第 33 卷 第 6 期 计 算 机 工 程 2007 年 3 月
Vol.33 No.6 Computer Engineering March 2007
·软件技术与数据库· 文章编号：1000—3428(2007)06—0072—02 文献标识码：A 中图分类号：TP311

线性逐步遗忘协同过滤算法的研究

郑先荣，曹先彬
(中国科技大学计算机科学与技术系，合肥 230027)

摘 要：协同过滤系统是目前最成功的一种推荐系统，但是传统的协同过滤算法没有考虑用户兴趣变化问题，导致用户兴趣发生变化时的推荐质量较差。该文借鉴心理学遗忘规律，提出了线性逐步遗忘协同过滤算法。该算法依据评价时间线性逐步减小每项评分的重要性。基于 MovieLens 数据集的实验结果表明，该算法在准确性方面优于传统的协同过滤算法。
关键词：协同过滤；兴趣变化；线性逐步遗忘

Research on Lineal Gradual Forgetting Collaborative Filtering Algorithm

ZHENG Xianrong, CAO Xianbin
(Department of Computer Science, University of Science and Technology of China, Hefei 230027)

【Abstract】 Collaborative filtering (CF)is the most successful recommended system to date, but traditional CF algorithm does not consider the problem of drifting users' interests which often results in poor recommendation when users' interests are changed. This paper develops a lineal gradual forgetting CF algorithm inspired by the forgetting law of psychology and it diminishes the importance of each rate with time. The experiment using MovieLens dataset shows that the new algorithm is more accurate than traditional CF algorithm.
【Key words】 Collaborative filtering; Interest drift; Lineal gradual forgetting

推荐系统是一种个性化的信息过滤技术[1]，它被用来预测某个特定用户是否会喜欢某个特定的商品(预测问题)，或被用来确定N件用户感兴趣的商品(TOP-N推荐问题)。目前最成功的推荐系统是协同过滤系统，很多的电子商务网站均使用了该技术，它的优点是能够基于信息的质量和品位进行过滤并能够推荐用户意想不到的有用信息[2]。

建立协同过滤推荐系统主要有基于用户的(User-based)和基于项目的(Item-based)两种方法[3,4]。基于用户的方法利用兴趣相似的邻居用户的评价，为当前用户产生推荐。基于项目的方法通过分析历史信息确定不同项目之间的关系，然后利用这些关系推荐项目。该方法基于这样的认识：如果用户曾经对某些项目感兴趣，那么与之相类似的项目用户也会感兴趣。实际上，用户兴趣的变化经常导致推荐的质量较差。因此，本文提出了线性逐步遗忘协同过滤算法，实验结果表明该算法有效地提高了用户兴趣变化情况下的推荐质量。

1 相关工作

Tapestry[5]是第 1 个使用协同过滤思想的系统，该文作者认为人们应该彼此合作完成信息过滤工作。为处理用户变化的兴趣，Koychev[6]引入了逐步遗忘的思想并使用了一个线性遗忘函数。Kular[7]建议使用一个核函数来学习用户的兴趣。

用户兴趣不但是多方面的，而且还是动态变化的，因此跟踪和学习用户兴趣是一个最基本且难以解决的问题[8]。通过组合基于用户的和基于项目的协同过滤方法，基于相似项目的邻居用户协同过滤算法[9]能够处理用户多兴趣下的个性化推荐问题。

2 逐步遗忘 CF 算法

2.1 用户兴趣变化问题

传统的协同过滤算法利用兴趣相似的邻居用户对某项目兴趣的大小，预测当前用户对该项目的喜好程度。但是通常的算法没有考虑用户兴趣变化问题，从而影响了算法的准确性。协同过滤算法需要为当前用户选择与他最近兴趣相似的邻居用户，同时淘汰仅与他过去兴趣相似的邻居用户。

因此，本文对传统的协同过滤算法进行了改进。先赋予每项评分一个按时间衰减的权重，即最近的评分权重大，过去的评分权重小，然后按加权后的评分确定用户间的相似度。

2.2 逐步遗忘 CF 的思想

为了后文表述方便，先定义几个符号和函数。

定义 1 starttime：参照时间，比如 Linux 系统中使用的 1970 年 1 月 1 日。

定义 2 ratetime：用户对项目的绝对评分时间，即实际评分时间。

定义 3 t：用户对项目的相对评分时间， 即绝对评分时间与参照时间的间隔时间。

定义 4 maxtime：最大间隔时间，即 maxtime = max(ratetime - starttime)。

定义 5 mintime：最小间隔时间，即 mintime = min(ratetime - starttime)。

定义 6 f(t)：遗忘函数，其中 mintime< t <maxtime，该函数单调递增。

实现评分重要性按时间衰减的具体做法是：在皮尔森相关系数中，用$R_{j,i}\times f(t)$代替$R_{j,i}$确定用户间的相似度。遗忘函数 f(t)的作用是增加最近评分的重要性和降低过去评分的重要性。

作者简介：郑先荣(1979－)，男，硕士生，主研方向：协同过滤，数据挖掘；曹先彬，副教授、博士
收稿日期：2006-05-10 **E-mail：**xrzheng@mail.ustc.edu.cn

遗忘函数 f(t)的选择非常关键，它的遗忘能力直接影响推荐系统的性能，f(t)的遗忘速度快，系统学习用户兴趣的过程就快，反之则慢。

推荐系统是一种 Web 智能系统，为了提高它的智能性，推荐系统需要从机器学习、数据挖掘和心理学等不同学科借鉴有价值的思想。

推荐系统中遗忘的作用符合心理学的遗忘理论。心理学认为，遗忘是一个自然的、必要的心理现象[10]，它对人的记忆活动除有消极作用外也有着重要的积极作用，遗忘可去除大脑中的无用信息，从而可腾出空间容纳其它更有价值的信息。

受此启发，本文认为：为了迅速捕捉用户兴趣的变化，在选择遗忘函数时应充分借鉴心理学关于人的遗忘理论的成果。遗忘规律表明：人的遗忘过程是逐步发生的。故此本文使用线性逐步遗忘策略。

2.3 线性逐步遗忘 CF 算法

(1)线性逐步遗忘函数

$$f(t)=m\times\frac{t-mintime}{maxtime-mintime}+1-m,$$
$$mintime\le t\le maxtime, 0\le m\le 1, 1-m\le f(t)\le 1$$

其中，参数 m 为 f(t)的一阶导数，它反映了 f(t)的遗忘能力。m 越大，f(t)遗忘得越快，反之越慢。当 m=1 时，f(t)对用户评分进行完全的线性遗忘；当 0<m<1 时，进行部分的线性遗忘；当 m=0 时，未进行线性遗忘。m 值的设置受推荐系统中用户兴趣变化速度的影响，若用户兴趣变化快，则 m 的值应大些，反之应小些。其它参数的含义同 2.2 节。

(2)算法描述

线性逐步遗忘 CF 算法的输出数据是 N 个推荐的项目，输入数据包括：m 个用户对 n 个项目的评分和相应的评分时间 t，邻居用户大小 k。算法步骤如下：

step1 利用皮尔森相关系数计算用户间的相似度。

$$w(a,j)=\frac{\sum_{i\in CRI}(R_{a,i}\times f(t)-\overline{R_a})(R_{j,i}\times f(t)-\overline{R_j})}{\sqrt{\sum_{i\in CRI}(R_{a,i}\times f(t)-\overline{R_a})^2\sum_{i\in CRI}(R_{j,i}\times f(t)-\overline{R_j})^2}} \quad (1)$$

其中，a是当前用户，j是其他用户，w(a,j)是a和j的相似度，CRI是a和j共同评分的项目集，f(t)为线性逐步遗忘函数，$R_{a,i}$是a对项目i的评分，$R_{j,i}$是j对项目i的评分。$\overline{R_a}$ 在原皮尔森相关系数中表示a评分过的所有项目的均值，但线性逐步遗忘CF算法用f(t)赋予每个评分一个权重，缩放了原始评分的大小，因此为了准确计算用户间的相似度，本文中的 $\overline{R_a}$ 表示a在CRI中所有加权评分的均值；同理，$\overline{R_j}$ 表示j在CRI中所有加权评分的均值。

step2 将与a相似度最高的前k个用户作为它的邻居用户 N_a。

step3 利用式(2)综合邻居用户的评价并预测 a 对项目 i 的评分。

$$P_{a,i}=\overline{R_a}+\frac{\sum_{j\in N_a}w(a,j)(R_{j,i}-\overline{R_j})}{\sum_{j\in N_a}w(a,j)} \quad (2)$$

step4 将预测评分最高的前 N 个项目作为推荐的项目。

(3)算法说明

由于用户兴趣的变化，推荐系统中旧评分的积极作用不大。线性逐步遗忘协同过滤算法依据评价时间线性逐步减小每项评分的重要性，因此它在准确性方面应高于传统的协同过滤算法。

实验结果表明，当用户兴趣变化时线性逐步遗忘协同过滤算法在准确性方面明显优于传统的协同过滤算法。

3 实验评价

3.1 数据集

MovieLens 是著名的协同过滤推荐系统，目前它提供的电影超过 5 000 部，用户超过 70 000 人。MovieLens 数据集含有本文必需的时间属性。为了避免数据稀疏问题的发生，本文在该数据集中提取记录时，要求每个用户至少评价过 50 部电影，每部电影至少被 50 个用户评价过。最终提取的数据集包括 51 个用户和 264 部电影，共 4 666 条记录。本文将其中 90%的数据作为训练集，剩下的 10%作为测试集。

3.2 评估标准

与大多数文献一样，本文以预测值和真实值之间的平均绝对误差(Mean Absolute Error，MAE)度量算法的精确度。本文选择的预测项目都是用户最近评分过的，目的是测试算法能否跟踪用户的最近兴趣。

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

3.3 实验结果

本文从不同角度比较了非线性逐步遗忘协同过滤算法和传统的协同过滤算法。

(1)随机选择若干用户，比较不同用户下两种算法的 MAE 值，其中邻居大小 k=10，遗忘速度 m=0.6。实验结果表明，线性逐步遗忘协同过滤算法在准确性方面优于传统的协同过滤算法，如图 1 所示。从图 1 中可以看出：对于不同用户，总体上线性逐步遗忘协同过滤算法的 MAE 值小于传统的协同过滤算法，但是对于个别用户(如 UserID=62)，前者的 MAE 值却比后者大，原因可能是这些用户的兴趣变化较慢，不适合采用线性逐步遗忘策略。

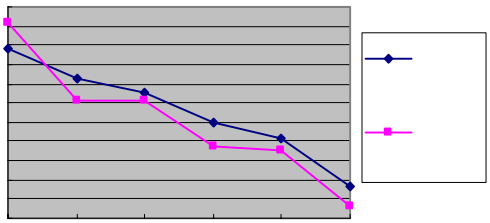

图 1 不同用户下的推荐算法精度比较

(2)对于上面的用户(UserID=127)，比较不同邻居数量下两种算法的 MAE 值，其中遗忘速度 m=0.6。实验结果表明，线性逐步遗忘协同过滤算法在准确性方面优于传统的协同过滤算法，如图 2 所示。

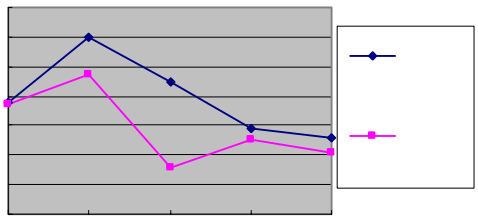

图 2 不同邻居大小下的推荐算法精度比较

从图 2 中可以看出：对于不同的邻居大小 k，线性逐步

(下转第 82 页)