结合贡献度与时间权重的协同过滤推荐算法

贾俊康,李玲娟
(南京邮电大学 计算机学院,江苏 南京 210023)

摘 要:传统的协同过滤推荐算法未考虑用户兴趣随时间动态变化,以及不同用户对同一项目评分差异过大对推荐效果的影响,导致推荐效果不理想。针对以上问题,以进一步提高基于用户的协同过滤推荐算法的精度为目标,设计了一种结合贡献度与时间权重的协同过滤推荐算法 CTCF。该算法在用户相似度计算中引入可信误差阈值、贡献度与时间权重。首先,利用用户评分信息构建用户-评分矩阵与用户-评分时间矩阵,依据可信误差阈值来计算用户贡献度;然后,引入拟合贡献度与时间因子的遗忘曲线得到时间权重,再将时间权重引入皮尔逊相关系数中计算用户相似度;找出目标用户的邻居集,并预测目标用户对邻居集对应项目中未评分项目的评分;最后,按评分由高到低生成 Top-N 推荐。在 MovieLens 数据集上的测试结果表明,CTCF 算法具有更高的 F1 值,有效提高了推荐精度和动态性。

关键词:贡献度;时间因子;相似度;协同过滤;推荐
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2023)03-0167-06
doi:10.3969/j.issn.1673-629X.2023.03.025

Collaborative Filtering Recommendation Algorithm Combining Contribution and Time Weight

JIA Jun-kang, LI Ling-juan
(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: The traditional collaborative filtering recommendation algorithm does not consider the dynamic changes of users' interests over time and the impact of excessive differences in the scores of different users on the same item on the recommendation effect, resulting in the unsatisfactory recommendation effect. In view of the above problems, a collaborative filtering recommendation algorithm CTCF, which combines contribution degree and time weight, is designed to further improve the accuracy of user-based collaborative filtering recommendation algorithm. The algorithm introduces the trusted error threshold, contribution degree and time weight into the user similarity calculation. Firstly, the user score matrix and user score time matrix are constructed by using the user score information, and the user contribution degree is calculated by the trusted error threshold. Then, the forgetting curve of fitting contribution degree and time factor is introduced to obtain the time weight, and the time weight is introduced into the Pearson correlation coefficient to calculate the user similarity, find out the neighbor set of the target user, and predict the score of the target user on the non-scored items in the corresponding items of the neighbor set. Finally, the Top-N recommendation is generated according to the score from high to low. The test results on MovieLens dataset show that CTCF algorithm has higher F1 value and effectively improves the recommendation precision and dynamics.

Key words: contribution degree; time factor; similarity; collaborative filtering; recommendation

0 引 言

随着互联网和信息技术的普及,人类全面进入 web2.0 时代以后,就面临着“信息超载”[1]问题,用户想要从海量数据中筛选出满意的结果变得愈加艰难。因此,作为信息过滤工具的推荐系统应运而生,它能够根据用户的需求,快速、准确地向用户进行有效推荐。推荐算法作为推荐系统的核心[2],直接决定了推荐效果。主流推荐算法包括基于内容的推荐、协同过滤推荐和混合推荐三类传统的推荐算法,以及各种与深度学习结合的推荐算法[3-4]。其中,协同过滤推荐算法是应用最为成功的推荐算法之一[5]。它又分为两类,一类是基于模型的协同过滤,另一类是基于内存的协同过滤。后者又有基于用户的协同过滤 UserCF 和基于项目的协同过滤 ItemCF 之分[3]。UserCF 源于生活

收稿日期:2022-05-11 **修回日期**:2022-09-14
基金项目:国家重点研发计划专项(2020YFB2104002);江苏省重点研发计划(BE2019740)
作者简介:贾俊康(1995-),男,硕士研究生,CCF 会员(K4545G),研究方向为个性化推荐;通讯作者:李玲娟,教授,CCF 会员(E200015276M),研究方向为数据挖掘、推荐系统、社团划分。

中的经验假设:兴趣爱好类似的用户,他们对同一项目可能有着相似的喜好程度[6],这种算法不是分析用户之间潜在的某种关系,而是从用户历史行为数据信息中计算用户相似度,从中寻找相似用户进行推荐,思路简单,易于实现[7]。但是这种算法至少存在三方面的问题:①未考虑时间因素,没有将用户不同时间的评分区别对待;②未考虑跟风用户带来的影响;③未考虑不同用户对同一个项目评分差异过大带来的影响。

由于存在以上问题,传统的 UserCF 算法的推荐精度仍有待提高。为此,该文设计了一种结合贡献度与时间权重的协同过滤推荐算法(collaborative filtering recommendation algorithm combining contribution and time weight),命名为 CTCF。该算法通过定义可信误差阈值与贡献度来减少不同用户对同一项目评分差异过大对推荐结果产生的负面影响;通过在用户相似度计算中引入拟合时间因子与贡献度的时间权重来动态模拟用户兴趣随时间的变化;既充分利用了传统的 UserCF 算法的优势,又避免其存在的问题,进一步提高了基于用户的协同过滤推荐算法的精度。

1 相关技术

1.1 UserCF 算法推荐过程

UserCF 的推荐过程如图 1 所示。

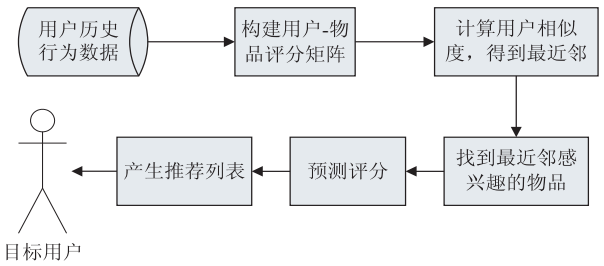

图 1 UserCF 算法推荐过程

主要步骤为[8]:

①利用用户的历史行为数据构建用户-项目评分矩阵;

②使用用户相似度计算方法计算用户相似度,并找出目标用户的最近邻;

③选择最近邻喜欢而目标用户未评分的项目;

④预测目标用户对第③步中得到的项目的评分,并将预测评分高(即喜欢程度高)的项目推荐给目标用户。

1.2 评分矩阵

假设 $U=\{u_1,u_2,\cdots,u_m\}$ 表示 m 个用户,$I=\{i_1,i_2,\cdots,i_n\}$ 表示 n 个项目。用户-项目评分矩阵可表示为 $\boldsymbol{R}_{m,n}=\{R_{u,i}\}$,$R_{u,i}$ 是用户 u 对项目 i 的评分。表 1 是 5 个用户对 6 个项目的评分示例,其中"/"表示该用户没有对项目进行过评分,评分值为 1 至 5 的整数,数值越大表示用户对这个项目喜好程度越高。

表 1 用户对项目的评分示例

用户	item1	item2	item3	item4	item5	item6
user1	1	1	1	/	5	2
user2	5	5	2	5	/	5
user3	/	2	/	3	5	1
user4	1	/	3	2	/	/
user5	2	/	/	1	4	/

与之对应的评分矩阵如下:

$$\boldsymbol{R}=\begin{pmatrix}1&1&1&/&5&2\\5&5&2&5&/&5\\/&2&/&3&5&1\\1&/&3&2&/&/\\2&/&/&1&4&/\end{pmatrix}$$

1.3 相似度计算方法

相似度的计算方法主要有余弦相似性[9]、欧几里得相似度[10]、皮尔逊相关系数[11]等。余弦相似度是两个评分向量夹角的余弦值。欧几里得相似度指的是两个点之间的真实距离。皮尔逊相关系数广泛用于度量两个变量之间的相关性,其计算公式如下:

$$\mathrm{sim}(a,b)=\frac{\sum_{i\in I_{a,b}}(g_{a,i}-\bar{g}_a)*(g_{b,i}-\bar{g}_b)}{\sqrt{\sum_{i\in I_{a,b}}(g_{a,i}-\bar{g}_a)^2}*\sqrt{\sum_{i\in I_{a,b}}(g_{b,i}-\bar{g}_b)^2}}\tag{1}$$

其中,$\bar{g}_a$ 表示用户 a 评分的平均值,$g_{a,i}$ 表示用户 a 对项目 i 的评分,$I_{a,b}$ 表示用户 a 和用户 b 已评分项目的交集。

2 CTCF 算法设计

该文所设计的 CTCF 算法的核心内容包括:可信误差阈值与贡献度的计算、拟合贡献度与时间因子的时间权重的计算、用户相似度的计算、评分的预测。

2.1 可信误差阈值与贡献度的计算方法设计

如前所述,传统的 UserCF 算法,在计算用户相似度时,可能会遇到不同用户对同一项目评分差异较大的情况,例如表 1 中 user1 和 user2 虽然共同评分的项目占比为 67.7%,但其中至少有 75% 的项目评分差异较大。这种不同用户对共同项目的评分差异从一定程度上能够反映出用户兴趣的差异,因此,在判断当前用户对目标用户的价值时,应该充分考虑这种差异。该文分别定义了可信误差阈值与贡献度,用可信误差阈值来计算贡献度,依据贡献度来评定当前用户对目标用户的价值大小,从而为进一步计算用户相似度提供权重。

定义 1 可信误差阈值:是不同用户对同一项目的

评分差异的度量,计算方法如公式(2)所示:

$$\chi = \frac{\sum_{i\in|I(a)\cap I(b)|}|(g_{a,i}-\bar{g}_{a,b})|}{|I(a)\cap I(b)|*2}+\bar{g}_{a,b} \quad (2)$$

其中,χ 表示用户 a 和用户 b 的可信误差阈值,$g_{a,i}$ 表示用户 a 对项目 i 的评分,$I(a)$ 和 $I(b)$ 分别是用户 a 和用户 b 已评分项目的集合,$|I(a)|$ 和 $|I(b)|$ 分别是用户 a 和用户 b 已评分项目的数量,$|I(a)\cap I(b)|$ 为用户 a 和用户 b 共同评分项目的数量,$\bar{g}_{a,b}$ 表示用户 a 和用户 b 共同评分项目的评分均值,计算方法如下:

$$\bar{g}_{a,b}=\frac{\sum_{i\in|I(a)\cap I(b)|}(g_{a,i}+g_{b,i})}{|I(a)\cap I(b)|*2} \quad (3)$$

定义 2 贡献度:是用户间的参考价值度量,反映不同用户的共同喜好程度高低,计算方法如公式(4)所示:

$$\sup=\frac{t}{|I(a)\cup I(b)|} \quad (4)$$

其中,t 为 $|g_{a,i}-g_{b,i}|<\chi$ 的数目,$|I(a)\cup I(b)|$ 为用户 a 和用户 b 评分项目的总数量。可以看出 $\sup\in[0,1]$,且评分相近的项目越多(即共同喜好程度越高),贡献度 sup 就越大,彼此的参考价值越高。

2.2　拟合贡献度与时间因子的时间权重计算方法设计

如前文所述,传统的 UserCF 算法忽略了用户兴趣爱好会随时间变化这一事实[12],由此得出的最近邻可能并不准确。一般来讲用户近期的行为更能充分说明其目前的兴趣爱好。随着时间推移,用户会逐渐减弱对某个项目的喜好,这种趋势呈现出先快后慢、最后趋于稳定的特点,这与艾宾浩斯曲线[13]很相似。因此,该文进一步将贡献度与时间因子相结合来对艾宾浩斯曲线加以改进,设计了拟合贡献度与时间因子的时间权重,计算方法如公式(5)所示。

$$\delta_i^a=e^{-\sup*\frac{T_{a,i}-T_{a,\text{first}}}{T_{a,\text{last}}-T_{a,\text{first}}}} \quad (5)$$

其中,δ_i^a 就是用户 a 对项目 i 的时间权重,反映用户兴趣随着时间推移的变化情况,时间越近的评分有越高的权重。sup 表示贡献度。$T_{a,\text{first}}$ 表示用户 a 第一次评价项目的时间,$T_{a,i}$ 表示用户 a 评价项目 i 的时间,$T_{a,\text{last}}$ 表示用户 a 最后一次评价项目的时间。

不难看出,该时间权重既考虑了不同用户对同一项目评分差异过大对推荐结果的影响,又动态模拟了用户兴趣随时间的变化。

2.3　改进的用户相似度计算方法设计

在计算用户相似度时,该文对皮尔逊相关系数加以改进,引入公式(5)的拟合贡献度与时间因子的时间权重,同时考虑目标用户 a 评价过的项目可能包含某用户 b 评价过的全部项目(即前者是后者的超集,如表 1 中 user2 与 user4)的情况,因后者不能为目标用户提供有用的信息,将其视为跟风用户[14],并将目标用户与跟风用户的相似度置零。改进后的相似度计算方法如公式(6)所示:

$$\text{sim}(a,b)=\begin{cases}0, I(b)\subset I(a)\\ \dfrac{\sum_{i\in I}(g_{a,i}*\delta_i^a-\bar{g}_a^{\delta})(g_{b,i}*\delta_i^b-\bar{g}_b^{\delta})}{\sqrt{\sum_{i\in I}(g_{a,i}*\delta_i^a-\bar{g}_a^{\delta})^2}\sqrt{\sum_{i\in I}(g_{b,i}*\delta_i^b-\bar{g}_b^{\delta})^2}}, \text{其他}\end{cases} \quad (6)$$

其中,$\text{sim}(a,b)$ 是用户 a 和用户 b 的相似度,a 是目标用户,$I(a)$ 和 $I(b)$ 分别是用户 a 和用户 b 已评分项目的集合,$\bar{g}_a^{\delta}$ 表示用户 a 对各项目的评分乘以时间权重后的平均评分。

2.4　评分预测方法

计算得到目标用户与其他用户的相似度后,按照相似度降序排列,取前 K 个作为目标用户的近邻集,产生近邻评价过的项目集,并用公式(7)[15]对其中目标用户未评价的项目进行评分预测。

$$R_{a,i}^*=\bar{g}_a+\frac{\sum_{b\in U_a}\text{sim}(a,b)*(g_{b,i}-\bar{g}_b)}{\sum_{b\in U_a}|\text{sim}(a,b)|} \quad (7)$$

其中,$R_{a,i}^*$ 表示目标用户 a 对未评价项目 i 的预测评分值,$\bar{g}_a$ 为目标用户 a 的平均评分值,U_a 为目标用户 a 的近邻集。

最后,按照目标用户 a 的预测评分的降序对项目排序,取 Top-N 个加以推荐。

2.5　CTCF 算法流程

CTCF 算法的总体流程如图 2 所示。

具体步骤可描述如下:

算法:结合贡献度与时间权重的协同过滤推荐算法。

输入:用户评分信息,目标用户 a,推荐结果个数 N;

输出:推荐给目标用户 a 的项目列表。

Step1:首先,对用户评分信息做预处理,用公式(8)将时间戳归一化;其中,t_i 为当前时间戳,$t_{\min}$ 为最小时间戳,$t_{\max}$ 为最大时间戳。

$$t=\frac{t_i-t_{\min}}{t_{\max}-t_{\min}} \quad (8)$$

然后,构建出用户-项目评分矩阵与用户-项目评分时间矩阵。

Step2:用公式(4)计算贡献度 sup,用公式(5)计

算时间权重。

Step3:用公式(6)计算相似度矩阵,并产生目标用户 a 的近邻集以及该近邻集已评价的项目集 I,在 I 中去除用户 a 已评价过的项目,构成 I'。

Step4:用公式(7)预测目标用户 a 对 I' 中各个项目的评分。

Step5:将预测得到的评分由高到低排序,并将前 N 个项目作为目标用户 a 的推荐集。

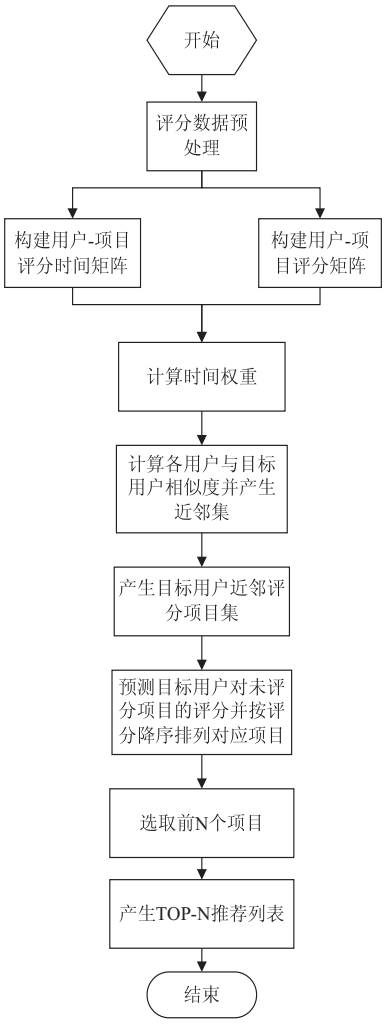

图 2 CTCF 算法流程

3 实验及结果分析

为了测试 CTCF 算法的性能,做了相关实验。

3.1 实验数据及环境

实验所用的数据是 MovieLens ml-100K 数据集中 u.data 文件的数据,有 943 个用户、1 652 部电影、100 000个评分[16]。随机选取其中的 75% 用作训练集,剩余的 25% 用作测试集,用户的评分范围为 1 ~ 5,数值越大表示用户兴趣越大。

实验环境:CPU 为 Intel Core i7,主频为 2.2 GHz,内存为 16 GB;操作系统是 macOS 10.13;编程语言是 Python3.7。

3.2 评价指标

该文采取目前比较主流的推荐算法评价指标召回率(recall)和精度(precision)来评估算法。分别用公式(9)和公式(10)进行计算。

$$\text{recall} = \frac{\sum_{a \in U} |R(a) \cap I'(a)|}{\sum_{a \in U} |I'(a)|} \tag{9}$$

$$\text{precision} = \frac{\sum_{a \in U} |R(a) \cap I'(a)|}{\sum_{a \in U} |R(a)|} \tag{10}$$

其中,$R(a)$ 表示算法推荐给用户 a 的项目集合,$I'(a)$ 表示用户 a 实际喜欢的项目集合,U 表示测试集中所有用户的集合。

为了比较全面地评价算法,还采用 F-Measure 作为评价指标,F-Measure 公式如下:

$$F\alpha = \frac{(1 + \alpha^2) * \text{recall} * \text{precision}}{\alpha^2 * (\text{recall} + \text{precision})} \tag{11}$$

当参数 $\alpha = 1$ 时,就是最常用的 F1 值,即:

$$F1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}} \tag{12}$$

F1 的值越大表示算法的综合表现越好。

3.3 实验结果分析

(1)可信误差阈值(χ)的有效性实验。

为验证定义的可信误差阈值(χ)的有效性,在 χ 取常数(0、1、2、3、4)以及用公式(2)进行动态计算的情况下,测试 CTCF 算法的 F1 值,结果如图 3 所示。

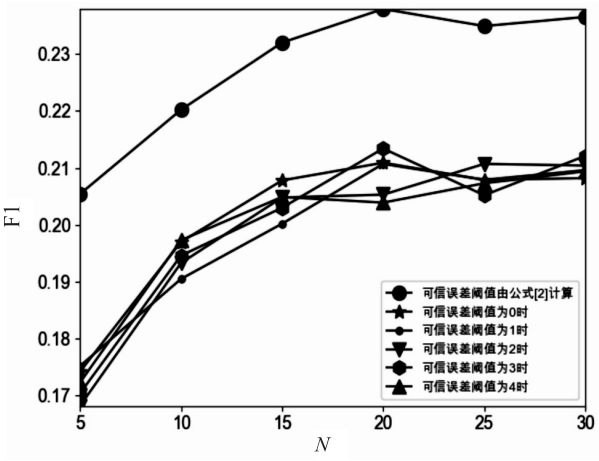

图 3 不同可信误差阈值对 F1 值的影响

从图 3 可以看出,随着近邻数的增加,F1 值不断增大,然后趋于稳定,而且采用该文定义的可信误差阈值时,F1 明显高于其他几种情况。

(2)CTCF 算法总体性能实验。

为验证 CTCF 算法能有效提升推荐质量,将其与采用皮尔逊相似度的 UserCF、使用了改进皮尔逊相似度的文献[17]的算法作对比,统计三种算法的 precision、recall 和 F1 值,结果如图 4 至图 6 所示。

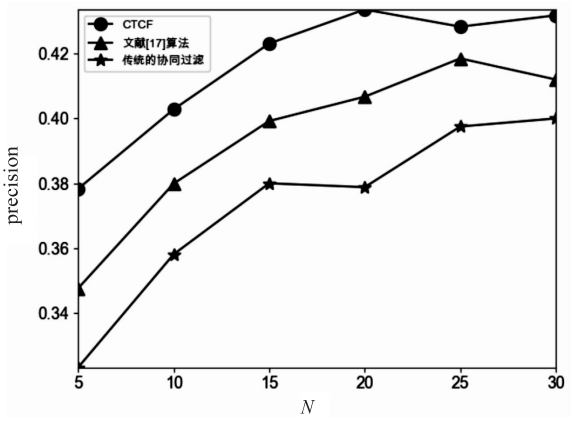

图 4　不同近邻数 N 对应的 precision

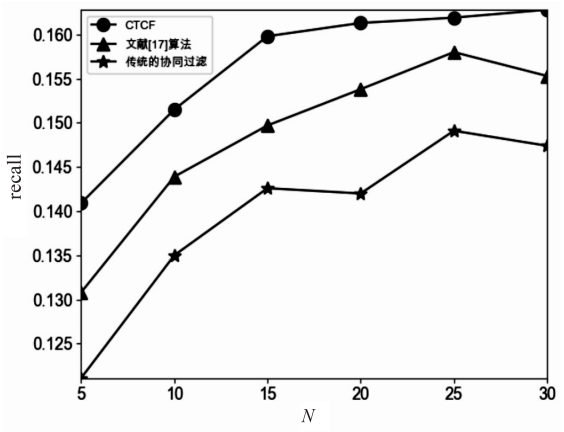

图 5　不同近邻数 N 对应的 recall

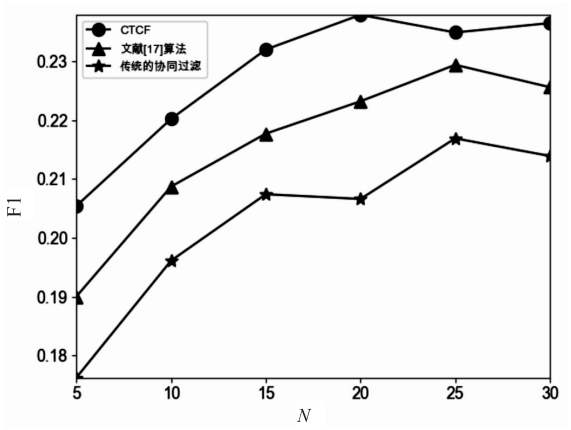

图 6　不同近邻数 N 对应的 F1

从图 4 和图 5 可以看出,不同近邻数下,CTCF 算法对 recall 和 precision 有了明显的提升。从图 6 可以看出,CTCF 算法的 F1 值始终高于另外两种算法。

时间复杂度也是评价一个算法优劣的重要指标,CTCF 算法与 UserCF 类似,最耗时的阶段是用户相似度计算,时间复杂度是 $O(n^2)$,n 为用户数;文献[17]算法的时间复杂度是 $O(n^3)$ 。为对比三种算法的时间开销,进行了相关实验,结果如图 7 所示。

由图 7 可知,CTCF 算法与传统的协同过滤算法耗时相近,远远小于文献[17]算法。综上,CTCF 算法在不增加时间复杂度的前提下,提升了推荐精度。

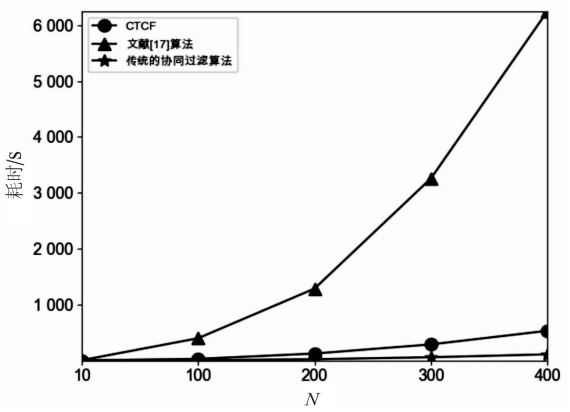

图 7　三种算法对应不同近邻数 N 的时间开销

4　结束语

以提高推荐精度为目标,该文设计了一种结合贡献度与时间权重的协同过滤算法 CTCF,该算法对传统的基于用户的协同过滤推荐算法做了改进,在用户相似度计算中引入贡献度与时间因子,降低了不同用户对相同项目评分差异过大带来的影响,同时在一定程度上增大了用户近期反馈行为的权重,对用户早期的行为信息权重进行了衰减,客观地反映了用户兴趣随时间的动态变化。在 MovieLens 数据集上与类似算法的对比实验结果表明,设计的 CTCF 算法能有效提高推荐质量。

参考文献:

[1] EPPLER M J, MENGIS J. The concept of information overload: a review of literature from organization science, accounting, marketing, MIS, and related disciplines[J]. Information Society, 2004, 20(5): 325-344.

[2] 彭珍连,王　健,何克清,等. 一种基于特征模型和协同过滤的需求获取方法[J]. 计算机研究与发展,2016,53(9):2055-2066.

[3] GUPTA S, DAVE M. An overview of recommendation system: methods and techniques[C]//1st international conference on advancements in computing and management (ICACM). Jaipur: Springer, 2019: 231-237.

[4] BENKESSIRAT S, BOUSTIA N, REZOUG N. Overview of recommendation systems[M]//Smart education and e-learning 2019. Singapore: Springer, 2019: 357-372.

[5] 郭弘毅,刘功申,苏　波,等. 融合社区结构和兴趣聚类的协同过滤推荐算法[J]. 计算机研究与发展,2016,53(8):1664-1672.

[6] 姚平平,邹东升,牛宝君. 基于用户偏好和项目属性的协同过滤推荐算法[J]. 计算机系统应用,2015,24(7):15-21.

[7] MELVILLE P, MOONEY R J, NAGARAJAN R. Content-boosted collaborative filtering for improved recommendations [C]//Eighteenth national conference on artificial intelligence. Edmonton: American Association for Artificial Intelli-

gence,2002:187-192.
[8] GONG S. A collaborative filtering recommendation algorithm based on user clustering and item clustering[J]. Journal of Software,2010,5(7):745-752.
[9] MOHANA H,SURIAKALA M. Integrated cosine and tuned cosine similarity measure to alleviate data sparsity issues for personalized recommendation[C]//2018 3rd international conference on computational systems and information technology for sustainable solutions (CSITSS). Bangalore: IEEE, 2018:41-49.
[10] KHOSHNESHIN M,STREET W N. Collaborative filtering via euclidean embedding[C]//Proceedings of the fourth ACM conference on recommender systems. Barcelona: ACM,2010: 87-94.
[11] XIA C,JIANG X,LIU S,et al. Dynamic item-based recommendation algorithm with time decay[C]//2010 sixth international conference on natural computation. Yantai: IEEE, 2010:242-247.
[12] 曾 安,高成思,徐小强. 融合时间因素和用户评分特性的协同过滤算法[J]. 计算机科学,2017,44(9):243-249.
[13] 梁艺伟. 基于遗忘曲线的英语绘本阅读推荐系统的设计与实现[D]. 北京:北京交通大学,2019.
[14] 李 聪,梁昌勇,马 丽. 基于领域最近邻的协同过滤推荐算法[J]. 计算机研究与发展,2008,45(9):1532-1538.
[15] ZHANG Na,WANG Zhiyong. Collaborative filtering recommendation algorithm based on hybrid similarity[M]//Recent developments in intelligent computing,communication and devices. Singapore:Springer,2019:617-625.
[16] 傅金京,李玲娟. 基于用户特征和评分的精准推荐策略研究[J]. 南京邮电大学学报:自然科学版,2021,41(1):107-114.
[17] 刘超慧,韩传福,陈天成,等. 融合惩罚因子和时间权重的协同过滤推荐算法[J]. 信息技术与网络安全,2020,39(5):17-21.

(上接第 166 页)

rithms applied to clustering problem and data mining[C]// Proceedings of the 7th WSEAS international conference on simulation, modelling and optimization. Beijing: [s. n.], 2007:219-224.
[20] KHOTIMAH B K,IRHAMNI F,SUNDARWATI T R I. A genetic algorithm for optimized initial centers k-means clustering in SMEs[J]. Journal of Theoretical and Applied Information Technology,2016,90(1):23-30.
[21] KUO R J,CHANG C K,NGUYEN T P Q,et al. Application of genetic algorithm-based intuitionistic fuzzy weighted c-ordered-means algorithm to cluster analysis[J]. Knowledge and Information Systems,2021,63(7):1935-1959.
[22] 姚平平,邹东升,牛宝君. 基于用户偏好和项目属性的协同过滤推荐算法[J]. 计算机系统应用,2015,24(7):15-21.
[23] MOHAMMADREZAPOUR O,KISI O,POURAHMAD F. Fuzzy c-means and k-means clustering with genetic algorithm for identification of homogeneous regions of groundwater quality [J]. Neural Computing and Applications,2020,32(8):3763-3775.
[24] AL-BAKRI N F,HASHIM S H. Collaborative filtering recommendation model based on k-means clustering[J]. Al-Nahrain Journal of Science,2019,22(1):74-79.