基于 AntStream 用户聚类的协同过滤推荐系统①

王卫平　寇艳艳　(中国科学技术大学 管理学院 安徽 合肥 230026)

摘　要： 系统规模的逐步扩大和用户兴趣的发展变化给传统协同过滤推荐系统带来了实时性减弱和准确性降低的问题。基于 K-Means 用户聚类的协同过滤技术虽然能在一定程度上解决这两个问题，算法本身却带有局部最优的缺陷。在保证实时性的前提下，为克服 K-Means 算法的缺陷，提出使用 AntClass 蚁群算法对用户聚类。同时提出将用户评分看作数据流，利用金字塔时间框架预处理数据，从而体现用户兴趣随时间的变化。于是，将 AntClass 蚁群算法和利用金字塔时间框架预处理过的数据流相结合，最终形成文中的 AntStream 算法。实验表明，AntStream 算法不仅改善了传统协同过滤推荐系统的实时性问题，而且更大程度提高了推荐质量。

关键词： 电子商务；推荐系统；协同过滤；蚁群聚类；金字塔时间框架

Collaborative Filtering Recommender Systems Based on Clustered Users Using AntStream Algorithm

WANG Wei-Ping, KOU Yan-Yan

(Department of Management, University of Science and Technology of China, Hefei 230026, China)

Abstract: Expansion of the scale to the traditional collaborative filtering recommendation systems and changes of users' interest bring problems of decreased accuracy and real-time responsiveness. Collaborative filtering recommender systems based on clustered users using K-Means Algorithm can solve these two problems in some extent, however, with a local optimum defects. Under the premise of ensuring the real-time responsiveness, AntClass algorithm applied to users is proposed to overcome the shortcomings of K-Means algorithm. This paper also proposed to take the users' ratings as a data stream, and use the pyramid time frame for data preprocessing, thus it reflects the change of users' interest with the time. As a result, AntClass algorithm and the data stream filtered by pyramid time frame were combined to form the AntStream algorithm in this article. The experiment result shows that AntStream algorithm has improved not only the real-time responsiveness and also the accuracy to a greater extent.

Keywords: E-commerce; recommender systems; collaborative filtering; ant colony clustering; pyramidal time frame

1　引言

电子商务推荐系统的诞生旨在解决信息过载问题，提供个性化推荐。从现有的推荐系统来看，它能有效的提高电子商务网站的交叉销售能力、提高客户对电子商务网站的忠诚度，并且将网站的浏览者转变为购买者，从而实现电子商务推荐系统的初衷。

协调过滤推荐系统是电子商务推荐系统中应用最成功的一种。然而，系统规模的扩大和用户兴趣的变化，使它在实时性和推荐准确性上面临挑战。基于 K-Means 用户聚类的两阶段协同过滤推荐系统能在一定程度上解决这两个问题，但 K-Means 算法的局部最优性限制了推荐准确性的提高程度。

针对上述问题，本文提出将用户评分信息看作数

① 收稿时间:2010-04-05;收到修改稿时间:2010-05-31

据流[1]，利用金字塔时间框架对数据进行筛选，使不同时间戳上的数据具有不同的利用率；再使用蚁群聚类算法—AntClass[2]对用户进行聚类，避免局部最优，从而形成 AntStream 算法。实验表明，该算法在保证 K-Means 用户聚类实时性的基础上能更大程度提高推荐质量。

2 协同过滤技术

2.1 传统的协同过滤技术

协同过滤推荐技术通过相似用户对某项目的评分来预测目标用户对该项目的评分。算法[3]分为两步：

1) 根据评分矩阵计算所有用户之间的相似度，最常用的计算相似度的方法为基于关联的方法和基于余弦距离的方法。具体的计算公式为(1)和(2)：

$$sim(x,y)=\frac{\sum_{s\in S_{xy}}(r_{x,s}-\overline{r}_x)(r_{y,s}-\overline{r}_y)}{\sqrt{\sum_{s\in S_{xy}}(r_{x,s}-\overline{r}_x)^2\sum_{s\in S_{xy}}(r_{y,s}-\overline{r}_y)^2}} \tag{1}$$

$$sim(x,y)=\cos(\vec{x},\vec{y})=\frac{\sum_{s\in S_{xy}}r_{x,s}r_{y,s}}{\sqrt{\sum_{s\in S_{xy}}r_{x,s}^2}\sqrt{\sum_{s\in S_{xy}}r_{y,s}^2}} \tag{2}$$

2) 当目标用户需要推荐的时候，在整个用户空间上寻找相似度最高的用户(即最近邻)，再使用最近邻的评分来预测该用户在评分矩阵上的空缺值。计算评分的方法[3]有：均值法；相似度加权法；平均归一化法。其中相似度加权法是最常用的一种：

$$r_{c,s}=k\sum_{c'\in\hat{C}}sim(c,c')\times r_{c',s} \tag{3}$$

评分预测结束后，即可将评分最高的一个或多个项目推荐给目标用户，从而完成推荐过程。

协同过滤技术是应用最成功的一种推荐技术，但系统规模的扩大，使得在整个用户空间上搜索最近邻极大的影响了实时响应性。另外，随着用户兴趣的发展变化，在整体历史数据上的预测评分也大大降低了推荐的准确性。

2.2 基于 K-Means 用户聚类的协同过滤推荐系统

基于 K-Means 用户聚类的协同过滤技术[4]分为离线和在线两个部分。离线部分，利用 K-Means 算法将相似用户聚类成簇，并计算簇内用户之间的相似度。在线部分，当目标用户到达时，判断所属簇，然后在簇中寻找最近邻。从而，与传统协同过滤技术相比，实时性得到了提高。

K-Means 算法是一种经典的聚类算法，它收敛速度快，但是必须提供初始划分。一旦初始划分不够准确，容易形成局部最优。所以，解决 K-Means 算法中的局部最优问题，能使用户聚类更合理，从而提高推荐系统的准确性。

3 基于AntStream用户聚类的协同过滤技术

3.1 蚁群聚类

蚁群聚类[5]起源于蚂蚁搬运食物、蚁卵的行为[6]，其中最经典的 LF 算法的基本思想是[7]：将需要聚类的对象随机放置在 2 维网格上，人工蚁群则在网格上移动。随着对象与环境相似度的增加，它们拾起的概率减少，放下的概率增加。蚁群的共同作用一段时间后，网格上则形成数个相似对象集中在一起的簇，从而达到聚类的效果。

LF 算法聚类数目往往偏高，并且收敛速度慢。AntClass 算法[2]针对这些问题进行了改进，将 K-Means 算法融入其中，从而更适用于现实对象的聚类。所以本文采用 AntClass 算法对用户进行聚类，使它在克服 K-Means 算法局部最优缺陷的前提下，更大程度的发挥蚁群聚类的优势。

3.2 金字塔时间框架

一个具有一定规模的推荐系统，交易的更新使之前的数据逐渐失去预测评分的价值。所以形成数据流后，我们有必要筛选它们[8]，找出最有价值的数据来预测评分。

金字塔时间框架[9]是一种很好的筛选数据的方法，它通过对不同时间戳上的数据按金字塔模式存储而实现。具体方法为：将带有时间戳标签 T(T=1…N) 的数据流根据 T 存储在不同粒度的次序上，次序分别为 $1\cdots\log_a(T)$ (即所有能被 a^i 整除的时间戳上的数据就被存储在次序 i 上)。并且，在每个次序上，只有最新的 (a^l+1) 个数据被存储。这样，存储的数据总量为 $(a^l+1)\cdot\log_a(T)$，相对比较稳定，并且离当前时间近的数

据利用率相对高，而离当前时间远的相对低。表 1 为数据的存储方式，当 $\alpha=2$，I=2 时，储存的时间戳为 55、54、53、52、51、50、48、46、44、40、36、32、24、16。

表 1　金字塔时间框架数据存储方式

次序($a=2$)	数据时间戳($l=2$)
0	55 54 53 52 51
1	54 52 50 48 46
2	52 48 44 40 36
3	48 40 32 24 16
4	48 32 16
5	32

3.3 基于 AntStream 用户聚类的协同过滤算法

本文提出使用金字塔时间框架对评分数据流进行筛选，再使用 AntClass 算法对用户聚类，两者结合形成 AntStream 算法。算法分为三部分：初始化部分；离线部分；在线部分。

3.3.1 初始化部分

初始化部分将系统内不断产生的评分数据看成数据流，用 T=1…N 作为时间戳标记，对于不同时间戳上的数据使用金字塔时间框架进行存储。这样，使得离当前时间比较近的数据利用率更高，而比较远的数据利用率相对低。

3.3.2 离线部分

离线部分的任务有三个：将初始化后的评分数据形成评分矩阵，并对评分矩阵的空缺项进行预处理；利用 AntClass 对用户聚类形成簇；计算簇内用户间的相似度(本文采用基于关联的相似度方法)。

其中核心部分为用 AntClass 对用户聚类，分为四步：

1) 蚁群聚类形成初始划分：

初始化数据对象和蚂蚁；

do{ 随机选择一只蚂蚁；

随机移动这只蚂蚁；

if(蚂蚁未携带数据对象)

按拾起条件在邻居网格内拾起数据对象；

if(蚂蚁携带数据对象)

按放下条件在邻居网格内放下数据对象；

}while(循环次数未到)；

a) 整体流程如下：首先将 N 个数据对象(即用户的评分向量)和 P 只蚂蚁随机分布在二维环形网格上；然后随机选择一只蚂蚁在二维网格上移动，如果蚂蚁未携带数据对象，则检测能否拾起一个对象后完成本次循环，否则继续移动直到拾起为止；如果蚂蚁已携带数据对象，先检测该蚂蚁记忆中是否有簇可放下该数据对象，有则蚂蚁直接移至该簇所在网格，否则检测邻居网格能否放下对象，能则放下后完成本次循环，否则继续移动后检测，直至放下为止。

b) 拾起条件：若邻居网格有一个对象，则按概率拾起这个对象；若邻居网格有两个对象，则按概率拾起任意一个，使原来的簇消失；若含两个以上对象，则按条件拾起簇中离中心最远的对象。

c) 放下条件：如果邻居网格为空，则按概率直接放下对象；如果网格中有一个对象，并且对象之间的距离满足创建新簇的条件，则放下对象，形成新簇；如果邻居网格中本来就含有一个簇，则测试新对象到中心的距离是否小于该簇的最大半径，是则将新对象放入该簇。另外，蚂蚁会记录下每次成功放置对象的簇，以便于下次快速放置其它对象。

2) 以上面的初始划分为基础，对整体数据进行 K-Means 聚类，以加速收敛并处理蚁群聚类留下的孤立点。

3) 针对第一次蚁群聚类后簇数目过高[2]的问题，将簇作为数据对象，进行第二次蚁群聚类，以控制簇的数目。

4) 第二次在整体数据范围上使用 K-Means 算法加速收敛，形成最终的聚类结果。

3.3.3 在线部分

当目标用户需要在线推荐时，算法在其所属簇内找出最近邻后预测评分，再进行推荐。本文使用相似度加权法来预测评分，并形成 top-N 推荐。

4 实验结果及其分析

4.1 数据集

实验数据集是MovieLens站点收集的历时7个月的数据，其中包含943个独立用户对1682部电影的100000次评分(1分-5分)。每条评分带有时间戳，且每部电影带有项目概要信息(电影名称、发行日期等)。本文依据时间戳，将前70%作为训练集，后30%作为测试集。另外，由数据集获得的评分矩阵存在稀缺性，本文使用项目概要信息对评分矩阵进行预处理，得到一个适用于蚁群聚类的评分矩阵。

4.2 度量标准

为度量预测评分的准确性，本文使用平均绝对偏差MAE计算预测评分与用户实际评分之间的偏差。假设预测评分集为 $\{p_1, p_2, ..., p_N\}$，用户的实际评分集为 $\{q_1, q_2, ..., q_N\}$，则:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N} \quad (4)$$

4.3 精确性检验

实验中，第一次蚁群聚类分别采用30万和40万次循环,寻找最近邻的条件设置为相似度大于0.75，0.8，0.85，0.9。计算K-Means用户聚类和AntStream用户聚类后预测评分的MAE可得图1和2。其中AntStream用户聚类中30万次循环形成37个簇，40万次循环形成32个簇。

由图可见，本文提出的AntStream用户聚类的MAE在大部分情况下要明显低于K-Means用户聚类的MAE。另外，采用40万次循环得到的结果更优于30万次循环，这是因为40万次循环使聚类收敛程度更大，簇的形成情况更成熟。

4.4 实时性检验

为了检验算法的实时性，本文将传统协同过滤技术向用户推荐项目所需时间(在整个用户空间上寻找最近邻；利用相似度和最近邻的评分预测目标用户的评分；产生推荐)与AntStream算法在线推荐项目的时间(寻找目标用户所在簇；在簇内寻找最近邻；利用簇内用户间的相似度预测评分；产生推荐)作比较。图3为给所有用户推荐项目的实验结果，AntStream算法比传统算法节省了近三分之一的时间。

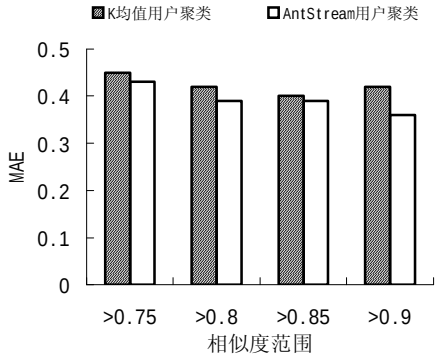

图1 30万次循环MAE比较图

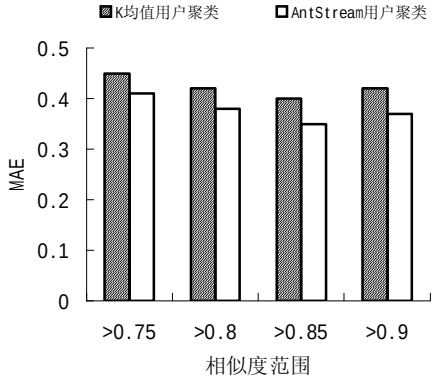

图2 40万次循环MAE比较图

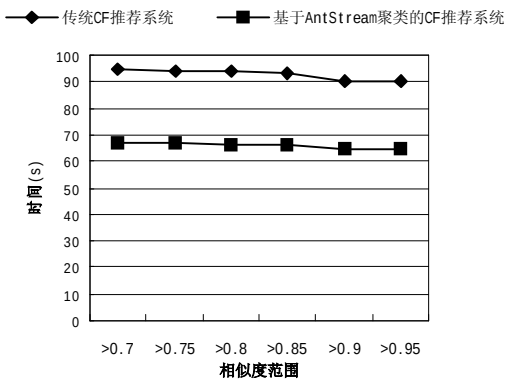

图3 40万次循环实时性比较图

5 结语

本文提出的基于AntStream用户聚类的协同过滤推荐技术,将金字塔时间框架和AntClass蚁群聚类融入协同过滤推荐系统，既克服了协同

过滤推荐系统推荐准确性和实时性的限制，又克服了基于 K-Means 用户聚类的局部最优缺陷。由实验结果可以看出，该算法能在保证实时性的前提下更大程度提高推荐质量。当然，基于 AntStream 用户聚类的协同过滤技术还存在一些问题需进一步研究，例如金字塔时间框架在时间跨度大，却存在严重稀疏性的推荐系统中如何标记数据流等。

参考文献

1 Xiaodan Song, Yun Chi, Koji Hino, Belle L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. Carey Williamson, ed. Proceedings of the 16th international conference on World Wide Web. New York: ACM, 2007:191－200.

2 Monmarché N, Slimane M, Venturini G. Antclass: discovery of clusters in numeric data by an hybridization of an ant colony with the Kmeans algorithm. No 213: Laboratoire d'Informatique Universite de Tours, 1999:E3i.

3 Gediminas Adomavicius, Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 2005,17(6):734－749.

4 李涛,王建东,叶飞跃,冯新宇,张有东. 一种基于用户聚类的协同过滤推荐算法.系统工程与电子技术, 2007, 29(7):1177－1178.

5 Boryczka U. Finding groups in data: Cluster analysis with ants. Applied Soft Computing Journal. 2009,9(1): 61－70.

6 Ashish Ghosh, Anindya Halder, Megha Kothari, and Susmita Ghosh. Aggregation pheromone density based data clustering. Information Science, 2008,178(13): 2816－2831.

7 Lumer E, Faieta B. Diversity and adaption in populations of clustering ants. Dave Cliff, ed. Proc of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3. Brighton: Cambridge: MIT Press, 1994:501－508.

8 杨怀珍,丛晓琪,刘枚莲. 基于时间加权的个性化推荐算法研究.计算机工程与科学, 2007,31(6):126－128.

9 Aggarwal CC, Jiawei Han, Jianyong Wang, Philip S. Yu. A framework for clustering evolving data streams. Johann Christoph Freytag, ed. Proc of the 29th VLDB Conference. Berlin: Morgan Kaufmann Publishers, 2003: 81－92.