic Engineering/Lu'an Vocation Technology College, Luan 237158, *China*

Abstract: This paper found k similar neighbors of each user by analyzing the users' interests from their operating records based on the collaborative filtering recommendation and then made the implementation of recommendations based on the k similar neighbors. The Pearson Correlation Coefficient could calculate the similarity among users. This paper added popular items penalty coefficient, simultaneous rating items penalty coefficient δ and rating different penalty coefficient λ to the Pearson Correlation Coefficient and carried out the improvement and revised to the Pearson Correlation Coefficient. The experimental results indicated that the recommendation effect of improved Pearson Correlation Coefficient was better than the original one.

Keywords: Personalized recommendation; similarity calculation; Pearson Correlation Coefficient; rating prediction

个性化推荐[1]比大众化推荐更能满足当前用户的需要，如同静态网页为用户显示相同内容，动态网页为用户显示个性化的内容。个性化推荐[2]算法从用户的历史行为中研究用户兴趣，根据每个用户的喜好生成推荐列表。

协同过滤推荐（Collaborative Filtering，CF）算法[3]是推荐系统中应用广泛、研究深入又简单的推荐算法，有基于用户相似性的协同推荐算法（UserCF）[4]和基于项目相似性的协同推荐算法（ItemCF）[5]，UserCF 从用户（User）对项目（Item）的评分矩阵中挖掘当前用户最相似的 k 个邻居，然后以 k 个邻居为基础实施推荐。本文采用改进的皮尔森相关系数计算用户间的相似度，降低共同评分项目过少、流行项目及评分差异三种因素对用户相似度的影响，改进相似度计算方法，提高评分预测的精度。实验结果表明，改进后的相似度计算方法的推荐效果好于原始相似度计算的推荐效果。

1 用户相似度评价指标

用户或项目的相似度计算是协同过滤推荐算法的核心[6]。计算用户 u_1 与 u_2 的相似度时，从评分矩阵中找出两个用户的共同评价项目集合，记作 V_{u1}、V_{u2}，将向量集合 V_{u1} 与 V_{u2} 的相似度作为用户 u_1 与 u_2 的相似度。常用的相似度度量方法有 Jaccard 相关系数、余弦相似度和皮尔森相关系数。

1.1 Jaccard 相关系数

Jaccard[7]相关系数用两个集合的并集和交集的比值来度量用户相似度，即。

$$Sim(u1,u2) = Jaccard(V_{u1},V_{u2}) = \frac{|V_{u1} \cap V_{u2}|}{|V_{u1} \cup V_{u2}|} \quad (1)$$

收稿日期: 2015-03-31 **修回日期**: 2015-08-31

基金项目: 2015 年度安徽高校自然科学研究重点项目(KJ2015A435); 安徽省 2016 年高校优秀青年人才支持计划重点项目(gxyqZD2016570);安徽省 2014 年高校优秀青年人才支持计划

作者简介: 陈功平(1980-),男,讲师,研究方向:个性化推荐，计算机网络. E-mail:wh0115140@126.com

***通讯作者**: Author for correspondence. E-mail:wh0115140@126.com

数字优先出版:2016-12-26 http://www.cnki.net

从计算公式可以看出，Jaccard 相关系数适用于计算离散型集合的相似度，若评分矩阵中各元素的值只用 1（喜欢）、-1（不喜欢）和 0（无关）表示，使用 Jaccard 相关系数计算用户相似度的效果较好。而对于非离散型的评分矩阵，因 Jaccard 相关系数没有考虑评分值对相似度的影响，因此对于 5 星或 10 级评分矩阵的相似度计算效果较差。

1.2　余弦相似度

余弦相似度[8]通过计算两个向量间的夹角余弦度值来衡量两个用户间的相似性，即。

$$Sim(u_1,u_2)=cos(V_{u1},V_{u2})=\frac{|V_{u1}\bigcap V_{u2}|}{\sqrt{|V_{u1}||V_{u2}|}} \tag{2}$$

从公式可以看出，两个向量间的夹角越小则相似度越大，若夹角为 0，认为 u_1 和 u_2 完全相似，即相似度值为 1，这种计算方法忽视了向量间的距离。如图 1 中的 u_1、u_2、u_3，根据余弦相似度计算 u_1 和 u_2 的相似度为 1，大于 u_1 和 u_3 的相似度，但在空间上，u_1 和 u_3 的距离更接近。余弦相似度同样没有考虑评分矩阵中的评分值对相似度的影响。

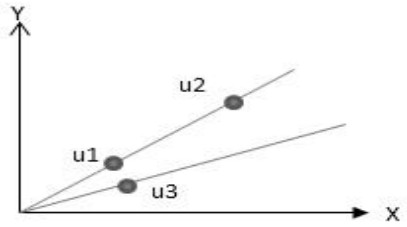

图 1 余弦相似度示意图
Fig.1 Sketch of cosine similarity

1.3　皮尔森相关系数

皮尔森相关系数[9]利用向量间的线性相关性表示用户相似度，即。

$$Sim(u1,u2)=\frac{\sum_{i\in V_{u1}\cap V_{u2}}(r_{u1,i}-\overline{r_{u1}})(r_{u2,i}-\overline{r_{u2}})}{\sqrt{\sum_{i\in V_{u1}\cap V_{u2}}(r_{u1}-\overline{r_{u1}})^2\sum_{i\in IV_{u1}\cap V_{u2}}(r_{u2}-\overline{r_{u2}})^2}} \tag{3}$$

其中，$r_{u1,i}$ 表示用户 u_1 对项目 i 的评分，$\overline{r}_{u1}$ 表示用户 u_1 对所有项目评分的平均值，皮尔森相似度的取值范围从[-1，1]，比 Jaccard 相关系数和余弦相似度的计算结果更精确[10]。

皮尔森相似度形式上和余弦相似度相似，计算时减去了用户的平均评价值，就是对余弦相似度做了一次归一化处理，统一了用户的评分标准。

2　皮尔森相关系数的改进

假如 u_1 和 u_2 用户共同评分项目较少且恰好满足公式（3）相似度为 1 的条件，这种情况在实际评分矩阵中非常常见，导致每个用户相似度为 1 的近邻的相同评价项目数较少，近邻计算结果与实际不符，因此需要改进原始皮尔森相似度的计算方式，加入限制条件。

2.1　热门项目惩罚

当某首歌曲流行时，几乎所有用户都会收听，所以热门项目对相似度计算的贡献较小[11]。这里将评分矩阵中评价数量作为项目热门的度量标准，为了降低热门项目对用户相似度的影响，将公式（3）修订为。

$$Sim^1(u1,u2)=\frac{\sum_{i\in V_{u1}\cap V_{u2}}\frac{1}{\log(1+N(i))}(r_{u1,i}-\overline{r_{u1}})(r_{u2,i}-\overline{r_{u2}})}{\sqrt{\sum_{i\in V_{u1}\cap V_{u2}}(r_{u1,i}-\overline{r_{u1}})^2\sum_{i\in IV_{u1}\cap V_{u2}}(r_{u2,i}-\overline{r_{u2}})^2}} \tag{4}$$

其中，$N(i)$表示项目 i 在评分矩阵中被评价的次数。

2.2　共项评分项目惩罚

为了降低偶尔相同评价对皮尔森相似度的影响，设置共同评价项目过少的惩罚阀值δ，将公式（4）改进为。

$$Sim^2(u1,u2)=\frac{min(|N(u1)\bigcap N(u2)|,\delta)}{\delta}\cdot Sim^1(u1,u2) \tag{5}$$

δ的取值太小则惩罚不明显，当取值达到一定限度时推荐效果趋于收敛。图 2 反应了δ值对推荐结果的影响，数据取自近邻数为 100 个、MovieLens 100 k 数据集中测试集的评分预测的 RMSE 值变化，当δ在 25 以后趋于收敛。从图 2 可以看出，δ可以有效提高评分预测的准确度。

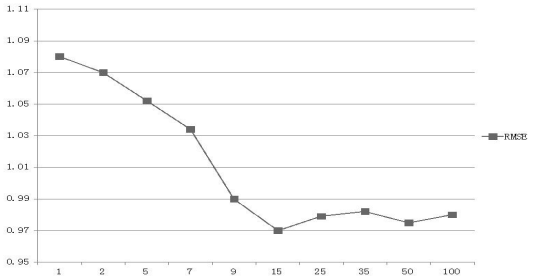

图 2 δ对 RMSE 值的影响

Fig.2 The influence of δ on RMSE value

2.3　共同评分差异性修正

皮尔森相似度已经对评分做了一次归一化处理，但如果 u_1 和 u_2 用户的平均分相似，而对某项目打分差值超过一定界限，认为这样的共同评分项对相似度贡献较小，加入共同评分差异性修正值λ。

$$Sim^3(u1,u2)=\frac{min(|N(u1)\bigcap N(u2)|,\delta)}{\delta}*\frac{\sum_{i\in V_{u1}\cap V_{u2}}\frac{1}{\log(1+N(i))}(\lambda(r_{u1}-\overline{r_{u1}})(r_{u2}-\overline{r_{u2}}))}{\sqrt{\sum_{i\in V_{u1}\cap V_{u2}}(r_{u1}-\overline{r_{u1}})^2\sum_{i\in IV_{u1}\cap V_{u2}}(r_{u2}-\overline{r_{u2}})^2}} \tag{6}$$

其中，$\lambda=\begin{cases}1, & |r_{u1}\text{-}r_{u2}|<r\ or\ (|r_{u1,i}\text{-}r_{u2,i}|>=r\ \ and\ \ |\overline{r_{u1}}-\overline{r_{u2}}|>\varepsilon)\\ (0,1), & |r_{u1,i}\text{-}r_{u2,i}|>=r\ \ and\ \ |\overline{r_{u1}}-\overline{r_{u2}}|<\varepsilon\end{cases}$

r 用来限制 u_1、u_2 对项目 i 评价差值的界限，｜r_{u1}-r_{u2}｜越小，即用户 u_1 和 u_2 对项目 i 的评分越接近，ε用于判定 u_1、u_2 平均分相似性的界限，修正值λ小于 1，用于降低用户平均分相似而评分差值大对相似度的影响。

3　基于皮尔森相似度的个性化推荐算法

3.1　基于用户相似度的评分预测

用户会为喜爱的项目打高分，所以评分预测是推荐系统的实现原理之一。得到用户 u 的 k 个近邻后就可以预测 u 对项目 i 的评分，预测公式如（7）所示。

$$\hat{r}_{ui}=\bar{r}_u+\frac{\sum_{v\in S(u,k)\bigcap N_u(i)}w_{uv}(r_{vi}-\bar{r}_v)}{\sum_{v\in S(u,k)\bigcap N_u(i)}|w_{uv}|} \tag{7}$$

其中，$\bar{r}_u$ 表示用户 u 所有评分的均值，$S(u,k)$表示和用户 u 兴趣最相似的 k 个用户集合，$N_u(i)$是对项目 i 有过评分的用户集合。

3.2　评价指标

使用均方根误差 RMSE[12]作为算法的评价指标，RMSE 反应的是预测分值与用户实际分值间的差异，值越小说明预测越准确[13]。

$$RMSE=\sqrt{\frac{\sum_{u,i\in T}(r_{ui}-\hat{r}_{ui})^2}{|T|}} \tag{8}$$

其中，r_{ui}表示用户 u 对项目 i 的实际评分，$\hat{r}_{ui}$ 表示用户 u 对项目 i 的预测评分，$|T|$ 表示预测数量。

3.3　用户相似度训练

算法：改进 Pearson 相关系数的用户近邻计算
输入：训练集评分矩阵 Train, k, δ, Max_δ, λ
输出：每个用户的 k 个最近邻矩阵 Sim
1：RMSE[u_1, u_2,…]中各元素初始值为 1
2：根据公式（3）计算用户间的皮尔森相似度并存入 Sim
3：k 个最近邻训练
3.1 根据公式（6）重新计算 Sim
3.2 根据公式（7）预测 Train 中 u_i 的评分
3.3 根据公式（8）计算 RMSE[u_i]值
3.4 δ++
3.5 直到 RMSE[u_i]<0.6 或δ>=Max_δ

4　实验结果及分析

使用 GroupLens 发布的 MovieLens 100 k 数据集验证实验效果，该数据集[14]存储了 10 万条用户对电影的评分记录，每条记录由（用户编号，电影编号，评分值）3 项组成，按照每个用户的评价数量，以 8:2 的比例随机分成训练集 Train（80 k 条记录）和测试集 Test（20 k 条记录），Movie Lens 数据集初始情况如表 1 所示。

4.1　实验参数设置

本文通过 4 组实验验证改进皮尔森相似度对推荐结果准确度的影响，每组实验任务名及参数设置如表 2 所示。

表 1 Movie Lens 数据集简介

Table 1 Introduction of Movie Lens data set

项目 Item	简介 Introduction
用户	943 个用户，id 用 1~943 表示
电影	1682 部电影，id 用 1~1682 表示
评分记录	10 万条

表 2 实验任务及参数设置

Table 2 Experimental tasks and parameters

Task	δ	Max_δ	r	ε	λ
Task1	0	0	5	5	1
Task2	3	10	0.5	2	0.7
Task3	3	15			
Task4	3	30			

Task1 的参数设置表示没有改进，相当于使用公式（3）为相似度计算方法，即原始皮尔森相似度系数，其余 3 组实验均对计算方法有所改进。

4.2　实验结果及分析

表 3 是 Train 集合的 RMSE 评价指标值。

从数据可见，在 k 值小于 200 的情况下，Task2~Task4 明显比 Task1 效果好，在 k 为 100 时，Task2~Task4 的预测效果最好；Task1 随 k 值的增加效果越来越好，k 值的增加会使得算法的时间复杂度升高；Task2~Task4 在 k 值相同时 RMSE 值差值很小。

表 4 是 Test 集合的 RMSE 评价指标值。

表 3 Train 集合的 RMSE 值
Table 3 RMSE values of Train

k	Task1	Task2	Task3	Task4
50	0.8999	0.637	0.636	0.637
100	0.797	0.613	0.613	0.613
150	0.739	0.625	0.623	0.625
200	0.723	0.640	0.637	0.639
300	0.701	0.670	0.669	0.670
400	0.687	0.692	0.693	0.694

表 4 Test 集合的 RMSE 值
Table 4 RMSE values of Test

k	Task1	Task2	Task3	Task4
50	1.102	1.029	1.028	1.028
100	1.076	0.977	0.977	0.976
150	1.022	0.952	0.952	0.952
200	0.982	0.945	0.943	0.944
300	0.963	0.941	0.939	0.939
400	0.948	0.937	0.935	0.937

在 Test 集合中，Task2~Task4 的预测效果好于 Task1，当 k=100 时，Task2 和 Task3 的 RMSE 指标值比 Task1 的指标值提高了 0.1。当 k 值升高，RMSE 值降低，即 k 值越大，预测评分越接近实际评分。

图 3 是 Train 集合的 RMSE 指标值与 k 值的关系图，对于传统的皮尔森系数而言，k 值越大预测越准确，而改进后的皮尔森相关系数的预测结果在 k=100 时出现拐点。

图 4 是 Test 集合的 RMSE 指标值与 k 值的关系图，与 Train 集合不同，二者都随着 k 值的升高而降低，改进相似度计算方法的测试效果好于传统计算方法。由图可以认为当 k 值超过一定限度后，改进相似度计算方法与传统相似度计算方法的预测结果的指标值越来越接近。

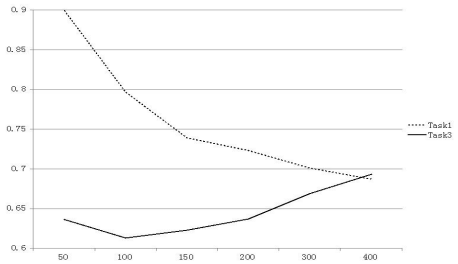

图 3 Train 集合的 RMSE 值比较
Fig.3 Comparison of RMSE values in Train

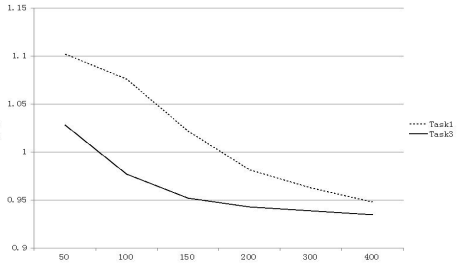

图 4 Test 集合的 RMSE 值比较
Fig.3 Comparison of RMSE values in Test

5 总 结

相似度计算方法是推荐算法的核心，皮尔森相关系数将用户评分数据参与到相似度计算过程，能够更准确的挖掘用户的兴趣，提高推荐效果。在实际应用时，由于用户数和项目数较大，相似度计算和推荐算法的时间复杂度过高，如果将相似度计算过程放在线下进行则不能实施实时推荐。降低相似度计算和推荐的时间复杂度、提供实时推荐是本文的下一个研究方向。

参考文献

[1] Zhuang HL, Tang J, Tang WB, *et al.* Actively learning to infer social ties[J]. Data Mining and Knowledge Discovery, 2012,25(2):270-297
[2] 高 明,金澈清,钱卫宁,等.面向微博系统的实时个性化推荐[J].计算机学报,2014,37(4):363-375
[3] Kaleli C. An entropy-based neighbor selection approach for collaborative filtering[J]. Knowledge-Based Systems,2013,56(3):273-280
[4] 吴 湖,王永吉,王 哲.两阶段联合聚类协同过滤算法[J].软件学报,2010,21(5):1042-1054
[5] 黄创光,印 鉴,汪 静,等.不确定近邻的协同过滤推荐算法[J].计算机学报,2010,33(8):1369-1377
[6] 刘树栋,孟祥武.基于位置的社会化网络推荐系统研究[J].计算机学报,2015,38(2):322-336
[7] Lu ML, Qin Z, Cao Y, *et al.* Scalable news recommendation using multi-dimensional similarity and Jaccard–Kmeans clustering[J]. The Journal of Systems and Software, 2014,95(4):242-251
[8] Aral S, Walker D. Identifying Influential and Susceptible Members of Social Networks[J]. Science, 2012,337(6092):337-341
[9] 张宇镭,党 琰,贺平安.利用 Pearson 相关系数定量分析生物亲缘关系[J].计算机工程与应用,2005(33):79-82,99
[10] Zhuang H, Savage EM. Variation and Pearson correlation coefficients of Warner-Bratzler shear force measurements within broiler breast fillets[J]. Poultry Science, 2009,88(1):214-20
[11] Jannach D, Zanker M, Felfemig A, *et al*. Recommender systems: an introduction[D]. Cambridge:CUP, 2010
[12] 陈克寒,韩盼盼,吴 健.基于用户聚类的异构社交网络推荐算法[J].计算机学报,2013,36(2):349-359
[13] 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
[14] 朱 夏,宋爱波,东 方,等.云计算环境下基于协同过滤的个性化推荐机制[J].计算机研究与发展,2014,51(10):2255-2269