面向电影推荐的时间加权协同过滤算法的研究

兰 艳[1] 曹芳芳[2]
(大连东软信息学院软件工程系 大连 116023)[1] (大连理工大学软件学院 大连 116600)[2]

摘 要 针对协同过滤算法的信息过期问题,提出一种改进的时间加权协同过滤算法(NTWCF)。考虑信息随时间推移导致信息影响力变化的因素,在信息半衰期的启发下,引入信息保持期的概念,通过在最近邻查找阶段和预测评分阶段采用一种新颖的时间加权函数为项目上的评分赋予不同的时间权重。电影数据集上的实验结果表明,它在一定程度上大幅度提高了预测推荐的准确性。接着,针对算法的实时性问题,利用时间加权的项目聚类优化NTWCF算法,提出综合时间权重和项目聚类的协同过滤算法(TWICCF),对评分信息时间加权后再对项目K-means聚类,在为目标项目查找最近邻时只在若干聚类构成的项目集中进行。相比NTWCF算法,TWICCF算法在推荐准确度和实时性上均取得了显著的提升。

关键词 协同过滤,电影推荐,信息半衰期,信息保持期,时间加权,项目聚类
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.04.060

Research of Time Weighted Collaborative Filtering Algorithm in Movie Recommendation

LAN Yan[1] CAO Fang-fang[2]
(Department of Software Engineering, Dalian Neusoft University of Information, Dalian 116023, China)[1]
(School of Software Technology, Dalian University of Technology, Dalian 116600, China)[2]

Abstract In order to deal with the outdated information problem of collaborative filtering algorithm, a new time weighted collaborative filtering algorithm (NTWCF) was proposed. Considering the influence of this change, it introduced the concept of information retention period which inspired by the information half-value period. At the stages of nearest neighbor searching and predictive scoring, this paper used a novel time weighted function to put time weight to the user's score. The experimental results of movie data sets show that it can greatly improve the accuracy of predicted ratings. Then, in order to improve the real-time speed of algorithm, this paper used time weighted item clustering algorithm to optimize NTWCF, and put forward a collaborative filtering algorithm based on time weight and item clustering (TWICCF). It used K-means to cluster the items based on time weighted scores, and then searched the nearest neighbors of the target item in the set of some clusters. TWICCF algorithm achieves significant improvements in both recommendation accuracy and real-time than NTWCF algorithm.

Keywords Collaborative filtering, Movie recommendation, Information half-value period, Information retention period, Time weighted, Item clustering

1 引言

在当前的推荐算法研究中,协同过滤算法是研究者最热衷且推荐效果最好的算法之一。因此,协同过滤算法被广泛应用在互联网各行各业中,如亚马逊、京东商城、淘宝、美丽说、当当网等电子商务网站中。1992年,在Tapestry系统[1]中,协同过滤的概念被首次提出,之后,新闻过滤系统GroupLens[2]、电影推荐系统MovieLens、笑话推荐系统Jster以及音乐推荐系统Ringo等相继出现。2000年,电子商务系统Amazon开始为用户提供推荐服务[3],当用户购买一件自己中意的商品时,Amazon会推荐其可能中意的其他商品,基于项目的协同过滤推荐算法首次被应用到实际系统中。

协同过滤算法虽然已经得到了广泛应用,但仍然不可避免地存在着数据稀疏性[4-5]、冷启动[6]、实时性差[5]、信息过期等问题[7]。为了达到更加高质量的预测和推荐,各种各样的改进方案被不断提出。其中,针对数据稀疏性问题,最常见的为使用维数约减技术对原始数据进行压缩[8],此外也可以直接移除无关紧要的用户或项目来降低用户-项目矩阵的维数;针对冷启动问题,有研究者[6,9]指出在初期可以根据用户属性或项目属性寻找其邻居,即结合基于内容的推荐;实时性问

到稿日期:2015-08-01 返修日期:2016-01-05 本文受面上基金:在线背包问题的算法和分析(11101065)资助。
兰 艳(1981—),女,硕士生,副教授,主要研究方向为算法优化,E-mail:lanyan@neusoft.edu.cn;曹芳芳(1991—),女,硕士生,主要研究方向为个性化推荐。

题也可以通过维数约减技术进行缓解，其中基于聚类技术的推荐算法[10-13]研究得最为广泛；针对信息过期问题，前人提出一系列推荐算法，算法考虑到信息影响力的变化，引入非线性遗忘函数[7,14-17]。

上述的改进算法在一定程度上都致力于解决传统协同过滤算法所面临的问题，但是通过有效解决稀疏性问题、实时性问题和信息过期问题来提高推荐质量，仍然存在巨大的研究空间。因此，为了解决信息过期的问题，本文首先提出改进的时间加权的协同过滤算法(New Time Weighted Collaborative Filtering，NTWCF)，认为虽然随着时间的推移信息的影响力非线性衰减，但在一段恒定的时间内影响力不会发生显著变化，将信息保持不变的时间窗融合到衰减函数中，利用信息的半衰期[7]和本文提出的信息的保持期概念，生成改进的时间加权函数，并将其引入到传统的余弦相似性计算中，以提高项目之间相似性计算的精确性，从而达到更好的推荐效果。采用电影数据集进行实验，可知 NTWCF 算法大幅度提高了算法预测的准确性。

同时，为了保证大数据环境下算法的实时性，本文利用时间加权的项目聚类技术(Time Weighted Item Clustering，TWIC)优化 NTWCF 算法，提出基于时间权重和项目聚类的协同过滤优化算法(Collaborative Filtering Based on Time Weight and Item Clustering，TWICCF)。换句话说，本文在对项目聚类时考虑信息影响力随时间改变的因素，采用改进的时间加权函数对评分信息加权后执行项目聚类。实验结果表明，TWICCF 明显提高了算法的执行效率，并在合适的聚类下表现出更好的推荐精准度。

2 传统协同过滤算法

2.1 协同过滤概述

协同过滤的思想概括起来就是为用户推荐与其志趣相投的其他用户感兴趣的项目。如图 1 所示，协同过滤算法一般分为 3 个步骤：首先收集用户的偏好信息，形成用户评分数据库；然后找到与其兴趣相似的用户形成其近邻用户集；最后根据近邻用户对项目的偏好来估算目标用户对该项目的偏好程度，根据评估结果进行推荐。协同过滤算法突破基于内容的推荐的局限性，不依赖于项目和用户本身的属性及特征，而是从更具价值的用户历史数据入手，利用用户之间的隐性关系，使得推荐结果更加符合用户的喜好，所以算法的研究得到了高度重视。

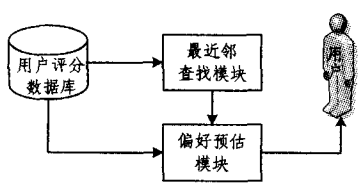

图 1 协同过滤推荐模型

目前协同过滤推荐算法大致可以分为两大类，分别为基于记忆的和基于模型的。其中，基于记忆的协同过滤算法根据相似性计算对象的不同，又可以分为基于用户的和基于项目的两种算法。

2.2 传统基于项目的协同过滤

基于项目的协同过滤技术认为用户对不同项目的喜好存在相似性，用户一般更加倾向于购买那些与他已经购买的项目相似的项目。该算法由于可扩展性强、稳定性强等特点，已经表现出越来越明显的优越性。

该算法总体上可以分为 3 个步骤：预处理评分信息、构造最近邻居集、预测评分及推荐。

2.2.1 预处理评分信息

对信息进行预处理，得到如表 1 所列的评分矩阵 R，用户集合 $U=\{U_1,U_2,\cdots,U_m\}$，m 表示 m 个用户，项目集合 $I=\{I_1,I_2,\cdots,I_n\}$，n 表示 n 个项目，$R_{i,j}$ 表示 U_i 对 I_j 的评分。

表 1 用户-项目评分矩阵 R

	I_1	I_2	I_j	…	I_n
U_1	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,n}$
U_2	…	…	…	…	…
U_i	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,n}$
…	…	…	…	…	…
U_m	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,n}$

表 1 中分值的高低表示用户对相应项目的喜好程度，数值越高代表用户对该项目的兴趣度越高，反之则越低。

2.2.2 构造最近邻居集

对于目标用户 u，我们的目的是为其未评分项目进行预测评分。这里，以 u 的未评分项目 I_i 为例进行说明。首先，计算目标项目 I_i 与集合 I 中其他项目的相似性；然后，取出其中相似性度最高的 k 个项目形成 I_i 的最近邻居集 $NN_i=\{I_1,I_2,\cdots,I_k\}$。计算相似度的方法有多种，其中最基本的有以下 3 种：余弦相似性、相关相似性和修正的余弦相似性[18-19]。

1)余弦相似性

余弦相似性方法类似于两个评分向量之间的夹角，夹角越小说明相似性越高，如式(1)所示。若评分矩阵 R 中评分值为空，则记为 0。

$$sim(i,j)=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2\cdot\|\vec{j}\|_2}=\frac{\sum\limits_{u\in U_{ij}}R_{u,i}R_{u,j}}{\sqrt{\sum\limits_{u\in U_i}R_{u,i}^2\sum\limits_{u\in U_j}R_{u,j}^2}} \tag{1}$$

其中，将 R 视为向量空间，$\vec{i}$ 和 $\vec{j}$ 分别为所有用户对项目 i 和 j 的评分向量，U_{ij} 代表对 i 和 j 均有评分的用户集。

2)相关相似性

相关相似性用来衡量两个评分向量间的线性相关性，即利用 Pearson 相关系数来衡量两个项目之间的相似性。

$$sim(i,j)=\frac{\sum\limits_{u\in U_{ij}}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum\limits_{u\in U_{ij}}(R_{u,i}-\bar{R}_i)^2\sum\limits_{u\in U_{ij}}(R_{u,j}-\bar{R}_j)^2}} \tag{2}$$

其中，$\bar{R}_i$ 和 $\bar{R}_j$ 分别为所有 m 个用户对 i 和 j 的评分的平均值。

3)修正的余弦相似性

采用基本的余弦方法度量相似性有一个严重的缺点，即忽略了不同用户对评分标准的理解存在差异。为了弥补这个缺点，修正的余弦相似性从每个评分中减去对应用户的平均评分后进行计算。

$$sim(i,j)=\frac{\sum_{u\in U_{ij}}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U_{ij}}(R_{u,i}-\bar{R}_u)^2\sum_{u\in U_{ij}}(R_{u,j}-\bar{R}_u)^2}} \tag{3}$$

其中，$\bar{R}_u$ 代表 u 对所有项目评分的平均值。

2.2.3　预测评分及推荐

得到 I_i 的最近邻居集 NN_i 后，通过目标用户 u 对 NN_i 中的项目的评分数据预测 u 对 I_i 的评分 $P_{u,i}$。

$$P_{u,i}=\bar{R}_i+\frac{\sum_{j\in NN_i}sim(i,j)(R_{u,j}-\bar{R}_j)}{\sum_{j\in NN_i}|sim(i,j)|} \tag{4}$$

其中，$\bar{R}_i$ 和 $\bar{R}_j$ 分别表示用户对 i 和 j 的平均评分，$sim(i,j)$ 表示项目 i 和 j 之间的相似度。

按照以上方法对目标用户 u 的所有未评分项目进行预测评分，并选取其中预测分数较高的前 N 个项目推荐给目标用户 u。

3　基于时间权重的协同过滤

随着大数据时代的到来，用户信息和项目信息日益增加，且变更速度越来越快，传统的协同过滤算法忽略了信息价值随着时间衰减对推荐质量的影响，也就是信息过期的问题。所以，近几年来，基于时间权重的协同过滤受到了国内外学者的高度关注，且具有较大的研究空间。

3.1　相关研究

信息过期问题，本质上是由于用户的兴趣会随着时间推移产生变化，而不是保持不变的。比如，用户 A 在两年前喜欢青春类电影，为电影《那些年，我们一起追的女孩》打了 5 分，但是随着时间的推移，他开始更加着迷于科幻电影，也为电影《前目的地》给出 5 分的评价。如果我们现在为用户 A 做推荐时不考虑用户兴趣会随时间改变的因素，则二者在推荐过程的权重一样，所以两种类型的电影会以相同的概率被推荐，这违背了用户 A 目前对科幻电影喜好程度更高的事实，导致推荐效果不够理想，用户体验变差。

为此，国内外的学者展开研究，力求剔除信息过期问题给推荐准确度带来的影响。1998 年，GrabtreeI 和 Soltysiak[20] 提出用户应该只对自己近期访问的项目感兴趣，而较长时间前的访问记录对现在的推荐没有意义。2000 年，Koychev 和 Schwab[21-22] 认为用户近期的评分信息比很长时间以前的评分信息价值更高，故将非线性遗忘函数加入推荐算法中。2005 年，Ding 和 Li[7] 考虑对用户的评分进行时间加权，为其赋予时间衰减的权重，目的是降低过时信息的重要性，从而提出时间加权的协同过滤算法。2007 年，邢春晓[23] 为协同过滤算法引入了基于线性时间函数的权重和基于项目相似度的权重，旨在突出近期评分数据的重要性，同时也没有忽略较长时间前的评分数据。

目前的研究主要集中在两个方面：1）限定信息有效的时间窗，将该时间窗内的用户评分作为用户-项目评分矩阵的有效信息，再用传统的协同过滤方法为用户进行推荐；2）引入非线性遗忘曲线，认为信息随时间呈指数衰减，然后在预测评分时将信息衰减函数当作用户评分的时间加权函数。为用户做出推荐。前者完全否定了远期数据对推荐的影响力，而后者忽略了信息在较小的时间窗内影响力保持不变，而且没有将加权函数加入项目相似度计算中，忽略了信息过期对项目间相似度的影响。

3.2　改进的时间加权的协同过滤

心理学家对遗忘现象的研究结果表明，人类的遗忘过程是逐步的、非线性的借鉴非线性遗忘函数。本文提出改进的时间加权函数，将信息被遗忘的过程当作信息价值衰减的过程，即信息被遗忘的程度代表着信息对于目前推荐所具有的参考价值，被遗忘程度越高表示该信息越没有价值。时间加权函数赋予评分信息合理的时间权重，以凸显近期评分信息对目前推荐的重要性，从而找出与目标用户最相似的最近邻居集。

3.2.1　非线性遗忘曲线

在日常生活中，遗忘对于每个人来说都不陌生。德国心理学家艾宾浩斯成为首位探索遗忘规律的人，根据他的实验结果可以发现，人类忘记事物或者知识的进程是不均匀的，我们遗忘的速度刚开始比较快，随后逐渐变慢。他根据自己的实验结果整理出了用来描述人类记忆遗忘的整个过程的艾宾浩斯遗忘曲线，如图 2 所示。

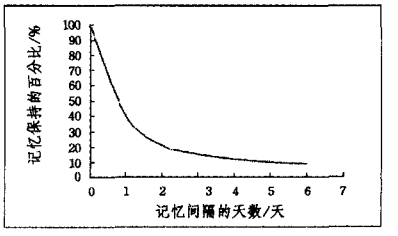

图 2　艾宾浩斯遗忘曲线

从图 2 可以看出，随着时间的不断推移，函数下降速率逐渐变慢，最后趋于稳定，这就是为什么我们总是能对一段时间之前的记忆有大概的印象，却想不起其中的细节。

3.2.2　改进的时间加权函数

前人[7] 利用非线性指数遗忘函数来刻画信息的衰减程度，即提出时间加权函数 $f(t)$，其值保持在(0,1]之间，用来体现不同时间的评分值对推荐的不同贡献程度。

为了描述信息从诞生到衰退以及最后消失的过程，引入信息半衰期[7] 的概念。

定义 1(信息的半衰期 T_0)　信息从发布到影响力下降到一半所需要的时间，即经过时间 T_0，信息的影响力减半。

于是，可以推出：

$$f(T_0)=(1/2)f(0) \tag{5}$$

由式(5)得出，经过时间 T_0 后，时间加权变为 1/2，即用户评分的可参考价值变为原来的一半。然后，定义衰减因子 λ：

$$\lambda=\frac{\ln 0.5}{T_0} \tag{6}$$

综上，得出时间加权函数 $f(t)$：

$$f(t)=e^{\lambda\cdot t} \tag{7}$$

其中，$t=t_{now}-t_{u,i}$，$t_{u,i}$ 表示 u 对项目 i 评分的时间，$f(t)$ 的值

代表时间加权值，即信息的衰减程度，其函数值保持在(0,1]之间，并且随着时间 $t_{u,i}$ 的增大而减小，表示用户近期的访问记录对预测有着更重要的价值。

虽然信息在总体上呈现非线性递减，但是在一定的时间内影响力不会发生显著变化。假设电影评分信息的信息保持期为 5 天，即一部电影一般在距离当前时间的 5 天里用户对它的评分信息的价值可以视为等同的，从第 6 天开始才进行衰减，并且第 6～10 天，信息的衰减程度是一致的。为此，本文提出信息保持期的概念，定义如下：

定义 2(信息的保持期 T') 信息的影响力保持不变的时间周期。

将信息保持期引入到时间加权函数中，提出改进的时间加权函数 $F(t)$：

$$F(t)=e^{\lambda\cdot T'\cdot\lfloor\frac{t}{T'}\rfloor} \tag{8}$$

同上，$\lambda=\ln 0.5/T_0$，$t=t_{now}-t_{u,i}$，$t_{u,i}$ 表示 u 对项目 i 评分的时间。

如图 3 所示，实线代表改进的时间加权函数，虚线代表原始的传统加权函数。这里取信息半衰期 $T_0=50$，保持期 $T'=5$，横轴表示距离评分的时间，纵轴表示随着时间的推移能够对评分赋予的时间权重。

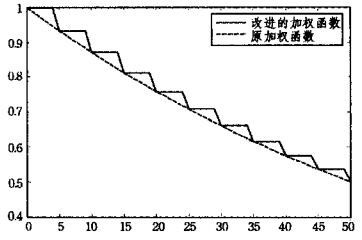

图 3 随时间的推移各函数对评分赋予的权重

对比图 3 中的实线和虚线可以看出，改进的时间加权函数加入信息保持期的概念之后，相当于在原来的加权函数上引入了一个信息基本保持不变的时间窗，使得信息的衰减呈现梯度指数衰减，更加符合实际情况。

3.2.3 算法描述

在传统的余弦相似性计算中加入改进的时间加权函数，为每个评分赋予时间权重，相对于目前的推荐来说更能代表项目间真正的相似程度，可以提高相似度度量的精准度。改进的相似度度量方法如下：

$$TSim(i,j)=\frac{\sum_{u\in U_{ij}}R_{u,i}\cdot F(\Delta t_{u,i})\cdot R_{u,j}\cdot F(\Delta t_{u,j})}{\sqrt{\sum_{u\in U_{ij}}(R_{u,i}\cdot F(\Delta t_{u,i}))^2}\times\sqrt{\sum_{u\in U_{ij}}(R_{u,j}\cdot F(\Delta t_{u,j}))^2}} \tag{9}$$

其中，$TSim(i,j)$ 表示项目 i 和 j 的相似度；$R_{u,i}$ 和 $R_{u,j}$ 分别表示用户 u 对项目 i 和 j 的评分；$F(\Delta t_{u,i})$ 为时间加权函数；$\Delta t_{u,i}=t_{now}-t_{u,i}$ 表示项目 i 被评分时间与目前时间的间隔时间。

新的预测评分公式为：

$$P_{u,i}=\bar{R}_i+\frac{\sum_{j\in NN_i}TSim(i,j)\times F(\Delta t_{u,j})\times(R_{u,j}-\bar{R}_j)}{\sum_{j\in NN_i}|TSim(i,j)|\cdot F(\Delta t_{u,j})} \tag{10}$$

其中，$P_{u,i}$ 表示 u 对 i 的预测评分，NN_i 代表 i 的最近邻集合，$\bar{R}_i$ 和 $\bar{R}_j$ 为项目 i 和 j 在整个用户空间上的平均评分。

在对相似性度量方法和预测评分方法进行改进后，整理出改进的时间加权的协同过滤算法 NTWCF 的具体算法步骤如下。

算法 1 NTWCF 算法

输入：带有评分时间戳的评分矩阵 R，目标用户 u，最近邻居数 k，信息半衰期 T_0，信息保持期 T'。

输出：向目标用户 u 推荐的 N 个项目。

Step1 将目标用户 u 尚未评分的项目作为目标项目集 I；

Step2 利用式(8)计算 R 中评分的时间加权值，形成时间加权矩阵；

Step3 形成邻居集，采用改进的式(9)计算目标项目 i∈I 和其他项目之间的相似度，并选取相似度最高的前 k 个项目形成目标项目 i 的最近邻集 NN_i；

Step4 计算预测评分值，采用式(10)预测目标用户 u 对目标项目 i 的评分 $P_{u,i}$；

Step5 对集合 I 中的项目重复 Step3 和 Step4，预测 I 中所有尚未评分项目在评分后，为 u 推荐预测值 $P_{u,i}$ 较高的前 N 个项目。

NTWCF 算法在提出信息保持期概念的基础上，将信息影响力保持不变的时间窗融合到信息指数衰减函数中，得出改进的时间加权函数，将其应用在项目相似度计算和预测评分两个阶段，并提出改进的相似度计算方法和预测评分方法，有效提高了算法的预测准确性。算法的时间复杂度为 $O(mn*n*n)$，其中 Step1 的时间复杂度为 $O(m)$；Step2 的时间复杂度为 $O(mn)$；Step3 和 Step4 是循环，循环次数为 n，一次循环时间复杂度为 $O(mn*n)$，所以时间复杂度为 $O(mn*n*n)$。

4 利用时间加权的项目聚类优化 NTWCF 算法

随着时代的发展，推荐系统中用户规模和项目规模急剧扩张，传统的基于项目的协同过滤要在整个项目集合上搜索查找目标项目的最近邻，不仅实时性达不到要求，而且效率极低。项目聚类是将那些看起来拥有相似评分的项目标识为一组，即一个聚类，每个聚类都有自己的聚类中心。当聚类形成之后，对目标项目的预测就只需要在相应的部分聚类上进行，减少了大量不必要的计算。

为了提高改进算法的实时响应速度，减少项目最近邻计算时的数据稀疏问题，本文在传统项目聚类的基础上考虑时间对信息的影响，提出时间加权的项目聚类并利用其优化 NTWCF 算法，然后总结出基于时间权重和项目聚类的协同过滤即 TWICCF 算法。

4.1 时间加权的项目聚类

传统聚类算法没有考虑时间因素对信息价值造成的影响，使得聚类结果不够准确，导致实际上相似的项目可能并未被聚到同一个类中。如 2.2 节所述，信息的影响力会随着时间发生衰减，本文提出如式(8)所示的改进的时间加权函数 $F(t)$ 并将其应用于基于项目的协同过滤算法中，得到 NTWCF 算法，该算法在预测准确性上表现出了相当明显的优势，说明 $F(t)$ 能够相对准确地表示信息随时间衰减的权值。因此，本文提出时间加权的项目聚类算法，这里采用 K-means

聚类算法对项目进行聚类。

鉴于对评分信息进行时间加权后更接近于其对当前推荐的实际影响力，时间加权的项目聚类算法首先利用 $F(t)$ 计算评分的时间加权值，使得加权后的评分值能更准确地表示它对于目前推荐的价值；然后根据这些评分信息衡量项目之间的距离，将真正相似的项目聚为一类。

该聚类算法主要分为以下 6 个步骤。

1)从 R 中得到所有 n 个项目的集合 I 和所有 m 个用户的集合 U。

2)设定半衰期 T_0 和保持期 T'，利用式(8)计算每个评分值 $R_{u,i}$ 的时间加权值 $F(\Delta t_{u,i})$，形成时间加权后的评分值 $R'_{u,i}$，如式(11)所示：

$$R'_{u,i}=R_{u,i} * F(\Delta t_{u,i}) \tag{11}$$

3)随机抽取 *ClusterNum* 个项目的加权后评分当作最初的聚类中心，记为集合：$CC=\{cc_1, cc_2, \cdots, cc_{ClusterNum}\}$，每个聚类中心 $cc_j \in CC$ 对应一个聚类，记为 c_j。

4)利用式(12)计算每个项目 $i \in I$ 与聚类中心 $cc_j \in CC$ 的相似性，将 i 放入与其最相似的聚类中心 cc_m 对应的聚类 c_m 中。

$$sim(i,cc_j)=\frac{\sum\limits_{u\in U_i} R'_{u,i}\times R_{u,\alpha_j}}{\sqrt{\sum\limits_{u\in U_i} R'^2_{u,i}}\times\sqrt{\sum\limits_{u\in U_i} R^2_{u,\alpha_j}}} \tag{12}$$

5)对于每个聚类，计算新的聚类中心，即一次迭代完成加入新的项目后更新聚类中心向量。

6)重复 4)和 5)直到聚类中心不再变化，达到收敛，得到 *ClusterNum* 个聚类和聚类中心。

由时间加权的聚类算法生成的 *ClusterNum* 个聚类中心 CC 和 *ClusterNum* 个聚类一一对应。表 2 所列的聚类中心矩阵中，m 行表示 m 个用户，*ClusterNum* 列表示 *ClusterNum* 个聚类，$R_{i,j}$ 代表 u_i 对 cc_j 中所有项目的平均评分。

表 2　聚类中心矩阵

	cc_1	…	cc_j	…	$cc_{ClusterNum}$
u_1	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,ClusterNum}$
u_2	…	…	…	…	…
u_i	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,ClusterNum}$
…	…	…	…	…	…
u_m	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,ClusterNum}$

4.2　基于时间权重和项目聚类的协同过滤

在协同过滤算法中，非常重要的一步就是为目标项目寻找邻居项目，当大量用户和项目加入时，在整个评分集中搜索会导致算法扩展性差。传统的协同过滤算法在全部的项目集合中查找目标项目的最近邻居所花费的时间导致了其在实时性上的失败，所以利用项目聚类为目标项目缩小最近邻候选集，能提高算法的执行效率。本文在 NTWCF 的基础上综合时间加权的项目聚类方法，提出了综合时间权重和项目聚类的协同过滤算法，即 TWICCF 算法。

4.2.1　TWICCF 算法流程

得到 3.1 节中的聚类结果之后，在形成目标项目的近邻集时，不基于整个项目集合，而基于与其较相似的若干聚类构成的项目集。TWICCF 算法采用时间加权的相似性度量方法衡量项目间的相似性，为目标用户构建最近邻居集，最后采用时间加权的预测评分方法产生预测评分并得到推荐结果，其算法流程如图 4 所示。

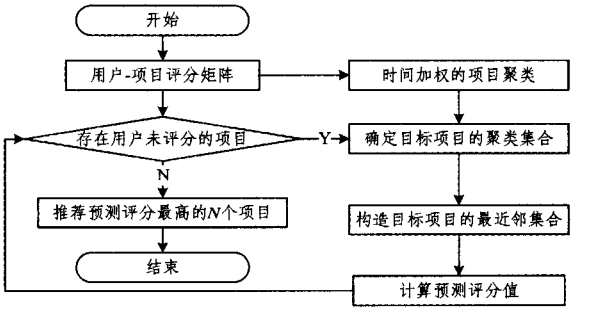

图 4　TWICCF 算法流程

4.2.2　TWICCF 算法描述

TWICCF 算法将时间因素对项目评分参考价值的影响考虑在内，在项目聚类、目标项目最近邻计算阶段和预测评分阶段对项目评分进行时间加权。

1)根据 4.1 节所述离线进行时间加权的项目聚类，首先对评分信息时间加权，然后根据项目在用户评分层面上的相似程度将项目分配到不同的组中，每个组中的项目被具有相同兴趣的用户所喜欢。所以，在为这些用户推荐产品时，只需在该用户感兴趣的聚类中进行搜索，就可以得到目标用户未评分项目的绝大多数邻居项目，从而避免不必要的计算，大力保证算法的实时性。

2)根据项目和各个聚类中心的相似度，为每个待预测的项目寻找搜索最近邻居的项目空间，通常为一个或者几个聚类，然后在这个新的项目空间上执行 NTWCF 算法。本文将这一过程称为狭义上的 TWICCF 算法。

算法 2　TWICCF 算法

输入：带有时间戳的评分矩阵 R，目标用户 u，最近邻居数 k，项目聚类集合 C，聚类中心集合 CC，阈值 rate，信息半衰期 T_0 和信息保持期 T'。

输出：向 u 推荐的 N 个项目。

Step1　将 T' 和 T_0 代入式(8)，使每个项目评分值得到时间加权值；

Step2　利用式(13)计算项目 i 和聚类中心 cc_j 的相似度，记录相似度大于阈值 ε 的聚类，将这些聚类中的项目作为目标项目最近邻集合的搜索空间：

$$sim(i,cc_j)=\frac{\sum\limits_{u\in U_i} R_{u,i}\times R_{u,cc_j}\times F(\Delta t_{u,i})}{\sqrt{\sum\limits_{u\in U_i}(R_{u,i}\times F(\Delta t_{u,i}))^2}\times\sqrt{\sum\limits_{u\in U_i} R^2_{u,cc_j}}} \tag{13}$$

Step3　利用相似性度量方法即式(9)计算最近邻搜索空间中的项目与目标项目 i 的相似性 Tsim(i,j)，选取相似度较高的 k 个项目构造 i 的最近邻集 NN_i；

Step4　根据 u 对 NN_i 中项目的评分及相应的时间加权值，采用改进的预测评分方法式(10)预测 u 对 i 的评分 $P_{u,i}$；

Step5　预测 I 中所有项目的评分之后，为 u 推荐 $P_{u,i}$ 较高的 N 个项目。

5　实验结果与分析

5.1　数据集

MovieLens 数据集主要包括如下信息：用户 ID 信息、产品 ID 信息、用户对产品的评分信息 Rating 及评分时间戳。

该站点共有 45000 个用户对 6600 部不同的电影发表过看法。用户的评分值为 1～5 分，1 和 2 分表示负面评价，4 和 5 分表示正面评价，3 分居中。本文采用的数据集为来自 943 个用户在 1682 部电影上的 100000 条评分。

本文从 MovieLens 中随机选取 5 组数据，其中每组包含 400 位随机用户对所有项目的评分信息，将每组数据集中每个用户的最近评分数据用作测试，其余评分数据用作训练。如此划分训练集和测试集，按照时间顺序并根据训练集来预测测试集中的评分，能够最大程度地还原现实生活的推荐。本文利用 5 组随机选取的数据集，采用交叉验证的方法对算法进行验证。

5.2 算法评估标准

任何算法都需要一定的评估办法来评定其性能，以便能够在应用于实践之前预先证明算法的可行性和合理性。本文用准确性和实时响应速度两个指标来对算法进行评估。

1)准确性。在推荐算法中，一般通过衡量预测评分的准确程度来体现推荐算法的质量。本文采用平均绝对偏差 MAE 作为判断算法准确性的标准。对于在测试集上的每个预测评分-真实评分对$\langle p_i, q_i \rangle$，统计它们的绝对误差总和，然后再计算平均值。

$$MAE=\frac{\sum_{i=1}^{n}|p_i-q_i|}{n} \tag{14}$$

其中，n 表示测试集中的项目总数，MAE 的值越高，证明预测评分越偏离实际评分值，预测越不准确。平均绝对误差主要是通过计算推荐算法对项目的预测评分和用户对项目的真实评分之间的偏差，用测试集中所有项目的平均偏差来衡量算法的整体预测准确性。

2)实时响应速度。本文利用相同参数条件下算法的执行时间来衡量算法的实时响应速度。

5.3 实验方案及结果分析

5.3.1 实验方案

本文将实验主要分为两个部分：

1)分析算法中参数对算法准确性的影响，包括聚类数目、目标项目和聚类中心的相似性阈值、信息半衰期、信息保持期、最近邻居数等参数。

2)进行对比性实验，主要是通过比较相同参数条件下本文提出的算法比已经存在的算法在准确性和响应速度两个方面具有的优越性。

5.3.2 参数分析实验结果

1)分析信息半衰期 T_0 对 NTWCF 算法的影响。

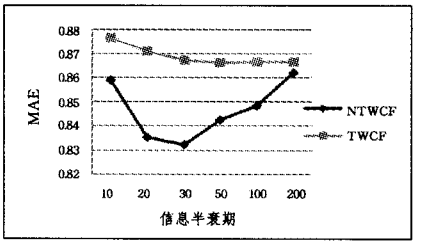

图 5 信息半衰期对算法 MAE 的影响

取信息保持期 $T'=3$，近邻数 $k=30$ 时，观察 NTWCF 算法在不同半衰期下的 MAE 值，并与没有引入时间保持期的时间加权的协同过滤算法(称为 TWCF 算法)进行对比。由图 5 可以得出，在信息保持期 $T'=3$ 的情况下，NTWCF 算法在信息半衰期为 30 时 MAE 最小，推荐准确度最高。与 TWCF 算法相比，在半衰期和最近邻数目一致的情况下，本文提出的 NTWCF 算法表现更优，平均绝对误差更小，推荐的准确性更高。

2)分析信息保持期 T' 对 NTWCF 算法的影响。分别取信息半衰期 T_0 为 20 天、30 天和 50 天，近邻数 $k=30$ 时，观察 NTWCF 算法的 MAE 随信息保持期 T' 的变化，如图 6 所示。

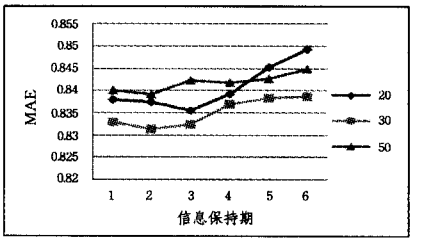

图 6 NTWCF 算法的 MAE 随 T' 的变化

由图 6 可知，信息半衰期 T_0 越小，算法对信息保持期 T' 的改变表现得越敏感，即当信息保持期发生变化时，算法的 MAE 会随之产生较大的变化；信息保持期 T' 会影响 NTWCF 算法的推荐效果，且不同的半衰期对应的信息保持期的最优值也有所不同。但从总体上来看，T' 为 2～3 天时算法呈现出最好的准确性。

3)分析聚类数 m 及聚类中心和目标项目的相似性阈值 $rate$ 对 TWICCF 算法的影响。根据对 T_0 和 T' 的分析，选取信息半衰期 $T_0=30$，信息保持期 $T'=2$，且 $rate$ 的值分别取 0.2，0.3 和 0.4 时，观察并分析聚类数 m 取不同值时对应的 TWICCF 算法的 MAE 变化，如图 7 所示。

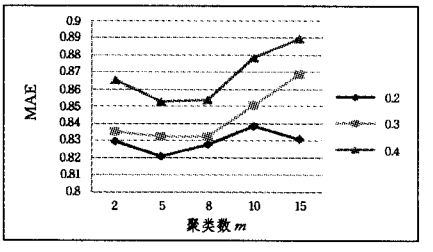

图 7 聚类数对算法的 MAE 的影响

由图 7 可知，不论 $rate$ 的值取多大，当聚类数取 5～8 时，算法的 MAE 都较低，预测的准确性都较高。当聚类数太大时，每个聚类中的项目数量较少，则在形成最近邻候选集时会有一部分真正的邻居项目被遗漏掉，从而造成推荐结果不准确。当聚类数目过小时，MAE 反而会升高，这是由于与聚类中心的相似度能够达到 $rate$ 的项目可能会变少，导致一些项目得不到准确的预测评分；此外，聚类数过小时最近邻候选集变大，一些实际上为非近邻的项目也可能会干扰实验，反而影响了推荐的结果。

5.3.3 算法对比实验结果

5.3.2 节中的分析基本给出了参数对算法的影响。在接下来的对比性实验中，本文提出的算法取信息保持期 $T'=2$，

信息半衰期 $T_0=30$，聚类数目 $m=5$，聚类中心和目标项目的相似性阈值 *rate* 为 0.3，以上参数为最近邻 k 取 30 时的最优参数，但近邻数取其他值时不一定为最优参数。

1)采用如上所述参数，分别对基于项目的协同过滤 IBCF 算法、TWCF 算法和本文提出的 NTWCF 算法在不同近邻数情况下的 MAE 进行比较，如图 8 所示。

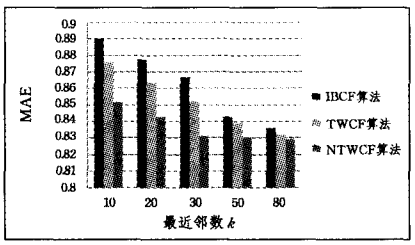

图 8　IBCF，TWCF 和 NTWCF 的 MAE 比较

实验结果表明，当最近邻居数增加时，IBCF 算法、TWCF 算法和 NTWCF 算法的 MAE 均呈递减趋势，所以合理地选取最近邻也是协同过滤算法取得成功的关键因素。在近邻数相同的情况下，本文提出的 NTWCF 算法的 MAE 取值最小，说明 NTWCF 算法在准确性上优于 TWCF 算法和 IBCF 算法，取得了最好的预测推荐准确度。

2)对传统基于项目聚类的协同过滤 ICCF 算法、传统聚类上的 NTWCF 算法及 TWICCF 算法的准确性进行比较。本文提出的 TWICCF 算法为利用时间加权聚类优化的 NTWCF 算法，所以本文将利用传统聚类结果的 NTWCF 算法称为 OTWICCF 算法，即在项目聚类阶段采用传统聚类方法。

如图 9 所示，在聚类数和近邻数相同的情况下，本文提出的 TWICCF 算法在准确性上明显优于其他两种算法；TWICCF 算法优于 OTWICCF 算法，说明本文提出的基于时间加权的项目聚类方法能够有效提高聚类结果的准确程度，从而使算法既精确又高效。总体来看，当最近邻居数较大时，ICCF 算法、OTWICCF 算法以及 TWICCF 算法的预测效果均更优。

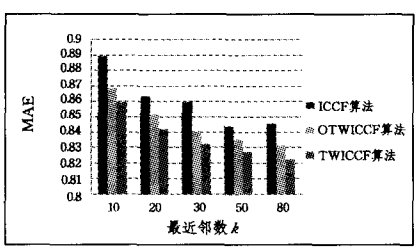

图 9　ICCF，OTWICCF 和 TWICCF 的 MAE 比较

3)比较相同近邻数及参数条件下，NTWCF 算法和 TWICCF 算法的 MAE 值。

利用聚类技术优化 NTWCF 算法提出的 TWICCF 算法主要是为了提高算法的执行效率，在不牺牲准确性的前提下，尽量减少不必要的计算，缩短算法的执行时间，有力保证实时性。

如图 10 所示，当参数相同时，NTWCF 算法和 TWICCF 算法的 MAE 基本没有变化，并且在合适的聚类下 TWICCF 算法的表现反而优于 NTWCF 算法。

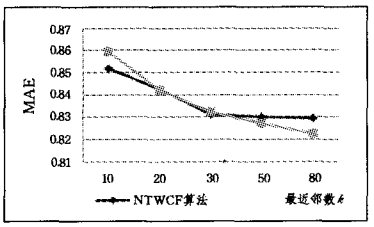

图 10　NTWCF 和 TWICCF 的 MAE 比较

4)对 TWICCF 算法和 NTWCF 算法的实时性进行比较。在推荐系统中，算法的执行效率是相当关键的，一个有效的算法可能因为响应时间过长而被丢弃。

由图 11 可知，TWICCF 算法的执行时间明显短于 NTWCF 算法的执行时间，虽然在 TWICCF 算法中项目聚类需要一定的时间，但是此过程可以离线完成。

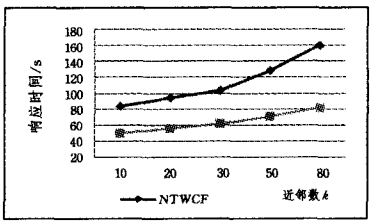

图 11　NTWCF 和 TWICCF 的实时性比较

综上所述，本文提出的 NTWCF 算法的预测评分准确性大幅度提升，且实验证明了在相同的参数条件下，TWICCF 算法的 MAE 不仅远远小于同类算法，也说明了信息随时间推移可参考性衰减理论的正确性。除此之外，利用时间加权的聚类结果优化的 TWICCF 算法不仅实时响应速度更快，有效地提高了推荐算法的实时性，而且其准确性也并未受到干扰，在合适的聚类下，预测结果反而更精准。

结束语　为了解决信息过期的问题，本文对考虑时间因素的协同过滤算法进行了研究，引入信息保持期的概念并对时间加权函数进行了优化，从而提出了一种改进的时间加权的协同过滤算法 NTWCF。利用电影数据集进行实验，结果表明，该算法显著地提高了预测的准确性，达到了更好的推荐效果。

针对推荐系统对实时性的要求，本文利用时间加权的项目聚类优化 NTWCF 算法，提出了 TWICCF 算法。实验结果表明，TWICCF 算法在推荐效果上不仅不差于 NTWCF 算法而且大幅度优于同类比较算法，在响应时间上大大优于 NTWCF 算法，从而在准确性和响应效率上达到了全面优化。

参 考 文 献

[1] GOLDBERG D, NICHOLS D, OKI B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.

[2] RESNICK P, IACOVOU N, SUCHAK M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]// Proceedings of the 1994 ACM conference on Computer supported cooperative work. New York: ACM, 1994: 175-186.

（下转第 322 页）

tion,1990,2(3):293-307.

[16] JOHNSON J L,PADGETT M L. PCNN models and applications[J]. IEEE Transactions on Neural Netmorks,1999,10(3):480-498.

[17] SHEN Y,ZHANG X M,HAN K G,et al. Research of image segmentation technology based on PCNN [J]. Modern Electronics Technique,2014,37(2):38-41. (in Chinese)
沈艳,张晓明,韩凯歌,等. PCNN 图像分割技术研究[J]. 现代电子技术,2014,37(2):38-41.

[18] AN Q,LI M,HE Y J,et al. Novel PCNN Model and its Application on Image Segmentation[J]. Computer Science,2014,41(6A):215-217. (in Chinese)
安琦,李敏,何玉杰,等. 一种优化脉冲耦合神经网络模型及在图像分割中的应用[J]. 计算机科学,2014,41(6A):215-217.

[19] JIN W B,SHEN J J,ZHANG Z F,et al. Approach to image segmentation with spatial moments based on PCNN [J]. Application Research of Computers,2009,26(12):4800-4802. (in Chinese)
金文标,沈晶晶,张智丰. 一种基于空间矩的 PCNN 图像分割方法[J]. 计算机应用研究,2009,26(12):4800-4802.

(上接第 301 页)

[3] GREG L,BRENT S,JEREMY Y. Amazon. com recommendations: Item-to-item collaborative filtering [J]. IEEE Internet Computing,2003,7(1):76-80.

[4] SU X,KHOSHGOFTAAR T M. A survey of collaborative filtering techniques[J]. Advances in Artificial Intelligence,2009,2009:4.

[5] WEI S Y,YE N,ZHANG S,et al. Collaborative filtering recommendation algorithm based on item clustering and global similarity[C]//2012 Fifth International Conference on Business Intelligence and Financial Engineering (BIFE). IEEE,2012:69-72.

[6] OU L Q,CHEN L,MA Y. Study on New Item Recommendation Method in Collaborative Filtering Algorithm[J]. Microcomputer Information,2005,21(11X):186-187. (in Chinese)
欧立奇,陈莉,马煜. 协同过滤算法中新项目推荐方法的研究[J]. 微计算机信息,2005,21(11X):186-187.

[7] DING Y,LI X. Time weight collaborative filtering[C]// Proceedings of the 14th ACM International Conference on Information and Knowledge Management. ACM,2005:485-492.

[8] POLAT H,DU W. SVD-based collaborative filtering with privacy[C]//Proceedings of the 2005 ACM Symposium on Applied Computing. ACM,2005:791-795.

[9] MELVILLE P,MOONEY R J,NAGARAJAN R. Content-boosted collaborative filtering for improved recommendations[C]// AAAI/IAAI. Edmonton,Canda,2002:187-192.

[10] UNGAR L H,FOSTER D P. Clustering methods for collaborative filtering[C]// AAAI Workshop on Recommendation Systems. 2000.

[11] GONG S J. A collaborative filtering recommendation algorithm based on user clustering and item clustering[J]. Journal of Software,2010,5(7):745-752.

[12] WANG X,YU Z,WANG C. Recommendation with Item Clustering Based Collaborative Filtering[C]//2014 International Conference on Computer Science and Electronic Technology (ICCSET 2014). Atlantis Press,2015.

[13] WEI S Y,YE N,ZHU J,et al. Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity[J]. Computer Science,2012,39(12):149-152. (in Chinese)
韦素云,业宁,朱健,等. 基于项目聚类的全局最近邻的协同过滤算法[J]. 计算机科学,2012,39(12):149-152.

[14] LIU D H,PENG D W,ZHANG H. Collaborative Filtering Algorithm Based on Time Weight and User's Feature[J]. Journal of Wuhan University of Technology,2012,34(5):144-148. (in Chinese)
刘东辉,彭德巍,张晖. 一种基于时间加权和用户特征的协同过滤算法[J]. 武汉理工大学学报,2012,34(5):144-148.

[15] DENG J,CHEN X Q. Study of Collaborative Filtering Recommendation Algorithm Based on Users' Interest Change[J]. Journal of Wuhan Polytechnic University,2013,32(4):48-51. (in Chinese)
邓娟,陈西曲. 基于用户兴趣变化的协同过滤推荐算法[J]. 武汉工业学院学报,2013,32(4):48-51.

[16] XING C X,GAO F R,ZHAN S N,et al. Collaborative filtering recommendation algorithm incorporated with user interest change[J]. Computer Research and Development,2007,44(2):296-301.

[17] ZENG D H,WANG T,YAN S F,et al. One Collaborative Filtering Recommendation Algorithm Based on Exponential Forgetting Function[J]. Science Mosaic,2013(7):10-15. (in Chinese)
曾东红,汪涛,严水发,等. 一种基于指数遗忘函数的协同过滤算法[J]. 科技广场,2013(7):10-15.

[18] SARWAR B,KARYPIS G,KONSTAN J,et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. ACM,2001:285-295.

[19] ZHENG C C,LI L. Research on method of similarity measurement in collaborative filter algorithm[J]. Computer Engineering and Applications,2014,50(8):147-149. (in Chinese)
郑翠翠,李林. 协同过滤算法中的相似性度量方法研究[J]. 计算机工程与应用,2014,50(8):147-149.

[20] GRABTREE I,SOLTYSIAK S. Identifying and Tracking Changing Interests[J]. International Journal of Digital Libraries,1998,2(1):38-53

[21] KOYCHEV I,SCHWAB I. Adaptation to Drifting User's Interests [C]// Proceedings of ECML 2000 Worksbop: Machine Learning in New Information Age. Barcelona,Spain,2000:39-46.

[22] KOYCHEV I. Gradual Forgetting for Adaptation to Concept Drift[C]//Proceedings of ECAI 2000. 2000.

[23] XING C X,GAO F R,ZHAN X S,et al. A Collaborative Filtering Recommendation Algorithm Incorporated with User Interest Change[J]. Journal of Computer Research and Development,2007,44(2):296-301. (in Chinese)
邢春晓,高凤荣,战思南,等. 适应用户兴趣变化的协同过滤推荐算法[J]. 计算机研究与发展,2007,44(2):296-301.