计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(2):246−252 [doi: 10.15888/j.cnki.csa.006747]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

协同过滤算法中冷启动问题研究①

邵　煜, 谢颖华

(东华大学 信息科学与技术学院, 上海 201620)
通讯作者: 邵　煜, E-mail: yvonnesy0921@163.com

摘　要: 为了解决传统协同过滤算法的冷启动问题, 提高算法的推荐质量, 本文针对协同过滤算法中的冷启动问题进行研究, 提出了两种改进的算法. 新用户冷启动: 融合用户信息模型的基于用户的协同过滤算法; 新项目冷启动: 采用层次聚类的基于项目的协同过滤算法. 将新算法在网络开源数据集 MovieLens 上进行实验验证, 比较改进算法和传统算法在查全率和查准率上的差异, 结果表明改进算法能够有效地提高算法的推荐质量, 缓解新用户和新项目的冷启动问题.

关键词: 协同过滤; 冷启动; 用户特征; 层次聚类; 相似度

引用格式: 邵煜,谢颖华.协同过滤算法中冷启动问题研究.计算机系统应用,2019,28(2):246–252. http://www.c-s-a.org.cn/1003-3254/6747.html

Research on Cold-Start Problem of Collaborative Filtering Algorithm

SHAO Yu, XIE Ying-Hua

(School of Information Science and Technology, Donghua University, Shanghai 201620, China)

Abstract: In order to solve the cold-start problem of the traditional collaborative filtering algorithm and to improve the performance of recommendation, this study focuses on the cold-start problem and proposes two algorithms. Cold-start problem of new users: user-based collaborative filtering algorithm integrated with user's information model, cold-start problem of new items: item-based collaborative filtering algorithm applying hierarchical clustering. After a series of experiments carried out on public data sets—MovieLens, comparing the difference between the precision and recall value of the improved algorithm and the traditional one, the results show that the new algorithm can effectively alleviate the cold start problem and improve the quality of recommendation.

Key words: collaborative filtering; cold-start; user characteristic; hierarchical clustering; similarities

互联网的快速发展给人们的生活带来了极大的便利, 同时, 海量的信息数据引发了"信息超载"问题[1]. 个性化推荐系统应运而生, 通过对用户兴趣建模, 在海量的数据中找到合适的物品推荐给用户, 极大地提高了信息使用率[2].

推荐系统常用的推荐算法有: 基于内容的推荐、基于关联规则的推荐、基于知识的推荐以及协同过滤推荐[3−5]. 其中, 协同过滤推荐算法是应用最成功最广泛的一种算法.

协同过滤算法依然面临着诸多局限性, 包括数据稀疏性、冷启动、可扩展等问题, 影响了系统的推荐结果和推荐质量.

1　传统的协同过滤算法

根据用户行为数据设计的算法称为协同过滤算法[6]. 协同过滤算法建立在数据挖掘的基础上, 算法的工作原理是: 根据用户的历史评分计算相似度, 相似度高的判为目标用户的邻居用户, 依据邻居集用户的历史评

① 收稿时间: 2018-07-27; 修改时间: 2018-08-21; 采用时间: 2018-08-29; csa 在线出版时间: 2019-01-28

分计算目标用户的可能评分值, 将评分值高的前 N 项推荐给目标用户 (Top-N 推荐).

协同过滤算法可以分为基于用户的协同过滤 (user-based CF) 和基于项目的协同过滤 (item-based CF).

1.1 冷启动问题

协同过滤的冷启动问题分为新用户冷启动、新项目冷启动和新系统冷启动, 协同过滤算法的核心是分析用户-项目评分矩阵, 计算相似度. 当系统中有新用户加入时, 该用户在系统中不存在历史评分数据, 不能根据传统算法计算用户间的相似度, 也就无法为其进行推荐, 这就是协同过滤算法的新用户冷启动问题[7]; 同样地, 当系统中加入新项目时, 由于没有该项目的历史评分记录, 推荐算法难以将该项目推荐给用户, 这就是新项目的冷启动问题.

1.2 冷启动问题的研究现状

近年来, 许多专家学者对冷启动问题提出了一系列的解决办法, 比如众数法、平均值法、相似度度量法等等[8], 减少了冷启动问题对推荐质量的影响.

Xu JW, Yao Y 等提出了一种新型的评分比较策略 (RaPare)[9]学习冷启动用户/项目的潜在性能, 通过寻找冷启动用户/项目和现有用户/项目之间的差异, 为冷启动用户/项目的潜在性能提供细粒度校准. Nguyen VD, Sriboonchitta S, Huynh VN 引入了一种用户社交网络和软评分相结合的协同过滤推荐系统[10], 通过用户社交网络提取社区特征来解决冷启动问题. Katarya R, Verma OP 将计算相似性的不对称方法与矩阵分解和基于典型性的协同过滤 (Tyco) 相结合[11], 实现了一种改进的电影推荐算法. 改进算法采用了 Pearson 相关系数计算相似度, 用线性回归进行预测, 得到更好的推荐结果.

2 相似度的计算方法

2.1 传统的相似度算法

协同过滤算法根据相似度的计算结果找到目标用户/项目的邻居集, 因此相似度的计算方法很重要, 目前常见的相似度计算方法有: 余弦相似度、欧几里德距离和 Pearson 相关系数等等.

余弦相似度:

$$sim(u,j)=\frac{\sum_{i=1}^{n}u_i\cdot j_i}{\sqrt{\sum_{i=1}^{n}(u_i)^2}\cdot\sqrt{\sum_{i=1}^{n}(j_i)^2}} \tag{1}$$

欧几里德距离:

$$d(u,j)=\sqrt{\sum(u_i-j_i)^2} \tag{2}$$

$$sim(u,j)=\frac{1}{1+d(u,j)} \tag{3}$$

Pearson 相关系数:

$$sim(u,j)=\frac{\sum_{i\in I_{u,j}}^{n}\left(R_{u,i}-\overline{R_u}\right)\left(R_{j,i}-\overline{R_j}\right)}{\sqrt{\sum_{i\in I_{u,j}}^{n}\left(R_{u,i}-\overline{R_u}\right)^2}\sqrt{\sum_{i\in I_{u,j}}^{n}\left(R_{j,i}-\overline{R_j}\right)^2}} \tag{4}$$

其中, $R_{u,i}$, $R_{j,i}$分别代表用户 u 和用户 j 对项目 i 的评分, $\overline{R_u}$, $\overline{R_j}$分别代表用户 u 和 j 的评分平均值, $I_{u,j}$是用户 u,j 的共同评分项目集合.

2.2 改进后的算法

2.2.1 融合用户信息模型的基于用户的协同过滤算法

通常, 各推荐网站的数据库中会含有关于用户属性信息的数据集, 常见的用户属性包括: 用户 ID,性别, 年龄, 职业, 个人偏好等等. 某些网站在用户注册时会询问其可能喜欢的类型, 以此达到更优的推荐效果. 具有相同或者相近属性的两个用户会表现出更相近的兴趣爱好[12]. 因此, 本文以用户的个人信息属性为切入点, 为了解决新用户的冷启动问题, 通过分析用户的基本信息特征, 设计了一种融合用户信息模型的基于用户的协同过滤算法.

新算法综合考虑了用户的 k 项基本信息值$attr_i$ (i=1, 2, 3, ⋯, k), 分别给不同的属性信息分配权重λ_i, 计算用户之间的特征差, 用$attr(u,v)$表示, 计算方法如式 (5) 所示.

$$attr(u,v)=\sum_{i=1}^{k}\lambda_i\cdot attr_i \tag{5}$$

其中, λ_i满足:

$$\sum_{i=1}^{k}\lambda_i=1 \tag{6}$$

求得用户特征差值$attr(u,v)$后, 利用 Sigmoid 函数

(式 (7)), 计算用户 u 和用户 v 之间的用户特征信息相似度$sim_{attr}(u,v)$, 如式 (8) 所示.

$$s(x)=\frac{1}{1+e^{-x}} \tag{7}$$

$$sim_{attr}(u,v)=2\times\left(1-\frac{1}{1+\exp(-attr(u,v))}\right) \tag{8}$$

当用户间相同特征较多时, 量化后的特征差值较小, $attr(u,v)$值较小, $sim_{attr}(u,v)$值较大, 表现出用户较高的相似性, 并且由于$attr(u,v)$值恒为正, 确保$sim_{attr}(u,v)$取值在0 ~ 1之间. 将$sim_{attr}(u,v)$值较小的前 K 个用户判定为目标用户的邻居用户, 预测新用户的评分值:

$$p_{i,j}=\overline{p_i}+\frac{\sum_{u\in C} sim(u_i,u_j)\times(R_{u,i}-\overline{R_u})}{\sum_{u\in C}\left|sim(u_i,u_j)\right|} \tag{9}$$

融合用户信息模型的基于用户的协同过滤算法流程图如图 1 所示.

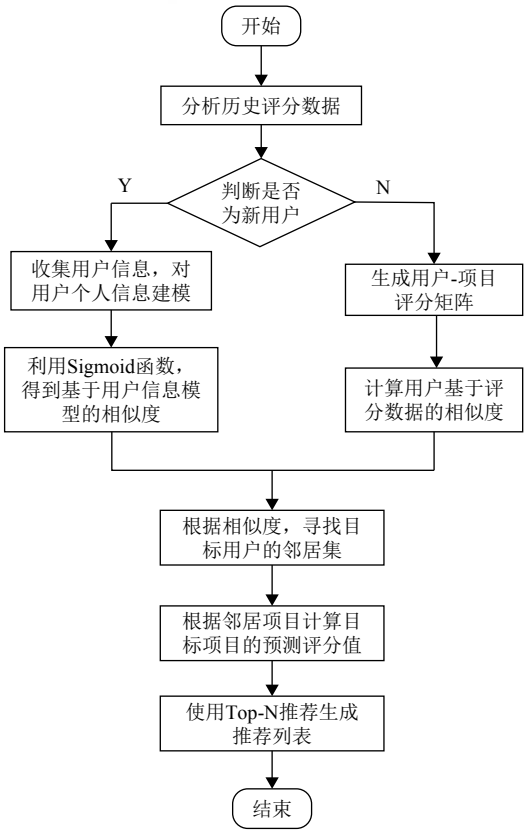

图 1 新用户冷启动算法

2.2.2 采用层次聚类的基于项目的协同过滤算法

协同过滤冷启动问题的另一方面是新项目的冷启动. 推荐系统中的项目都有各自的内容信息, 比如书籍的书名、出版年份、类型、作者; 食品的类别、成分; 音乐的年份、流派、作曲者等等. 在没有项目历史评分记录数据的情况下, 本文的新算法根据这些内容信息, 分析物品内容之间的相关度获取新项目和其他项目之间的相似度, 提出了采用凝聚式层次聚类的新项目相似度算法.

凝聚式层次聚类的相似度计算主要分为三步进行:

(1) 数据初始化处理

对于数值类信息, 可直接用于欧式距离计算. 对非数值类信息, 计算项目属性信息的补集元素个数, 作为欧氏距离中某一维上的距离长度值.

比如书本 A 的出版年份是 2015 年, 类别标签有: 数据分析/Python 编程/深度学习, 出版社是人民邮电出版社; 书本 B 的出版年份是 2018 年, 类别标签有: 数据分析/Matlab 编程/深度学习, 出版社是人民邮电出版社. 那么, A_1–B_1=3, A_2–B_2=1, A_3–B_3=0. 这三个值用于计算 A, B 之间的欧式距离.

(2) 计算欧式距离

假设项目有 n 种内容信息, 项目 i 对应的第 k 种内容信息记为i_k, 欧式距离的计算公式如式 (10) 所示.

$$d(i,j)=\sqrt{\sum_{k=1}^{n}(i_k-j_k)^2} \tag{10}$$

以 5 本不同的书籍举例, 表 1 是这 5 本书对应的欧氏距离初始矩阵.

表 1 欧式距离初始矩阵

	$b1$	$b2$	$b3$	$b4$	$b5$
$b1$	0	3.74	2.44	3.46	1
$b2$	3.74	0	2.82	1.73	3.31
$b3$	2.44	2.82	0	3.74	1.73
$b4$	3.46	1.73	3.74	0	3.46
$b5$	1	3.31	1.73	3.46	0

(3) 凝聚式层次聚类

根据第 (2) 步得到的欧式距离矩阵, 选择距离最近的两个簇 $b1$, $b5$. 合并 $b1$, $b5$ 为簇$\{b1, b5\}$,接着利用组平均准则, 选取其他簇与合并簇所有点之间距离的平均值作为下一步的邻近值, 更新欧式距离矩阵, 如表 2 所示.

$$\begin{cases} AVG.distance(\{b1,b5\},b2)=3.525 \\ AVG.distance(\{b1,b5\},b3)=2.085 \\ AVG.distance(\{b1,b5\},b4)=3.46 \end{cases} \tag{11}$$

表 2 迭代一次后的欧式距离矩阵

	$b1, b5$	$b2$	$b3$	$b4$
$b1, b5$	0	3.525	2.085	3.46
$b2$	3.525	0	2.82	1.73
$b3$	2.085	2.82	0	3.74
$b4$	3.46	1.73	3.74	0

由表 2 合成新的合并簇{$b2, b4$},重复前面的步骤,继续迭代更新矩阵.

$$\begin{cases} AVG.distance(\{b2,b4\},b3) = 3.28 \\ AVG.distance(\{b1,b5\},\{b2,b4\}) = 3.4925 \end{cases} \tag{12}$$

由表 3 合成新的合并簇{$b1, b5, b3$}, {$b2, b4$}, 最终聚类结果如图 2 所示, 可以直观地找到目标书籍的邻近集 (和目标书籍在同一簇中的其他书籍), 根据邻近集书籍的评分预测目标书籍的可能获得的评分.

表 3 迭代两次后的欧式距离矩阵

	$\{b1, b5\}$	$\{b2, b4\}$	$b3$
$\{b1, b5\}$	0	3.4925	2.085
$\{b2, b4\}$	3.4925	0	3.28
$b3$	2.085	3.28	0

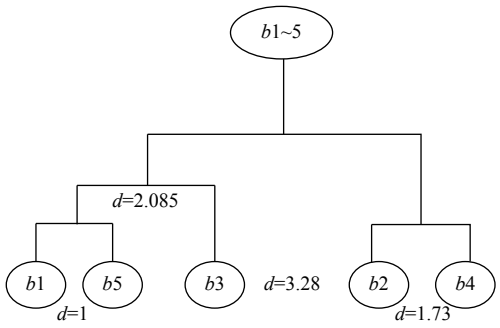

图 2 书籍聚类结果树图示

评分预测:

$$p_{ij} = \overline{p_j} + \frac{\sum_{r\in S} sim(j,r)\times\left(R_{i,r} - \overline{R_r}\right)}{\sum_{r\in S}|sim(j,r)|} \tag{13}$$

采用凝聚式层次聚类的基于项目的协同过滤算法流程图如图 3 所示.

3 实验结果与分析

3.1 实验数据集

本文选用 GroupLens 提供的网络开源数据集 MovieLens 作为测试数据集[13], 整个数据集包括了六千多个电影观看者对 3900 多部电影的 10 万多条评价. 由三部分组成, 包括用户集, 电影集和用户-电影评分集.

3.2 实验评价标准

本文采用 Top-N 推荐[14], 对推荐结果的质量用查准率 *Precision*、查全率 *Recall* 来衡量, 计算方式如式 (14)、(15).

$$Recall = \frac{\sum_{u\in U}|R_{(u)}\cap T_{(u)}|}{\sum_{u\in U}|T_{(u)}|} \tag{14}$$

$$Precision = \frac{\sum_{u\in U}|R_{(u)}\cap T_{(u)}|}{\sum_{u\in U}|R_{(u)}|} \tag{15}$$

其中, $R_{(u)}$是训练集上基于用户行为所给的推荐列表, $T_{(u)}$是测试集上的用户行为列表.

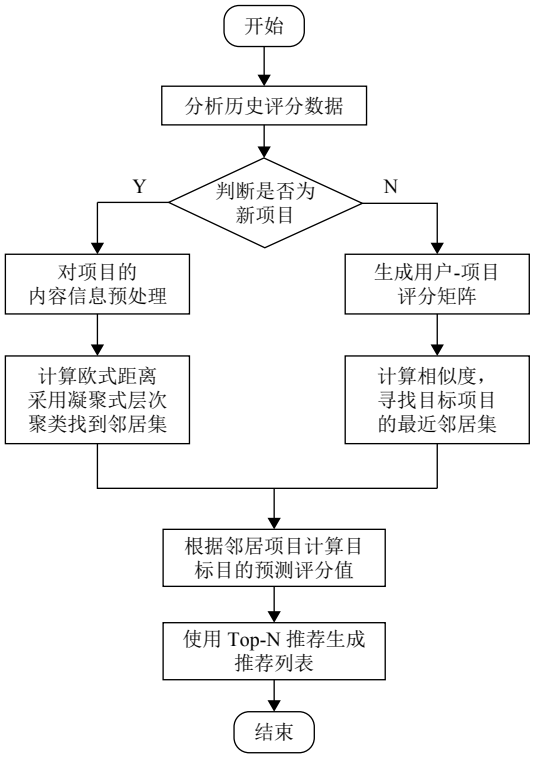

图 3 新项目冷启动算法

3.3 实验结果验证

3.3.1 新用户冷启动算法验证

本文实验以 MovieLens 数据集为例, 通过电影推荐验证新算法的性能. 将训练集和测试集的比例分为 9:1, 新用户与老用户的比例设为 3:7, 改变邻居数 K 值

在 5~40 之间, 进行多组实验.

MovieLens 的用户集中包含了性别、年龄、职业三项用户属性, 本文分析这三项特征信息, 计算出用户之间的特征差$attr(u,v)$如式 (16).

$$attr(u,v)=\alpha\cdot sex+\beta\cdot age+\gamma\cdot occupation \tag{16}$$

本文的 α、β、γ 皆取 1/3, 满足式 (6) 的条件.

对 User.data 的预处理:

(1) 性别判定

MovieLens 数据集中男性用户性别表示为 M, 女性用户性别表示为 F. 若两用户性别相同, *sex* 取值为 0, 若不同, *sex* 取值为 1.

(2) 年龄量化 (表 4)

表 4 MovieLens 用户年龄的量化

年龄	7–17	18–25	26–35	36–45	46–60	61–73
量化值	1	2	3	4	5	6

(3) 职业量化 (表 5)

表 5 MovieLens 用户职业的量化

职业类别	other	Academic/educator	artist	clerical/admin	college/grad student	customer service	doctor/health care
量化值	0	1	2	3	4	5	6
职业类别	executive/managerial	farmer	homemaker	K-12 student	lawyer	programmer	retired
量化值	7	8	9	10	11	12	13
职业类别	sales/marketing	scientist	self-employed	technician/engineer	tradesman/craftsman	unemployed	writer
量化值	14	15	16	17	18	19	20

由此, 式 (14) 中 *sex* 取值为 0 或 1, *age*, *occupation* 的取值为两用户特征值量化后差值的绝对值.

求得用户特征差值$attr(u,v)$后, 利用 Sigmoid 函数, 计算用户 u 和用户 v 之间的用户特征信息相似度 $sim_{attr}(u,v)$, 如式 (8).

传统协同过滤算法采用改进的余弦相似度 User-IIF 算法[15], 如式 (17) 所示, 降低了用户 u 和用户 v 共同兴趣列表中热门物品对他们相似度的影响.

$$w_{uv}=\frac{\sum_{i\in N_{(u)}\cap N_{(v)}}\frac{1}{\log(1+|N(i)|)}}{\sqrt{|N(u)||N(v)|}} \tag{17}$$

求得相似度后, 计算预测评分, 为用户进行推荐. 进行多次实验, 求得多组评估指标, 并用 Matlab 进行仿真, 仿真结果如图 4、图 5 所示.

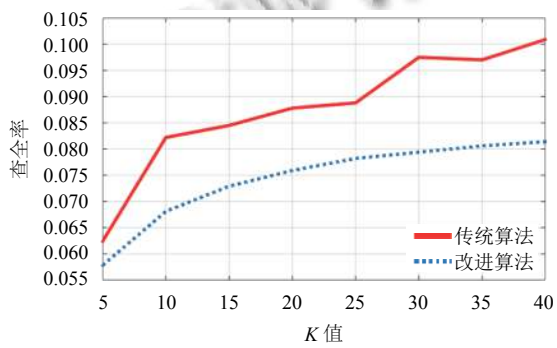

图 4 改进前后算法查全率随 K 值的变化

图 4 表示基于用户的协同过滤算法在新用户的情况下, 新算法与传统算法的查全率与邻居数的变化关系图. 由图可知, 随着邻居数 K 值增加, 系统的查全率呈上升趋势, 并且新算法的查全率高于传统算法, 说明新算法的检索结果更有效.

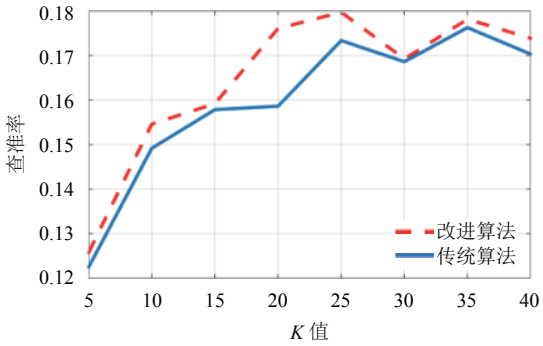

图 5 改进前后算法查准率随 K 值的变化

图 5 表示基于用户的协同过滤算法, 在新用户的情况下新算法与传统算法查准率与 K 值的关系. 可以看出, 新算法推荐结果的准确率高于传统算法. 改进后算法推荐精度更高, 有效地改善了系统新用户的冷启动问题.

结合图 4、图 5 发现, 在改进后的 User-based CF 中, 当邻居数取 35 时算法查全率最大, 查准率也比较高, 算法推荐质量较好.

3.3.2 新项目冷启动算法验证

本文提出的新项目的冷启动算法采用了凝聚式层

次聚类的思想, 在 MovieLens 数据集上进行实验结果验证, 将训练集和测试集的比例分为 9:1, 新项目和老项目的比例为 3:7.

Step 1. 对 movie.data 的数据初始化处理.

(1) 年份 (year) 关键词: 直接表示为 i_y、j_y.

(2) 派别 (genres) 关键词: 遍历电影所属的派别, 若两部电影有属于一个相同的派别, 则 g 值减 1, 否则, g 值保持不变 (g 初始值为 3), 最后得到目标电影和其他电影派别上的距离 g 值.

Step 2. 欧式距离的计算.

在对电影的基本数据处理之后, 计算电影之间的欧几里德距离. 欧式距离计算如式 (10).

Step 3. 层次聚类.

最后选取不同的邻居数 K 值在 5~40 之间, 进行多次实验比较加入新项目时, 改进前后基于项目的协同过滤算法在推荐精度上的变化, 如图 6、图 7 所示.

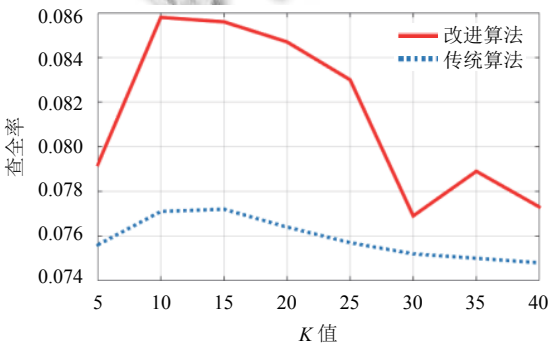

图 6 改进前后 ItemCF 算法查全率随 K 值的变化

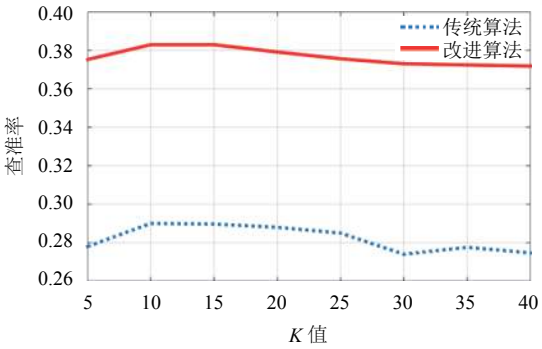

图 7 改进前后 ItemCF 算法查准率随 K 值的变化

传统方法采用的相似度计算公式如式 (18), 减轻了热门物品和其他众多物品相似的可能性.

$$w_{ij}=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)||N(j)|}} \tag{18}$$

从图 6 可以看出, 采用了层次聚类的算法在查全率上优于传统算法, 在邻居数取 10 的时候, 查全率达到最大值, 在邻居数取 25~35 之间时, 查全率波动较大, 并且呈下降趋势. 整体上看, 改进后的层次聚类算法推荐结果中被检索到的更多, 查全率更高.

图 7 比较了采用凝聚式层次聚类的协同过滤和传统基于项目的协同过滤算法在查准率上的性能, 由图可见, 算法的查准率比较平稳, 在加入了新项目后, 改进后算法的查准率优于传统算法的值, 表现出更好的推荐精度.

结合图 6、图 7, 在改进后的 Item-based CF 中, 邻居数取 10 时, 算法的推荐性能较好.

4 结论

本文首先介绍了协同过滤算法以及算法的冷启动问题, 重点对新用户和新项目的冷启动问题进行研究, 提出了融合用户信息模型的基于用户的协同过滤算法和采用层次聚类的基于项目的协同过滤算法. 具体研究工作如下:

(1) 针对新用户的冷启动, 算法提取了用户个人特征信息, 为用户信息建模, 调用 Sigmoid 函数求得基于用户特征模型的相似度.

(2) 对于新项目的冷启动, 算法提取项目的信息属性, 计算出欧式距离, 采用凝聚式层次聚类的方法, 找到目标项目的邻居项目, 计算预测评分, 完成推荐.

(3) 选用网络开源数据集 MovieLens 进行实验验证, 将新算法与传统算法多次实验对比. 结果表明, 在新用户和新项目的情况下, 新算法推荐结果的查全率、查准率都有所提升, 有效地缓解了传统协同过滤算法的冷启动问题, 改善了推荐质量.

参考文献

1 Maes P. Agents that reduce work and information overload. Communications of the ACM, 1994, 37(7): 30–40. [doi: 10.1145/176789.176792]

2 Guo YY, Liu QC. E-commerce personalized recommendation system based on multi-agent. Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery. Yantai, China. 2010. 1999–2003.

3 Bobadilla J, Ortega F, Hernando A. A collaborative filtering similarity measure based on singularities. Information Processing & Management, 2012, 48(2): 204–217.

4 Hu R, Pu P. Enhancing collaborative filtering systems with personality information. Proceedings of the 5th ACM Conference on Recommender Systems. Chicago, IL, USA. 2011. 197–204.
5 Wang JW. A collaborative filtering systems based on personality information. Proceedings of 2015 International Industrial Informatics and Computer Engineering Conference. Xi'an, China. 2015. [doi: 10.2991/iiicec-15.2015.163]
6 Goldberg D, Nichols D, Oki BM, *et al*. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 1992, 35(12): 61–70. [doi: 10.1145/138859.138867]
7 申辉繁. 协同过滤算法中冷启动问题的研究[硕士学位论文]. 重庆: 重庆大学, 2015.
8 马卓. 协同过滤推荐算法的研究与改进[硕士学位论文]. 秦皇岛: 燕山大学, 2015.
9 Xu JW, Yao Y, Tong HH, *et al*. RaPare: A generic strategy for cold-start rating prediction problem. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(6): 1296–1309. [doi: 10.1109/TKDE.2016.2615039]
10 Nguyen VD, Sriboonchitta S, Huynh VN. Using community preference for overcoming sparsity and cold-start problems in collaborative filtering system offering soft ratings. Electronic Commerce Research and Applications, 2017, 26: 101–108. [doi: 10.1016/j.elerap.2017.10.002]
11 Katarya R, Verma OP. Effective collaborative movie recommender system using asymmetric user similarity and matrix factorization. Proceedings of 2016 International Conference on Computing, Communication and Automation. Noida, India. 2016. 71–75.
12 乔雨, 李玲娟. 推荐系统冷启动问题解决策略研究. 计算机技术与发展, 2018, 28(2): 83–87. [doi: 10.3969/j.issn.1673-629X.2018.02.019]
13 http://www.grouplens.org/node/73.
14 程淑玉. 基于协同过滤算法的个性化推荐系统的研究[硕士学位论文]. 合肥: 合肥工业大学, 2010.
15 Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, WI, USA. 1998. 43–52.