

基于人工蜂群的项聚类推荐算法

郭 磊, 胡 燕

(武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

摘 要: 旨在改善协同过滤推荐算法中暴露的推荐准确性和实时性问题, 提出基于人工蜂群的项聚类推荐算法. 在数据处理过程中利用人工蜂群聚类算法对项目聚类, 从相似性最高的几个聚类中搜索目标项目最近邻居, 以此剔除相似性较低项目的干扰提高推荐准确性, 同时也大大缩小项目空间, 提高了推荐实时性; 在评分预测过程中采用时间权重函数优化传统评分预测模型以提高评分预测的准确性, 从而提高推荐准确性. 实验结果表明, 推荐的准确性和实时性得到了有效提高.

关键词: 协同过滤; 推荐算法; 人工蜂群; 时间权重函数

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2017)06-0031-05

Item-clustering Recommendation Algorithm Based on Artificial Bee Colony

GUO Lei, HU Yan

(Wuhan University of Technology, Institute of Computer Science and Technology, Wuhan 430070, China)

Abstract: Aiming at improving the accuracy and real-time problem exposing in collaborative filtering recommendation algorithm, A new Item-clustering recommendation algorithm based on artificial bee colony algorithm is proposed. In the process of data processing, the artificial bee colony clustering algorithm is used to cluster the items, Search for the nearest neighbors of the target item from the highest similarity of several clusterings, in order to eliminate the interference of similar items to improve the accuracy of the recommendation, At the same time, it also greatly reduces the item space and improves the real-time recommendation; In the process of scoring, the time weighting function is used to optimize the traditional score prediction model in order to improve the accuracy of the prediction, so as to improve the accuracy of recommendation. The experimental results show that the accuracy and real-time performance are improved effectively.

Key words: collaborative filtering; recommendation algorithm; artificial bee colony; time weighting function

1 引言

电商的兴起带给人们方便的同时也困扰着人们:琳琅满目的商品让人无从选择. 推荐系统应运而生. 协同过滤作为最早且最成功的推荐算法^[1]得到了广泛应用. 它旨在挖掘用户历史行为, 并以此为基础作出推荐, 效果甚好. 但随着数据规模的增大, 协同过滤也暴露诸多问题, 主要体现在数据稀疏性、推荐准确性和实时性、冷启动等. 本文利用人工蜂群聚类算法对项目进行聚类, 仅在与目标项目相似性较

高的几个聚类中进行搜索, 可以显著减少搜索时间, 提高推荐实时性, 与此同时, 也直接剔除了相似性很低的项目对最近邻选取的干扰, 提高了推荐的推荐效果. 准确性; 此外, 在预测项目评分的过程中利用时间权重函数优化传统的评分预测模型, 进一步提高推荐准确性.

2 协同过滤推荐算法

传统协同过滤算法暴露出来的种种问题向人们发起了挑战, 相关方面的研究和改进也层出不穷. 文

献[2]通过计算地理位置的距离对项目集进行预过滤,缩小推荐的范围大幅减少推荐过程中计算量,能解决推荐的实时性问题[3];文献[4]从用户的角度出发定义了项目的新颖性,设计了相应的离线实验方案和评价指标,并将相异和时间普及融入到传统的协同过滤算法中,使预测用户未来需求的能力和推荐列表的覆盖能力得到明显改善。

基于项聚类的推荐算法通过对项目聚类来缩小搜索空间,对推荐实时性的提高有较大的帮助。其中,聚类方法的选择尤为重要。文献[5]选取了 K-means 算法作为聚类方法,但初始聚类中心的选取对聚类结果影响太过强烈,容易出现局部最优的情况,最终影响推荐结果的准确性;文献[6]吸取了文献[5]的教训,在其基础上利用粒子群优化算法对 K-means 的初始聚类中心选取进行了优化,虽然缓解了初始聚类中心对聚类的影响,但耗费了大量的时间在聚类上。

本文利用人工蜂群算法全局搜索能力强、收敛速度较快、控制参数少且易于实现等优点,对 K-means 算法的初始聚类中心选取进行优化,有效的降低了初始聚类中心对聚类结果的影响,得到了不错的聚类效果,从而减少了项目空间,大大提高了推荐实时性。同时,剔除了相似性较低的项目的干扰,且结合聚类算法本身的精度,也提高了推荐算法的准确性。此外,时间权重函数的引入优化了评分预测模型,也使得推荐准确性得到了提升。

3 基于人工蜂群的项聚类推荐算法

3.1 算法优化构思

本文主要围绕人工蜂群聚类算法和时间权重函数这两个点对传统协同过滤推荐算法进行改进,大致思路如图 1 所示。

3.2 基于 K-means 的人工蜂群聚类算法

3.2.1 原始 K-means 算法

设聚类样本 $x = \{x_i \in R^d, i = 1, 2, 3, \dots, n\}$, 划分为 k 个不同类别 $C = \{C_1, C_2, \dots, C_k\}$ 。式中, $x_i (i = 1, 2, 3, \dots, n)$ 为 d 维向量。

聚类中心计算如下:

$$C_j = \frac{1}{C_j} \sum_{i=1}^n x_i; j = 1, 2, 3, \dots, k \quad (1)$$

K-means 的准则函数如下:

$$J = \sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, C_j) \quad (2)$$

式中, $d(x_i, C_j)$ 表示 x_i 到其所属类中心 C_j 的距离; J

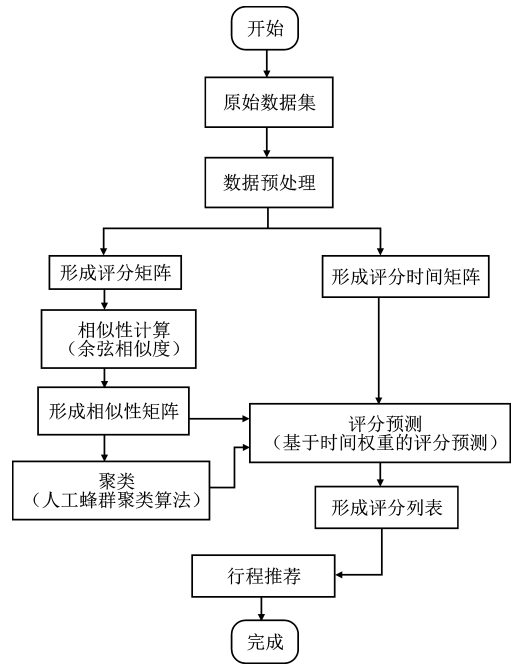


图 1 算法流程图

表示类内距之和。由于应用于推荐算法,这里 $d(x_i, C_j)$ 代表相似性。

3.2.2 人工蜂群算法

人工蜂群算法 (Artificial Bee Colony Algorithm, ABC) 是群体智能优化算法的一种[7],它启发于蜂群的觅食行为,算法实现过程如下:

(1) 初始化阶段:设置相关参数,按式(3)随机生成含有 SN 个蜜源的初始种群 $X = \{X_1, X_2, X_i, \dots, X_{SN}\}$ 。

$$X_{ij} = X_{jmin} + \text{rand}(0, 1) (X_{jmax} - X_{jmin}) \quad (3)$$

其中: $j \in \{1, 2, \dots, L\}$, X_{jmax} 和 X_{jmin} 分别代表该解向量第 J 维分量的上、下限; $\text{rand}(0, 1)$ 为 $(0, 1)$ 区间的随机数。

(2) 搜索开始阶段:引领蜂按式(4)随机搜索新的蜜源。

$$V_{ij} = X_{ij} + R_{ij} (X_{ij} - X_{kj}) \quad (4)$$

其中, $k \in \{1, 2, \dots, SN\}$, $j \in \{1, 2, \dots, L\}$, k, j 在各自区间随机选取。 R_{ij} 为 $[-1, 1]$ 区间的随机数。接着根据式(5)计算新蜜源 V_i 的适应度 fit_i

$$\text{fit}_i = \frac{CN_i}{J_i}; i = 1, 2, 3, \dots, SN \quad (5)$$

其中, CN_i 表示属于第 i 类的元素个数;由公式(2)可以推导出

$$J_i = \sum_{x_j \in C_i} d(x_j, C_i); j = 1, 2, 3, \dots, n \quad (6)$$

若新蜜源适应度大于原蜜源,则取而代之;否则保留原蜜源。接着跟随蜂以轮盘赌方式按式(7)计

算的概率跟随蜜源。

$$P_i = \frac{\text{fit}_i}{\sum_{k=1}^{\text{SN}} \text{fit}_k} \quad (7)$$

然后同样按式(4)搜索蜜源并通过适应度判断保留哪个蜜源。

(3) 搜索过程中:如果蜜源 X_i 经过 limit 次迭代适应度不变,则放弃 X_i 。其对应的引领蜂转换为侦察蜂。并按式(3)随机生成新蜜源取代 X_i 。

达到最大迭代次数 MCN 后,选取适应度最大的蜜源为最优解,完成寻优。

3.2.3 人工蜂群聚类算法

将 K-means 和 ABC 算法进行融合,算法思想为:先用 ABC 算法得到初始聚类中心,再用 K-means 进行聚类,并用得到的聚类中心更新蜜源,两个算法交替迭代直到达到最大迭代次数具体算法步骤如下:

(1) 设置相关参数。设置聚类数 k 、各蜜蜂数量、最大迭代次数 MCN 及控制参数 limit 。

(2) 使用 ABC 算法进行一次迭代,得到的最优解作为初始聚类中心。

(3) 用 K-means 对初始聚类中心进行迭代,用迭代结果更新蜜源。记录当前最优蜜源。

(4) 如果迭代次数小于 MCN,则转向(2)继续迭代;否则,输出最优解作为聚类的结果。

3.3 基于时间权衡的项目评分预测

3.3.1 传统评分预测方法

传统评分预测方法如下:

$$P_{i,j} = \frac{\sum_{c=1}^k A_{i,c} \times \text{sim}(I_j, I_c)}{\sum_{c=1}^k \text{sim}(I_j, I_c)} \quad (8)$$

式中, I_j 表示项目 j , I_c 表示项目 j 的最近邻居, A_{ij} 表示用户 i 对项目 j 的评分, $\text{sim}(I_j, I_c)$ 为项目间相似性。

3.3.2 时间权衡函数

随着时间的推移,用户的兴趣也会发生变化,然而传统的评分预测方法并没有考虑到这一点,使得推荐的准确性有所降低。本文结合 Ebbinghaus H 遗忘曲线构造时间权重函数 $f(t)$ ^[8]。

首先,引入信息半衰期 T_0 ^[8] 这个概念,顾名思义,它指的是信息从产生到影响力减半所用时间。因此我们可以推出:

$$f(T_0) = (1/2)f(0) \quad (9)$$

然后,定义衰减因子 λ :

$$\lambda = \frac{\ln 0.5}{T_0} \quad (10)$$

时间权重函数如下:

$$f(t) = e^{-\lambda t} \quad (11)$$

上式中, $t = T_{\text{now}} - T_{ij}$, 表示从用户评分到当前时间的时间差。 $f(t)$ 的取值区间为 $(0, 1]$, 它的大小反映了信息的衰减程度 ($f(t)$ 越小, 信息衰减越厉害)。从式(11)可以看出, 评分时间 T_{ij} 的大小直接影响着信息的衰减程度: T_{ij} 越大, $f(t)$ 越大。这表明用户近期的评分对推荐更具意义, 应该赋予更高的权重。

值得注意的是, 信息半衰期 T_0 的选取对推荐有一定的影响, 将在后面的实验中进行研究。

3.3.3 评分预测

利用时间权重函数优化传统的评分预测模型, 最终得到的基于时间权重的项目评分预测公式如下:

$$P_{i,j} = \frac{\sum_{c=1}^k A_{i,c} \times \text{sim}(I_j, I_c) * f(\Delta t_{i,c})}{\sum_{c=1}^k \text{sim}(I_j, I_c) * f(\Delta t_{i,c})} \quad (12)$$

其中, $\Delta t_{i,c}$ 为评分时间与当前时间的时间差。

3.4 算法实现过程

本文将按照以下四个步骤完成推荐过程: (1) 形成用户 - 项目的评分矩阵和评分时间矩阵; (2) 相似性度量; (3) 项目聚类, 计算最近邻居; (4) 预测评分, 并完成推荐。

3.4.1 形成用户 - 项目的评分矩阵和评分时间矩阵

获取实验数据集, 将数据集中的项目评分和评分时间分别保存到评分矩阵 $\mathbf{A}(m, n)$ 和评分时间矩阵 $\mathbf{T}(m, n)$ 中。其中, m 行和 n 列分别代表 m 个用户和 n 个项目; A_{ij} 表示用户 i 对项目 j 的评分, T_{ij} 表示用户 i 对项目 j 进行评分的时间。表示如下:

$$\mathbf{A} = \begin{bmatrix} A_{11} & \cdots & A_{1j} & \cdots & A_{1n} \\ \vdots & & \vdots & & \vdots \\ A_{i1} & \cdots & A_{ij} & \cdots & A_{in} \\ \vdots & & \vdots & & \vdots \\ A_{m1} & \cdots & A_{mj} & \cdots & A_{mn} \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} T_{11} & \cdots & T_{1j} & \cdots & T_{1n} \\ \vdots & & \vdots & & \vdots \\ T_{i1} & \cdots & T_{ij} & \cdots & T_{in} \\ \vdots & & \vdots & & \vdots \\ T_{m1} & \cdots & T_{mj} & \cdots & T_{mn} \end{bmatrix}$$

值得注意的是,并不是每个用户都会对每个项目进行评分,一般情况下评分矩阵是个稀疏矩阵,在此我们对评分矩阵做简单的处理:用项目的平均评分去填充该项目的未评分项.这样不仅缓解了未评分项对推荐结果的影响,还降低了计算复杂度.

与之对应,评分时间矩阵也要进行相应的预处理:用项目的平均评分时间去填充该项目的未评分项.评分时间矩阵的引入是为后续预测评分时使用时间权重函数作准备.

3.4.2 相似性度量

协同过滤中计算项目之间相似性的方法有很多.其中,余弦相似性具有简单易处理的优点,故选择它作为相似性度量方法,按式(13)进行计算:

$$\text{sim}(a, b) = \cos(\vec{a}, \vec{b})$$

$$= \frac{\sum_i^m A_{ia} \times A_{ib}}{\sqrt{\sum_i^m A_{ia}^2} \times \sqrt{\sum_i^m A_{ib}^2}} \quad (13)$$

式中, a, b 表示项目; A_{ia} 表示用户 i 对项目 a 的评分,通过评分矩阵获得.

按照式(13)计算各个项目之间的相似性,将计算结果保存相似性矩阵 $S(n, n)$ 中, n 行和 n 列均代表 n 个项目; S_{ij} 表示项目 i 和项目 j 之间的相似性.显然,该矩阵为一对角矩阵,即 $S_{ij} = S_{ji}$. 矩阵表示如下:

$$S = \begin{pmatrix} 1 & \cdots & S_{1i} & \cdots & S_{1n} \\ \vdots & & \vdots & & \vdots \\ S_{i1} & \cdots & 1 & \cdots & S_{in} \\ \vdots & & \vdots & & \vdots \\ S_{n1} & \cdots & S_{ni} & \cdots & 1 \end{pmatrix}$$

项目相似性矩阵的引入是为了避免重复计算项目间的相似性,减少了计算量,提高了推荐的实时性.

3.4.3 计算最近邻

(1) 项目聚类

以上面得到的项目相似性矩阵为基础,利用人工蜂群聚类算法对项目集进行聚类^[9-10].

(2) 最近邻搜索

聚类结束后,选取与目标项目相似性最高的几个聚类作为最近邻的搜索空间.由于搜索空间大大减小,在线的搜索速度将明显调高,推荐的实时性会得到显著的改善.

最近邻搜索具体过程如下:

①通过项目相似性矩阵获取目标项目与各个聚

类中心的相似性.

②设定相似性阈值,将与目标项目相似性大于阈值的聚类中心所对应的聚类作为搜索空间,通过项目相似性矩阵获取搜索空间内所有项目与目标项目的相似性.

③将相似性最高的前 N 个项目设为目标项目的最近邻居.

值得注意的是,阈值的选取对推荐也有一定的影响.当推荐精度不够时,可适当减小阈值,从而增大搜索空间以提高推荐精度;当推荐实时性不够(推荐系统响应时间过长)时,可适当增大阈值从而减小搜索空间以提高推荐实时性.阈值的选择可通过实际需求来权衡.

3.4.4 完成推荐

按式(12)分别计算所有未评分项目的预测评分,并选取预测评分最高的前 N 个项目推荐给当前用户.至此,完成推荐.

4 实验结果与分析

4.1 实验数据集与评价标准

本实验使用 MovieLens 站点最新提供的小型数据集(更新于 2016.1). 此数据集包含了 700 个用户对 10 000 个电影的 100 000 条评分(0.5~5 分),对该数据集进行预处理,提取用户对项目的评分和评分时间信息保存在项目评分矩阵和评分时间矩阵中.

为了检验本算法的推荐准确性,实验选取 MAE(平均绝对误差)作为评价标准. MAE 将推荐算法对各个项目的预测评分与用户的实际评分进行比较来度量推荐的准确性. MAE 计算公式如下:

$$\text{MAE} = \frac{\sum_{i=1}^n |p_i - q_i|}{n} \quad (14)$$

式中, $\{p_1, p_2, \dots, p_n\}$ 为推荐算法对各个项目的预测评分集合, $\{q_1, q_2, \dots, q_n\}$ 为对应的用户的实际评分集合. 由式(14)可以看出, MAE 越小表示推荐准确性越高.

4.2 信息半衰期 T_0 的选取

文献[8]分别取 T_0 为 30、60、200 进行对比试验. 本文模仿文献[8]进行对比试验,得出了同样的规律:选取较小的半衰期可以获得更好的推荐效果. 故本文选取 30 作为半衰期,后面的实验以此半衰期为基础.

4.3 实验结果分析

本文算法(ABC-ICF Algorithm)与 Sarwar B

等人提出的经典 Item-based 推荐算法[8]及引用较多的文献[5]的算法(K-ICF Algorithm)进行对比. 试验中各参数的取值如下:初始蜂群规模为 40(引领蜂=跟随蜂=20),项目聚类数 $k=50$,控制参数 limit=50,总的迭代次数 MCN=1000. 将与目标项目相似性最高的前 10 个项目聚类作为搜索空间. 最终的项目推荐个数从 10 增加到 50(以 10 为间隔). 三种算法的实验结果数据对比如图 2 所示.

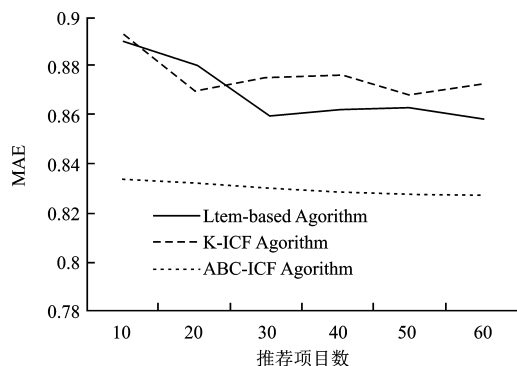


图2 三种算法对比图

从上图可以看出,使用本文算法进行推荐得到的 MAE 值维持在 0.83 左右,明显小于另外两个算法相,由此可以得出,本文算法确实能够有效提高推荐的准确性;此外,通过聚类分析有效的缩小了项目空间,对推荐响应时间的减少也是毋庸置疑的.

5 结束语

为了缓解实时性和推荐准确性对推荐系统所产生的影响,本文引入人工蜂群聚类算法对项目聚类,剔除差异较大的项目的干扰,提高推荐准确性的同时缩小了项目空间,大大提高了实时性;此外,利用时间权重函数对传统的评分预测模型进行了优化,进一步提高了推荐的准确性. 实验也证明了本文算法的有效性

参考文献:

- [1] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]//Proceedings of UAI98. [S. l.]: ACM Press, 1998: 43-52.
- [2] 廖开际, 欧阳解文, 席运江. 基于位置信息的协同过滤推荐算法[J]. 系统工程, 2015, 32(12): 121-125.
- [3] 李荟, 谢强, 丁秋林. 一种基于情景的协同过滤推荐算法[J]. 计算机技术与发展, 2014, 23(10): 42-46.
- [4] Zhang L, Peng L F, Phelan C A. Novel recommendation of user-based collaborative filtering[J]. Journal of Digital Information Management, 2014, 12 (3): 165-175.
- [5] 邓爱林, 左子叶, 朱扬勇. 基于项目聚类的协同过滤推荐算法[J]. 小型微型计算机系统, 2004, 25(9): 1665-1670.
- [6] 张凤娟. 基于粒子群优化的项聚类推荐算法研究[J]. 计算机工程, 2009, 35(23): 178-180.
- [7] 秦全德, 程适, 李丽, 等. 人工蜂群算法研究综述[J]. 智能系统学报, 2014, 9(2): 127-135.
- [8] Ding Y, Li X. Time weight collaborative filtering[C]//Proceedings of the 14th ACM international conference on Information and knowledge management. New York: ACM, 2005: 485-492.
- [9] 曹永春, 蔡正琦, 邵亚斌. 基于 K-means 的改进人工蜂群聚类算法[J]. 计算机应用, 2014, 34(1): 204-207.
- [10] 喻金平, 郑杰, 梅宏标. 基于改进人工蜂群算法的 K 均值聚类算法[J]. 计算机应用, 2014, 34(4): 1065-1069.

作者简介:

郭磊 男, (1991-), 硕士研究生. 研究方向为数据挖掘.
E-mail: 425003928@qq.com.

胡燕 女, (1965-), 教授, 硕士生导师. 研究方向为智能信息检索.