

# 融合项目聚类和时间权重的动态协同过滤算法

吴佳炜<sup>1</sup>, 沈玲玲<sup>1,2</sup>, 钱 钢<sup>1</sup>

(1.南京师范大学计算机科学与技术学院,江苏 南京 210097)

(2.南京师范大学商学院,江苏 南京 210097)

**[摘要]** 传统基于项目的协同过滤算法离线计算项目间的相似度,提高了向用户推荐的速度,但极大的数据稀疏度影响了推荐质量,且该算法也忽略了用户兴趣随时间变化这一现象. 针对上述问题,提出了一种融合项目聚类和时间权重的动态协同过滤算法,根据用户偏好对项目进行聚类,找出类别偏好相似的候选邻居,再在候选邻居中搜寻最近邻,排除与目标项目共同评分较少的项目干扰,提高了搜寻相似项目的准确性. 同时,引入时间权重来反映用户兴趣随时间的变化,从整体上提高推荐质量. 在 MovieLens 数据集上进行实验,实验结果表明,本文所提出算法的推荐质量较传统的协同过滤算法有显著提高.

**[关键词]** 协同过滤,推荐系统,类别偏好,相似度,时间权重

**[中图分类号]** P208 **[文献标志码]** A **[文章编号]** 1672-1292(2017)03-0063-07

## Dynamic Collaborative Filtering Algorithm Fusing Item Clustering and Time Weight

Wu Jiawei<sup>1</sup>, Shen Lingling<sup>1,2</sup>, Qian Gang<sup>1</sup>

(1.School of Computer Science and Technology, Nanjing Normal University, Nanjing 210097, China)

(2.School of Business, Nanjing Normal University, Nanjing 210097, China)

**Abstract:** The traditional item-based collaborative filtering algorithm calculates item-item similarity offline and improves the real-time performance of recommender system, but the big data sparsity problem still impacts the quality of the algorithm and it also ignores the phenomenon that users' interests change over time. To address the issues above, this paper proposes a dynamic collaborative filtering algorithm fusing item clustering and time weight. The proposed algorithm first clusters items according to the user's preference, then finds out candidate neighbors who are similar to the target item in class preference. Then it searches for nearest neighbors in the candidate neighbor set, which eliminates the interference of the items those have few co-ratings with the target item. At the same time, this algorithm introduces time weight to reflect the change of users' interests over time, which improves recommendation quality from the overall. Experimental results based on MovieLens dataset show that the recommendation quality of the new algorithm is significantly improved compared with traditional item-based collaborative filtering algorithm and user-based collaborative filtering algorithm.

**Key words:** collaborative filtering, recommender system, class preference, similarity, time weight

随着互联网的快速发展,大数据时代应运而生,推荐系统(recommender system)的出现帮助用户解决了信息超载带来的困扰. 协同过滤(collaborative filtering)<sup>[1]</sup>是推荐系统中使用最成功的技术,其根本思想是对比类似用户来展现推荐用户的偏好,从而向其推荐最有可能感兴趣的资源.

尽管协同过滤技术在个性化推荐系统中有其显著的优越性,但随着系统中用户数目和项目数目的几何数级的增加,该技术面临着较为严峻的挑战<sup>[2]</sup>. 首先是缺乏可扩展性,即数据的庞大规模导致计算复杂度大大增加,无法及时向用户产生推荐. 其次就是数据稀疏性问题,系统中的项目资源数量巨大,而用户往往只对一小部分项目进行评分,因而可利用的评分信息非常有限. 此外,传统的基于项目的协同过滤推

收稿日期:2017-02-23.

基金项目:国家自然科学基金(61503188)、江苏省自然科学基金(BK20150982).

通讯联系人:沈玲玲,博士研究生,讲师,研究方向:管理科学与工程. E-mail:llshen509@163.com

荐算法<sup>[3-4]</sup>不能及时反映用户的兴趣随时间变化的现象,用户往往对近段时间的项目比较感兴趣,而对过去项目兴趣衰减,这一问题也影响到了推荐系统的推荐质量。

针对缺乏可扩展性的问题,研究人员提出了很多解决方案,包括数据降维模型,如奇异值分解(singular value decomposition, SVD)<sup>[5]</sup>等;基于聚类技术的模型,如基于用户聚类(user clustering)的方法<sup>[6]</sup>、联合聚类(co-clustering)方法<sup>[7]</sup>等;基于项目的协同过滤模型(item-based collaborative filtering)<sup>[8]</sup>。针对用户兴趣随时间变化的问题,文献[9]提出了时效量化这一概念,根据随时间发生梯度逐渐下降过程,指出了适合时效量化的指数函数,从而提高了推荐质量。

然而,基于项目的协同过滤算法仍然不能解决数据稀疏<sup>[10]</sup>所带来的问题。对此,本文提出了一种新的基于项目聚类的动态协同过滤算法,通过用户的评分对项目进行偏好聚类,排除与目标项目共同评分较少的项目的干扰,同时引入了时间权重来反映用户对项目的兴趣变化,整体上提高了算法的推荐质量。

## 1 相关研究

### 1.1 基于项目的协同过滤算法

传统的协同过滤推荐算法用一个  $m \times n$  用户—项目评分矩阵  $R$  来表示用户对项目的评分,其中  $m$  代表用户数量, $n$  代表项目数量, $r_{u,i}$  表示用户  $u$  对项目  $i$  的评分值。在计算相似度时,本文采用 Pearson 相关相似性<sup>[11]</sup>的计算方法。

基于项目的协同过滤算法使用 PCC(IPCC)<sup>[12]</sup>定义项目  $i$  和  $j$  的相似度为

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}, \quad (1)$$

式中,  $U = U_i \cap U_j$ , 表示同时调用过项目  $i$  和项目  $j$  的用户集;  $\bar{r}_i$  和  $\bar{r}_j$  表示项目  $i$  和项目  $j$  被所有用户评分后的平均评分;  $\text{sim}(i, j)$  表示两个项目间的相似度,取值范围在  $(-1, 1)$  之间,值越大代表项目  $i$  和  $j$  越相似,当  $U = \text{null}$  时  $\text{sim}(i, j) = \text{null}$ 。

基于用户的协同过滤算法使用 PCC(UPCC)<sup>[13]</sup>定义用户  $u$  和  $v$  的相似度为

$$\text{sim}(u, v) = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}, \quad (2)$$

式中,  $I = I_u \cap I_v$ , 表示同时被用户  $u$  和  $v$  调用过的服务项目;  $r_{u,i}$  表示用户  $u$  对服务项目  $i$  的评分;  $\bar{r}_u$  和  $\bar{r}_v$  表示用户  $u$  和用户  $v$  对所有不同项目的平均评分;  $\text{sim}(u, v)$  表示两个用户间的相似度,取值范围在  $(-1, 1)$  之间,值越大代表用户  $u$  和  $v$  行为越相似,当  $I = \text{null}$  时  $\text{sim}(u, v) = \text{null}$ 。

### 1.2 局限性

基于项目的协同过滤算法通过项目间的共同评分来计算项目相似度,当项目间获得的共同评分较多时,计算出的项目相似度的准确率较高。随着互联网的快速发展,项目资源以几何数级增长,而用户对项目的评分数据却相当稀疏,项目间获得的共同评分较少,从而大大影响了项目间相似度计算的准确率。基于项目的协同过滤算法的核心思想是在整个项目空间内搜寻最近邻,很多项目与目标项目共同评分较少,或者原本与目标项目差别就很大,但因为计算出较高的评分相似性而被选为最近邻,降低了搜寻最近邻的准确性。此外,基于项目的协同过滤算法忽略了用户兴趣随时间的变化,用户一般对最近的项目有较高的兴趣,而对过去项目的兴趣度较小,这也会影响到算法的推荐质量。

## 2 算法思想

### 2.1 用户聚类

利用式(2)可以对用户进行聚类,将相似度较高的用户划分到同一类中,不同类中的用户兴趣差别较大,整个聚类算法的步骤如下:

输入数据源:用户—项目评分矩阵  $R$ 。

输出:  $l$  个聚类.

算法步骤:

从项目集合  $I = \{Item_1, Item_2, \dots, Item_n\}$  中检索所有  $n$  个项目;

从用户集合  $U = \{User_1, User_2, \dots, User_m\}$  中检索所有  $m$  个用户;

从  $m$  个用户中选出评分量最多的  $l$  个用户作为初始聚类中心,记为  $\{W_1, W_2, \dots, W_l\}$ ;

根据聚类中心初始化  $l$  个聚类  $C_1, C_2, \dots, C_l$ , 记为  $C = \{C_1, C_2, \dots, C_l\}$ ;

repeat

for 每个用户  $u_i \in U$

for 每个聚类中心  $W_l \in \{W_1, W_2, \dots, W_l\}$

根据式(2)计算  $u_i$  与  $W_l$  的相似性  $\text{sim}(u_i, W_l)$

end for

$\text{sim}(u_i, W_m) = \max \{ \text{sim}(u_i, W_1), \text{sim}(u_i, W_2), \dots, \text{sim}(u_i, W_l) \}$

聚类  $C_m = C_m + u_i$

end for

until 聚类的成员不再变化

## 2.2 偏好项目

设系统中的用户集合为  $U = \{u_1, u_2, \dots, u_n\}$ . 根据算法将用户聚为  $l$  类, 即  $C = \{c_1, c_2, \dots, c_l\}$ , 每个类中包含的用户尽可能相似, 类与类之间的用户尽可能不同, 有如下性质:  $U = c_1 \cup c_2 \cup \dots \cup c_l, c_i \cap c_j = \emptyset (1 \leq i \leq l, 1 \leq j \leq l)$ . 虽然系统中用户数量庞大, 但一个项目通常只会获得一小部分用户的评分, 且这些评分往往集中在一个或少数几个用户类中, 因此项目在这些用户类上获得的评分相对稠密. 项目所获评分在用户类上的分布反映了项目在各用户类中受到的偏好程度. 由表 1 可见, 项目  $i_1$  的评分集中在类  $c_2$  中, 表明  $i_1$  更多受到类  $c_2$  的偏好; 项目  $i_2$  的评分集中在  $c_1$  和  $c_2$  中, 项目  $i_3$  的评分集中在  $c_3$  中, 相对于  $i_3, i_1$  与  $i_2$  受到的用户类别偏好更加相似.

表 1 不同类的评分数据

Table 1 Rates of different classes

项目	$c_1$					$c_2$					$c_3$				
	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$
$i_1$	3						5	3	2	4			2		
$i_2$	2	4	3	1		4	3	5	3	5				1	
$i_3$					4						5	2		2	1

## 2.3 确定候选邻居

**定义 1**(类别偏好值) 设  $U$  表示整个用户空间,  $C$  表示经过聚类生成的用户类集合,  $i$  为项目空间中任一项目,  $U_i = \{u \in U | r_{u,i} \neq \emptyset\}$  表示对项目  $i$  进行过评分的用户集合,  $\forall c_s \in C$ , 则称  $P_{i,s} = |c_s \cap U_i| / |U_i|$  ( $0 \leq P_{i,s} \leq 1$ ) 为项目  $i$  在  $c_s$  上的类别偏好值.

根据类别偏好值构造出项目类别偏好矩阵  $P(n, l)$ , 如表 2 所示,  $n$  行表示  $n$  个项目,  $l$  列表示  $l$  个用户类,  $P_{i,s}$  表示项目  $i$  在用户类  $s$  上的偏好值. 在  $P(n, l)$  上计算目标项目  $j$  与其他项目的类别偏好相似性, 按照类别偏好相似性从高到低的顺序排列所有项目, 组成  $j$  的候选邻居列表  $T_{cn}$ ; 然后计算每一项目与  $T_{cn}$  中其所在行前  $k'$  个项目评分的相似性, 生成项目最近邻列表  $T_{nn}$ , 在  $T_{nn}$  中找到  $j$  所对应的行, 顺序取出前  $k$  个用户  $u$  已评分的项目, 最后计算  $r_j(u, j)$ .

表 2 项目—类别偏好矩阵  $P(n, l)$

Table 2 Item-category preference matrix  $P(n, l)$

	$c_1$	$\dots$	$c_s$	$\dots$	$c_l$
Item <sub>1</sub>	$P_{1,1}$	$\dots$	$P_{1,s}$	$\dots$	$P_{1,l}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Item <sub>i</sub>	$P_{i,1}$	$\dots$	$P_{i,s}$	$\dots$	$P_{i,l}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Item <sub>n</sub>	$P_{n,1}$	$\dots$	$P_{n,s}$	$\dots$	$P_{n,l}$

## 2.4 基于时间权重的项目相似度

### 2.4.1 基于时间的用户兴趣加权函数

设用户  $u$  已访问过的项目集为  $I_u$ , 通过定义一个时间窗口  $T$  来获取用户  $u$  在近期  $T$  时间段内访问过

的项目集  $I_{uT}$ , 则  $I_{uT}$  在某种程度上反映了用户  $u$  在近一段时间的兴趣. 对于项目  $i \in I_u$  来说, 无论用户  $u$  访问项目  $i$  的时间早晚, 若  $I_{uT}$  中存在较多项目与项目  $i$  的相似度较高, 说明项目  $i$  和用户  $u$  近期的兴趣有一定的关联度, 则在未来的一段时间内, 用户  $u$  感兴趣的项目很可能依然和项目  $i$  相似, 因此定义基于时间的用户兴趣加权函数为:

$$f(u, i) = \frac{\sum_{j \in I_{uT}} \text{sim}(i, j)}{|I_{uT}|}, \quad (3)$$

式中,  $|I_{uT}|$  表示  $I_{uT}$  项目的个数.

通过改变时间窗口  $T$  的长短, 可得到不同的近段时间用户  $u$  访问的项目集合  $I_{uT}$ , 从而影响推荐质量.

#### 2.4.2 基于时间权重的项目相似度计算

在计算项目相似性时考虑到用户兴趣度随时间变化的影响, 将时间权重结合到项目相似度的计算公式中, 因此定义项目  $i$  和项目  $j$  间的相似度为

$$\text{sim}(i, j)_{\text{time}} = \frac{\sum_{u \in U} (f(u, i) \times (r_{u,i} - \bar{r}_i)) (f(u, j) \times (r_{u,j} - \bar{r}_j))}{\sqrt{\sum_{u \in U} (f(u, i) \times (r_{u,i} - \bar{r}_i))^2} \sqrt{\sum_{u \in U} (f(u, j) \times (r_{u,j} - \bar{r}_j))^2}}, \quad (4)$$

式中,  $U = U_i \cap U_j$  表示同时调用过项目  $i$  和项目  $j$  的用户集;  $\bar{r}_i$  和  $\bar{r}_j$  表示项目  $i$  和项目  $j$  被所有用户评分后的平均评分.

#### 2.5 未评分值预测

基于项目的协同过滤算法依靠相似项目来预测用户  $u$  对项目  $i$  的评分:

$$r_i(u, i) = \bar{r}(i) + \frac{\sum_{j \in T_{nn}} \text{sim}_{\text{time}}(i, j) (r(u, j) - \bar{r}(j))}{\sum_{j \in T_{nn}} \text{sim}_{\text{time}}(i, j)}, \quad (5)$$

式中,  $\bar{r}(i)$  表示项目  $i$  的平均评分;  $\bar{r}(j)$  表示项目  $j$  的平均评分.

#### 2.6 算法步骤

算法 基于项目聚类的动态协同过滤算法 (Proposed CF)

输入: 用户—项目评分矩阵  $\mathbf{R}(m, n)$ , 目标用户  $u$ , 待评分项目  $i$ , 用户类别数  $l$ , 候选邻居数  $k'$ , 最近邻数  $k$ .

输出:  $u$  对  $i$  的预测评分.

过程如下:

- (1) 把用户集  $U$  划分为  $l$  类, 得到用户类集合  $C = \{c_1, c_2, \dots, c_l\}$ ;
- (2) 在  $\mathbf{R}(m, n)$  上计算出项目  $j$  已评分的用户集合  $U_j = \{u \in U | R_{u,j} \neq \phi\}$ ;
- (3)  $\forall c_s \in C$ , 计算项目  $j$  在  $c_s$  上的类别偏好值  $P_{j,s} = |c_s \cap U_j| / |U_j|$ ;
- (4) 循环执行(2)、(3), 得到每个项目在各用户类上的偏好值, 构造项目—类别偏好矩阵  $\mathbf{P}(n, l)$ ;
- (5) 在  $\mathbf{P}(n, l)$  上计算项目间类别偏好相似性, 生成项目候选邻居列表  $T_{cn}$  并保存;
- (6) 在  $\mathbf{R}(m, n)$  上计算每一项目与  $T_{cn}$  中前  $k'$  个候选邻居的评分相似性, 生成项目最近邻列表  $T_{nn}$  并保存;
- (7) 在  $T_{nn}$  中找到  $i$  所对应的行, 顺序取出前  $k$  个  $u$  已评分的项目;
- (8) 根据式(4)计算  $u$  对  $i$  的预测评分  $r_i(u, i)$ .

### 3 实验和分析

#### 3.1 实验数据集

文中实验采用的是 MovieLens<sup>[14]</sup> 数据集对算法进行评估. 实验中使用训练集实现用户对未评分过的电影进行评分预测, 同时与测试集中的评分进行比较.

训练集包含了 952 个用户对 1 596 部电影的 9 万条评分记录(评分值为 1~5 的整数), 测试集包括了

478 个用户的 3 万条评分记录(评分值为 1~5 的整数).

本实验的用户评分的稀疏度为:

$$1 - \frac{90\,000}{952 \times 1\,596} \times 100\% = 94.08\%,$$

由此可见,评分数据相当稀疏.

### 3.2 实验度量标准

本文实验采用平均绝对偏差(mean absolute error, MAE)<sup>[15-16]</sup>和归一化折损累计增益(normalized discounted cumulative gain, NDCG)<sup>[17]</sup>作为度量算法优劣的标准.

MAE 通过计算用户的预测评分与实际评分之间的偏差来度量预测的准确性,MAE 的值越小,推荐质量越高. 设测试集中共有  $N$  条评分数据,分别为  $\{q_1, q_2, \dots, q_n\}$ ,算法对这些数据的预测值为  $\{p_1, p_2, \dots, p_n\}$ ,则算法的 MAE 为

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

MAE 值越低,表示推荐算法的精确度越高.

NDCG 常作为信息检索评价指标,其基本思想为:与查询越相关的文档排位越靠前,NDCG 的值就越大. 本文借鉴其思想,将推荐作为一个排名任务,将其应用于推荐系统评价中,用于评价推荐算法的整体推荐质量. 定义如下:

$$\text{DCG}_u = \sum_{k=1}^N \frac{2^{\text{rel}_k} - 1}{\text{lb}(k+1)},$$

式中,  $\text{DCG}_u$  表示折损累积增益;  $k$  为推荐项目所处位置;  $\text{rel}_k$  表示推荐项目列表中排序为  $k$  的项目的相关性等级. 在本文中,如果对于目标用户  $u$  的推荐项目列表中第  $k$  个项目出现在测试集中,则  $\text{rel}_k = 1$ , 否则  $\text{rel}_k = 0$ .

$$\text{NDCG}_u = \frac{\text{DCG}_u}{\text{IDCG}_u},$$

式中,  $\text{IDCG}_u$  表示最为理想的资源推荐列表,即所有推荐项目均按用户  $u$  的喜欢程度排序时的  $\text{DCG}_u$  值,本文中使用测试集中用户  $u$  最优的推荐项目列表来计算该值. NDCG 就是所有目标用户  $\text{NDCG}_u$  值的平均值. 由此可见, NDCG 是一个描述算法推荐结果整体情况好坏的评价指标.

### 3.3 实验结果

实验过程分为 4 步,首先探讨各参数对本文算法的影响,然后在最优参数基础上对本文算法和传统协同过滤算法作比较.

(1) 用户类别数  $l$  和候选邻居数  $k'$

首先考察用户类别数对本文算法的影响,将用户聚为 10~90 类,固定候选邻居数  $k' = 500$ ,时间窗口  $T = 10$ . 如图 1 所示,当用户类别数  $l$  取 50 时,算法具有最小的 MAE 值,之后的实验均取  $l = 50$ . 接下来考察候选邻居数的影响,将用户聚为 50 类,候选邻居数  $k'$  在 100~900 变动. 如图 1 所示,  $k'$  取 200 时,算法的 MAE 值达到最小;随后再增加  $k'$  值,算法的 MAE 值增大,推荐质量下降. 因此,本文取  $k' = 200$  进行实验.

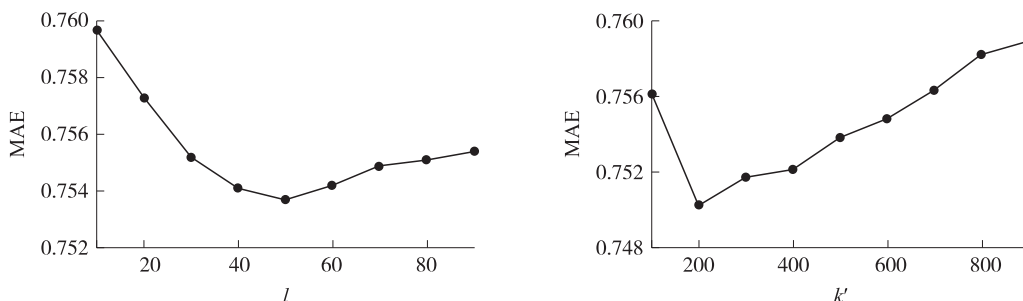


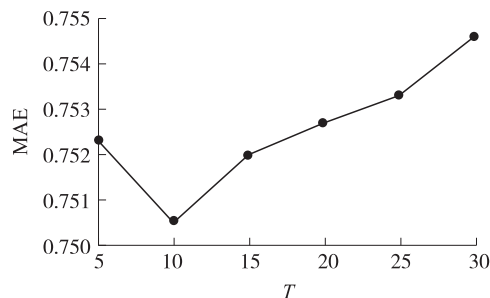
图 1 不同  $l$  值和  $k'$  值下的 MAE 值比较

Fig. 1 Comparison of MAE by different results of  $l$  and  $k'$



## (2) 时间窗口 $T$

用户兴趣会随着时间的变化而变化,为了分析时间窗口  $T$  的取值对 MAE 值的影响,将时间窗口  $T$  分别定为  $[5, 30]$  之间的整数,同时设定用户聚类  $l = 50$ , 候选邻居数  $k' = 200$ . 当  $T$  在  $[5, 30]$  之间的整数变化时对应的 MAE 曲线变化图如图 2 所示,随着时间窗口  $T$  取值的增大,MAE 值呈现先下降后上升的趋势,当时间窗口  $T$  取值为 10 时,MAE 的值最小,说明算法中  $T = 10$  时算法性能最好. 图 2 表明时间窗口的长短对预测准确性有一定的影响,时间过长则无法反映用户当前兴趣.

图 2 不同时间窗口  $T$  下的 MAE 值比较Fig. 2 Comparison of MAE by different results of  $T$ 

## (3) 与传统的协同过滤算法的比较

为了更好地检验文中基于项目聚类的动态协同过滤算法 (Proposed CF,  $l = 50, k' = 200, T = 10$ ) 的预测效果,分别与基于用户的协同过滤推荐算法 (UBCF)<sup>[18]</sup> 和基于项目的协同过滤推荐算法 (IBCF) 进行比较.

从图 3 可以清晰地看出,3 种算法的 MAE 值随着邻居数的增大而减小,当邻居数  $k$  为 50 时 3 种算法的 MAE 值最小,说明取  $k = 50$  时算法的预测精度最高. 此外,本文基于项目聚类的动态协同过滤算法的 MAE 的值明显小于另外两种算法的 MAE 值,说明本文提出的算法预测准确性更高.

从图 4 中可以看出,随着推荐资源个数的增加,各算法的 NDCG 指标均有所下降,但本文提出的算法的 NDCG 值比其他算法高,且下降后的值也能保持在较好的水平. 这说明本文算法不仅能够捕捉到用户感兴趣的项目,且能将其进行较优的排序,使得项目推荐列表整体推荐效果较优.

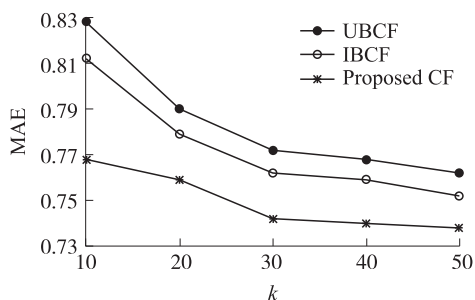


图 3 不同算法的 MAE 值比较

Fig. 3 Comparison of MAE by different algorithms

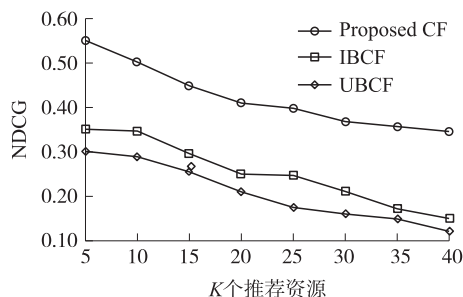


图 4 不同算法的 NDCG 值比较

Fig. 4 Comparison of NDCG by different algorithms

## 4 结语

在大数据的时代背景下,用户和数据资源的几何数级增长速度使得传统的协同过滤推荐算法暴露出数据稀疏、冷启动等问题,且传统的协同过滤算法不能准确捕捉用户兴趣随时间的变化. 基于此,本文提出了一种融合项目聚类和时间权重的动态协同过滤推荐算法,将项目聚类优化,并引入时间权重因子来及时反映用户兴趣随时间的变化.

从实验可以看出,如果参数设置得当,算法预测的准确性会有较大程度地提高,与传统的协同过滤算法的比较进一步验证了文中所提算法的可行性.

实验结果表明,本文所提算法的推荐质量较传统的协同过滤算法有显著提高,本文算法对用户有显著群体特征的系统具有很好的推荐效果. 但本文实验所用数据集为电影评分数据集,数据集中的用户没有显著群体划分,而是通过聚类算法模拟真实群体特征数据集. 未来的工作将尝试把本文算法应用于真实的群体特征数据集,以提高算法的应用价值.

## [参考文献] (References)

- [1] AL-SHAMRI M Y H. Power coefficient as a similarity measure for memory-based collaborative recommender systems[J]. Expert systems with applications, 2014, 41(13): 5 680-5 688.

- [2] CHOI K, SUH Y. A new similarity function for selecting neighbors for each target item in collaborative filtering[J]. Knowledge-based systems, 2013, 37(1): 146–153.
- [3] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报, 2003, 14(9): 1 621–1 628.  
DENG A L, ZHU Y Y, SHI B L. A collaborative filtering recommendation algorithm based on item rating prediction[J]. Journal of software, 2003, 14(9): 1 621–1 628. (in Chinese)
- [4] 罗奇, 余英, 赵呈领, 等. 自适应推荐算法在电子超市个性化服务系统中的应用研究[J]. 通信学报, 2006, 27(11): 183–186, 192.  
LUO Q, YU Y, ZHAO C L, et al. Research on personalized service system in E-supermarket by using adaptive recommendation algorithm[J]. Journal on communications, 2006, 27(11): 183–186, 192. (in Chinese)
- [5] SUGANESHWARI G, IBRAHIM S P S. A survey on collaborative filtering based recommendation system[C]//Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC - 16'). Springer International Publishing, 2016.
- [6] NAGARAJU S, KASHYAP M, BHATTACHARYA M. An effective density based approach to detect complex data clusters using notion of neighborhood difference[J]. International journal of automation and computing, 2017, 14(1): 1–11.
- [7] OMAR R, HIROTAKE O, SHIGEMI K. The robustest clusters in the input-output networks: global CO<sub>2</sub> emission clusters[J]. Journal of economic structures, 2017, 6(1): 3.
- [8] 黄典. 基于项目的协同过滤推荐算法的改进[J]. 中国科技信息, 2016(1): 64–66.  
HUANG D. Improvement of item-based collaborative filtering recommendation algorithm[J]. China science and technology information, 2016(1): 64–66. (in Chinese)
- [9] ZHANG Y, LIU Y. A collaborative filtering algorithm based on time period partition[C]//Third International Symposium on Intelligent Information Technology and Security Informatics. New York: IEEE Computer Society, 2010: 777–780.
- [10] 田伟, 彭玉青. 基于电子商务应用的协同过滤技术改进综述[J]. 计算机工程与科学, 2008, 30(10): 61–63, 66.  
TIAN W, PENG Y Q. Improvement research of the CF algorithm for E-commerce[J]. Computer engineering and science, 2008, 30(10): 61–63, 66. (in Chinese)
- [11] 陆诗琴. 个性化推荐技术中的互信息相似度应用研究[D]. 桂林: 桂林理工大学, 2015.  
LU S Q. Research on the application of mutual information similarity in personalized recommendation technology[D]. Guilin: Guilin University of Technology, 2015. (in Chinese)
- [12] RESNICK P, IACOVOU N, SUCHAK M, et al. Group Lens: an open architecture for collaborative filtering of net news[C]//ACM Conference on Computer Supported Cooperative Work. ACM, 1994: 175–186.
- [13] 朱文奇. 推荐系统用户相似度计算方法研究[D]. 重庆: 重庆大学, 2014.  
ZHU W Q. Research on user's similarity calculation method[D]. Chongqing: Chongqing University, 2014. (in Chinese)
- [14] KIM B M, LI Q, CHANG S P. A new approach for combining content-based and collaborative filters[J]. Journal of intelligent information system, 2006, 27(1): 79–91.
- [15] YANG X, GUO Y, LIU Y. A survey of collaborative filtering based social recommender systems[J]. Computer communications, 2014, 41(5): 1–10.
- [16] 梁昌勇, 冷亚军, 王勇胜, 等. 电子商务推荐系统中群体用户推荐问题研究[J]. 中国管理科学, 2013, 21(3): 153–158.  
LIANG C Y, LENG Y J, WANG Y S, et al. Research on group recommendation in e-commerce recommender systems[J]. Chinese journal of management science, 2013, 21(3): 153–158. (in Chinese)
- [17] 魏强, 金芝, 许焱. 基于概率主题模型的物联网服务发现[J]. 软件学报, 2014(8): 1 640–1 658.  
WEI Q, JIN Z, XU Y. Service discovery for internet of things based on probabilistic topic model[J]. Journal of software, 2014(8): 1 640–1 658. (in Chinese)
- [18] ZHAO Z D, SHANG M S. User-based collaborative-filtering recommendation algorithms on Hadoop[C]//Third international conference on knowledge discovery and data mining. New York: IEEE Computer Society, 2010: 478–481.

[责任编辑: 严海琳]