

文章编号: 1004-4353(2019)03-0234-07

# 基于标签和协同过滤的改进推荐算法研究

金晶, 怀丽波\*

( 延边大学 工学院, 吉林 延吉 133002 )

**摘要:** 针对基于标签和协同过滤的个性化推荐(TCF)没有考虑评分数据的作用和用户兴趣标签稀疏的问题, 提出了一种加入评分数据并扩展用户兴趣标签的基于标签和协同过滤的改进推荐算法(ITCF). 首先, 以项目-标签相关度构造项目特征向量, 并结合评分构造用户特征向量和用户-标签关联度; 其次, 对用户的历史偏好标签集进行基于标签相似性和基于近邻用户偏好的扩展; 最后, 以 MovieLens 数据集为例对 ITCF 算法的有效性进行实验验证. 实验结果表明, 在稠密的数据集中, ITCF 算法的平均准确率和平均召回率比文献[2]和[3]算法的平均准确率和平均召回率分别提升约 2.0% 和 1.7%; 在稀疏的数据集中, 当推荐项目数不超过 20 时, ITCF 算法的平均准确率和平均召回率约比文献[2]和文献[3]算法的平均准确率和平均召回率分别提升约 0.2% 和 0.8%. 因此, 本文提出的 ITCF 算法具有较好的应用前景.

**关键词:** 推荐算法; 协同过滤; 标签扩展; 召回率; 准确率

**中图分类号:** TP391

**文献标识码:** A

## Research on improved recommender algorithm based tag and collaborative filtering

JIN Jing, HUAI Libo\*

( College of Engineering, Yanbian University, Yanji 133002, China )

**Abstract:** Aiming at the problem that the personalized recommendation based on tag and collaborative filtering (TCF) does not consider the function of rating data and spare user interest tags, this paper proposes an improved recommendation algorithm based on the tag and the collaborative filtering (ITCF), which adds rating data and extends user interest tag. Firstly, the item feature vector is constructed by the item-tag correlation degree, and the user feature vector and the user-tag correlation degree are constructed by combining the rate. Secondly, the tag set of users' historical preference is extended based on tag similarity and nearest neighbor user preference. Finally, the experiment is carried out to verify the efficiency of ITCF algorithm on the MovieLens data set. The experimental results show that compared with that of reference [2] and [3] TCF algorithm, the precision and recall rate of the proposed ITCF algorithm are increased by 2.0% and 1.7% in the dense data set, and 0.2% and 0.8% in the sparse data set when the number of recommended items does not exceed 20. Therefore, the ITCF algorithm proposed in this paper has a good application prospect.

**Keywords:** recommendation algorithm; cooperative filtering; tag extension; recall rate; precision rate

## 0 引言

个性化推荐因有助于用户快速地获取所需信息, 目前被广泛地应用于各种社交网络、音视频娱乐网站以及网络商务平台等. 目前, 推荐算法主要有协同过滤、混合推荐、基于内容和基于人口统计的推荐

等<sup>[1]</sup>,但这些方法普遍存在数据稀疏、冷启动和可扩展性的问题.为缓解算法中存在的冷启动和数据稀疏问题,一些学者对此进行了研究.例如:刘健以 TF-IDF(term frequency-inverse document frequency)计算用户-标签相关度和项目-标签相关度的值,并依此构建了用户兴趣模型,提高了推荐精度<sup>[2]</sup>;蔡强等以 TF-IDF 方法分别生成用户和资源的标签特征向量来构造用户对资源的偏好矩阵,以余弦相似度(SIM)计算了资源相似性,该方法大大降低了计算的复杂度<sup>[3]</sup><sup>[7]</sup>.针对标签无法准确地反映用户的喜好程度的问题,郭彩云等利用标签、用户评分值以及指数遗忘函数等评价指标捕捉了用户兴趣的变化程度,得到了较好的推荐效果<sup>[4]</sup>.对于标签的语义歧义问题,李昊阳等用标签生成主题标签簇,然后通过对目标项目预测产生推荐目录,以此提高了推荐精度<sup>[5]</sup>;G.Pitsilis 等使用标签聚类方法提高了推荐的准确性,并降低了信息过载<sup>[6]</sup>.针对标签推荐中忽略项目和标签间具有相互作用的问题,Zheng Xiaolin 等提出了一种可解释性的项目标签联合推荐方法,该方法不仅可提高推荐的准确性,同时也可缓解冷启动的问题<sup>[7]</sup>.但在上述研究中,研究者均没有综合考虑评分数据和标签数据的作用,仍存在标签数据稀少的问题.基于此,本文在基于标签和协同过滤的个性化推荐算法的基础上,提出一种基于标签和协同过滤的改进推荐算法,并通过实验验证本文方法的有效性.

## 1 TCF 算法介绍

### 1.1 基于标签的推荐算法

基于标签的协同过滤推荐算法(TCF)是在传统的协同过滤算法中引入标签因子,即通过标签反映用户偏好和项目特征的优势,以此改进推荐质量.在信息推荐时,标签通常被分成用户-标签、项目-标签两个部分来挖掘用户的喜好,并依此构造用户兴趣模型.

定义用户集合  $U = \{u_1, u_2, \dots, u_i, \dots, u_m\}$ ,  $m$  为用户总数;标签集  $T = \{t_1, t_2, \dots, t_k, \dots, t_l\}$ ,  $l$  为标签总数.项目集  $I = \{i_1, i_2, \dots, i_j, \dots, i_n\}$ ,  $n$  是项目总数.用户-标签相关度( $UT_{u,t}$ )的计算公式<sup>[3]</sup><sup>[7]</sup>为:

$$UT_{u,t} = \log\left(\frac{m}{N_{m_t} + 1}\right) \times \frac{N_{u,t}}{N_{l_u}}. \quad (1)$$

其中  $N_{m_t}$  代表打标签  $t$  的用户总数,  $N_{u,t}$  代表用户  $u$  打标签  $t$  的次数,  $N_{l_u}$  代表用户  $u$  打标签的次数总和.项目-标签相关度( $IT_{i,t}$ )的计算公式<sup>[3]</sup><sup>[7]</sup>为:

$$IT_{i,t} = \log\left(\frac{n}{N_{n_t} + 1}\right) \times \frac{N_{i,t}}{N_{l_i}}, \quad (2)$$

其中  $N_{n_t}$  代表被打标签  $t$  的项目总数,  $N_{i,t}$  代表项目  $i$  被打标签  $t$  的次数,  $N_{l_i}$  代表项目  $i$  被打标签的次数总和.

构建用户兴趣模型时先分别以公式(1)和(2)建立相关矩阵  $UT_{m \times l}$ 、 $IT_{n \times l}$ ,然后再利用  $IT_{n \times l}$  的转置矩阵  $TI_{l \times n}$  计算用户  $u_i$  对已有用户行为的项目  $i_j$  的喜爱程度  $HP_{u_i, i_j}$ ,以此获得用户的历史兴趣矩阵  $HP_{m \times n}$ .  $HP_{u_i, i_j}$  的计算公式<sup>[3]</sup><sup>[7]</sup>为:

$$HP_{u_i, i_j} = \sum_{t_k \in T(i_j)} UT_{u_i, t_k} \times TI_{t_k, i_j}, \quad (3)$$

其中  $T(i_j)$  是项目  $i_j$  的标签集.

### 1.2 基于协同过滤的推荐算法

协同过滤推荐是指建立好用户兴趣模型后,以项目间的相似性预测用户  $u_i$  对未操作项目  $i_i$  的偏爱程度( $PP_{u_i, i_i}$ ),并以此构建矩阵  $PP_{m \times n}$ .然后将  $PP_{m \times n}$  的每行按照由高到低的顺序排序,取前  $N$  个进行推荐.协同过滤推荐的原理简单,因此目前被广泛采用在推荐系统领域中<sup>[8-9]</sup>.  $PP_{u_i, i_i}$  的计算公式<sup>[3]</sup><sup>[7]</sup>为:

$$PP_{u_i, i_i} = \sum_{i_j \in I(u_i), i_i \in \text{Not}I(u_i)} HP_{u_i, i_j} \times S1_{i_i, i_j}, \quad (4)$$

其中  $\text{Not}I(u_i)$  是用户  $u_i$  未使用的项目集,  $I(u_i)$  是用户  $u_i$  的历史项目集,  $S1_{i_i, i_j}$  是项目  $i_i$  和  $i_j$  的余弦相

似度.

## 2 融入评分并扩展标签的 TCF 改进推荐算法

针对标签数据稀少的问题,本文提出一种基于标签和协同过滤的改进推荐算法(ITCF).首先利用项目-标签相关度构造项目特征向量,并结合评分构造用户-标签关联度,然后对用户的偏好标签集进行基于标签相似性和基于邻居用户偏好的扩展.

### 2.1 项目特征向量的构建

因为项目-标签相关度能很好地代表项目的内容特性,所以本文以项目-标签相关度  $IT_{ij,t_k}$  表示项目  $i_j$  的特征向量  $i_j$ ,其定义为  $i_j = (IT_{ij,t_1}, IT_{ij,t_2}, \dots, IT_{ij,t_k}, \dots, IT_{ij,t_l})$ . 该特征向量是构建项目-标签相关矩阵  $IT_{n \times l}$  和项目相似矩阵  $SI_{n \times n}$  的计算依据.构造项目-标签相关矩阵和项目相似矩阵的算法步骤如下:

Step 1 统计每个项目  $i_j$  的标签集  $T(i_j)$  和标签  $t_k$  标记项目  $i_j$  的次数  $n_{ij,t_k}$ ;

Step 2 记录项目总数  $n$  和 step 1 中的标签集个数  $N(T(i_j))$ ;

Step 3 采用 TF-IDF 计算项目-标签相关度  $IT_{ij,t_k}$ , 建立  $IT_{n \times l}$  矩阵;

Step 4 取  $IT_{n \times l}$  矩阵的任意两个行向量(即特征向量)进行余弦相似度计算,得到项目相似程度  $SI_{i_i,i_j}$  并建立矩阵  $SI_{n \times n}$ .

### 2.2 结合评分的用户-标签关联度的计算

以用户  $u_i$  打过标签的项目集  $I(u_i)$  的评分集  $R(I(u_i))$  表示用户  $u_i$  的特征向量  $u_i$ ,  $u_i = (r_{u_i,i_1}, r_{u_i,i_2}, \dots, r_{u_i,i_j}, \dots, r_{u_i,i_n})$ . 因为用户-标签相关度的大小通常与用户使用标签次数的多少和用户对该标签项目的评分高低有关,所以本文利用公式(5)来表示用户对标签的喜爱程度.

$$UT_{u,t} = \sum_{i \in I(u), t \in T(u)} \frac{r_{u,i}}{\sum_{i \in I(u)} r_{u,i}} \times IT_{i,t}, \quad (5)$$

其中  $u$  是用户集中的用户,  $I(u)$  是用户  $u$  打过标签和评分的项目集,  $T(u)$  是用户项目集中含有的标签集,  $UT_{u,t}$  是用户-标签偏爱程度,  $IT_{i,t}$  是采取 TF-IDF 计算的项目-标签相关度.

建立用户-标签相关矩阵的算法步骤如下:

Step 1 统计每个用户  $u$  的未使用项目集  $NotI(u)$ 、使用过的项目集  $I(u)$  及对应的评分集  $R(I(u))$ , 并以  $R(I(u))$  表示用户的特征向量  $u$ ;

Step 2 通过余弦相似度计算用户特征向量  $u_i$  和  $u_j$  的相似度  $S2_{u_i,u_j}$ , 并建立相似度矩阵  $S2_{m \times m}$ ;

Step 3 统计每个用户  $u$  使用过的标签集  $T(u)$  及未使用的标签集  $NotT(u)$ ;

Step 4 以公式(5)计算用户-标签相关度  $UT_{u,t}$ , 并以此建立  $UT_{m \times l}$  矩阵.

### 2.3 用户偏好标签集的扩展

**2.3.1 基于近邻用户的标签扩展** 用户在接受推荐时,因对兴趣相同的用户(即近邻用户)所推荐的项目更感兴趣,所以可以根据用户  $u_j$  近邻用户的用户-标签相关度计算  $u_j$  对没有历史行为的标签  $t$  的相关度  $UT1_{u_j,t}$ . 本文将  $UT1_{u_j,t}$  的计算公式定义为:

$$UT1_{u_j,t} = \sum_{u_i \in Nei(u_j)} \frac{S2_{u_j,u_i} \times UT_{u_i,t}}{\sum_{u_i \in Nei(u_j)} S2_{u_j,u_i}}, \quad (6)$$

其中  $UT1_{u_j,t}$  是经近邻扩充得到的用户  $u_j$  对标签  $t$  的偏爱程度,  $Nei(u_j)$  是与用户  $u_j$  相似度最高的前  $K$  个用户组成的近邻集合,  $S2_{u_j,u_i}$  是用户与近邻用户间的相似度.

基于近邻用户的标签扩展的算法步骤为:

Step 1 对  $S2_{m \times m}$  的每行进行相似度排序,并取前  $K$  个相似的用户集  $Nei(u_j)$ ;

Step 2 统计邻居用户  $u_i \in Nei(u_j)$  使用过而用户  $u_j$  未使用过的标签集  $T(Nei(u_j))$ ;

Step 3 利用公式(6) 计算基于邻居用户标签的扩展用户-标签相关度  $UT1_{u,j,t}$  的值,并构建用户-标签矩阵  $UTI_{m \times l}$ .

**2.3.2 基于标签相似度的标签扩展** 本文根据标签相似度对用户-标签矩阵进行扩充,即根据标签的相似度引入用户对未有历史行为的标签的喜爱程度( $UT2_{u,t_j}$ ),其计算公式定义如下:

$$UT2_{u,t_j} = \sum_{t_j \in NotT(u), t_i \in T(u)} \frac{S3_{t_j,t_i} \times UT_{u,t_i}}{\sum_{t_i \in T(u)} S3_{t_j,t_i}}. \quad (7)$$

其中  $UT2_{u,t_j}$  是以标签相似度扩充的用户  $u$  对标签  $t_j$  的喜爱程度,  $NotT(u)$  是不在原标签集里的标签,  $S3_{t_j,t_i}$  是不在原用户标签集里的标签  $t_j$  与原标签  $t_i$  的相似度.

标签相似度的计算公式<sup>[10]</sup> 为:

$$S3_{t_j,t_i} = \frac{\sum_{i \in I(t_j) \cap I(t_i)} n_{i,t_j} \times n_{i,t_i}}{\sqrt{\sum_{i \in I(t_j)} n_{i,t_j}^2 \sum_{i \in I(t_i)} n_{i,t_i}^2}}, \quad (8)$$

其中  $t_j$  和  $t_i$  表示两个不同的标签,  $I(t_i)$  表示标签  $t_i$  的项目集,  $n_{i,t_i}$  表示对项目  $i$  标注标签  $t_i$  的次数.

基于标签相似度的标签扩展算法的步骤为:

Step 1 统计每个项目集  $I(t_i)$ ;

Step 2 利用式(8) 计算标签间的相似程度  $S3_{t_j,t_i}$ , 并建立  $S3_{l \times l}$  矩阵;

Step 3 采用式(7) 计算基于相似标签的扩展用户-标签相关度  $UT2_{u,t_j}$  的值,并构建  $UT2_{m \times l}$  矩阵.

**2.3.3 融入标签扩展的用户-标签相关矩阵** 本文将融入标签扩展的用户-标签相关度的计算公式定义为:

$$UT3_{u,t} = \begin{cases} UT_{u,t}, & t \in T(u); \\ UT1_{u,t}, & t \in TE1(u) - TE2(u); \\ UT2_{u,t}, & t \in TE2(u) - TE1(u); \\ \alpha UT1_{u,t} + (1 - \alpha) UT2_{u,t}, & t \in TE1(u) \cap TE2(u). \end{cases} \quad (9)$$

其中,  $TE1(u)$  是用户  $u$  基于近邻用户的标签扩展所得到的前  $E$  个标签组成的标签集,  $TE2(u)$  是基于标签相似度的标签扩展所得到的前  $E$  个标签组成的标签集,  $\alpha$  是赋予  $UT1_{u,t}$  的结合权重(用于  $UT1$  和  $UT2$  的信息融合). 考虑到二者对预测用户-标签相关度的影响同样重要,因此本文将  $\alpha$  取为 0.5. 由此得到的  $UT3_{u,t}$  即为包含历史行为标签和扩展后标签的用户-标签相关度,其相关度的矩阵为  $UT3_{m \times l}$ .

## 2.4 ITCF 过程及算法描述

输入:  $\langle$ 用户,项目,标签,评分值 $\rangle$ ,邻居数  $K$ ,推荐个数  $N$ ,近邻用户、标签扩展标签数  $E$ .

输出: 输入的用户偏好的  $TopN$  推荐列表.

步骤 1 按 2.1 中的步骤得到每个项目  $i_j$  的标签集  $T(i_j)$ 、项目-标签关联矩阵  $IT_{n \times l}$ 、项目相似程度矩阵  $SI_{n \times n}$ .

步骤 2 按 2.2 中的步骤得到每个用户  $u_i$  的历史项目集  $I(u_i)$  所未使用过的项目集  $NotI(u_i)$ 、用户相似程度矩阵  $S2_{m \times m}$  以及用户-标签关联矩阵  $UT_{m \times l}$ .

步骤 3 按 2.3 中的公式得到标签相似程度矩阵  $S3_{l \times l}$ 、综合扩展后的用户-标签矩阵  $UT3_{m \times l}$ ,并赋值给  $UT_{m \times l}$ .

步骤 4 对  $i_i \in NotI(u_i)$ ,  $i_j \in I(u_i)$ ,  $t_k \in T(i_j)$ ,根据  $SI_{n \times n}$ 、 $UT_{m \times l}$  和  $IT_{n \times l}$  的转置矩阵  $TI_{l \times n}$ ,按照式(3) 建立用户兴趣模型,按式(4) 预测用户  $u_i$  对未操作项目  $i_j$  的偏爱程度  $PP_{u_i,i_j}$ ,并以此构建矩阵  $PP_{m \times n}$ .

步骤 5 对  $PP_{m \times n}$  的每行进行排序,记录所有行前  $N$  个标签的集合,由此得到推荐列表  $Rcmd(u)$ .

3 实验结果与分析

3.1 实验数据与环境

实验采用 MovieLens 数据集(10 M),实验环境为 Intel 1.60 GHz CPU, 4.00 GB 运行内存的 PC 机. 为了更准确地验证算法的有效性,实验选取标签数大于 10 的标签集,以排除冷门标签对用户和项目的影 响. 另外,为分析算法在稠密数据集和稀疏数据集下的性能,本文选取标签数量为前 98 位的用户和标签数在 10~100 的用户集来进行分析. 数据集的分类如表 1 所示.

表 1 数据集分类

数据集	用户数	电影数	标签数
data 1	98	1 220	789
data 2	335	1 093	780

3.2 评价标准

本文采用准确率(Recall)和召回率(Precision)对推荐算法进行性能评测,其计算公式如下:

$$Recall = \frac{\sum_{u \in U} |Rcmd(u) \cap Test(u)|}{\sum_{u \in U} |Test(u)|},$$

(10)

$$Precision = \frac{\sum_{u \in U} |Rcmd(u) \cap Test(u)|}{\sum_{u \in U} |Rcmd(u)|}.$$

(11)

其中  $U$  表示用户集,  $|Rcmd(u)|$  表示推荐集中的项目个数,  $|Test(u)|$  表示测试集中的项目个数,  $|Rcmd(u) \cap Test(u)|$  表示两个集合的交集个数.

3.3 实验结果分析

根据文献[11],测试每组样例集时,邻居用户数  $K$  取 5,扩展标签数  $E$  取 40. 推荐个数  $N$  的值分别取 5、10、15、20、25、30、35、40 和 45 进行测试,并与文献[2]和[3]中的两种 TCF 算法进行比较,实验结果见表 2—表 5 和图 1—图 4.

表 2 不同算法对 data 1 的准确率

算法	不同推荐数目下的准确率								
	5	10	15	20	25	30	35	40	45
ITCF	0.096	0.074	0.065	0.060	0.056	0.052	0.049	0.046	0.044
文献[2]中的 TCF	0.063	0.054	0.047	0.043	0.041	0.037	0.036	0.034	0.034
文献[3]中的 TCF	0.058	0.047	0.037	0.045	0.030	0.030	0.030	0.030	0.029

表 3 不同算法对 data 1 的召回率

算法	不同推荐数目下的召回率								
	5	10	15	20	25	30	35	40	45
ITCF	0.019	0.030	0.039	0.049	0.057	0.063	0.069	0.074	0.081
文献[2]中的 TCF	0.013	0.022	0.029	0.035	0.049	0.045	0.051	0.055	0.062
文献[3]中的 TCF	0.012	0.019	0.022	0.026	0.030	0.037	0.043	0.048	0.053

表 4 不同算法对 data 2 的准确率

算法	不同推荐数目下的准确率								
	5	10	15	20	25	30	35	40	45
ITCF	0.028	0.018	0.015	0.013	0.011	0.010	0.009	0.008	0.008
文献[2]中的 TCF	0.026	0.016	0.013	0.011	0.011	0.010	0.009	0.008	0.008
文献[3]中的 TCF	0.023	0.016	0.014	0.013	0.012	0.010	0.009	0.009	0.009

表 5 不同算法对 data 2 的召回率

算法	不同推荐数目下的召回率								
	5	10	15	20	25	30	35	40	45
ITCF	0.055	0.072	0.089	0.102	0.108	0.118	0.121	0.127	0.133
文献[2]中的 TCF	0.050	0.064	0.076	0.089	0.107	0.115	0.119	0.130	0.137
文献[3]中的 TCF	0.046	0.064	0.082	0.098	0.113	0.122	0.127	0.147	0.156

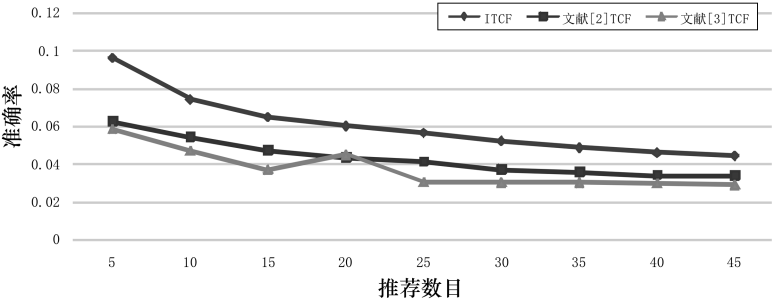


图 1 不同算法对 data 1 的准确率

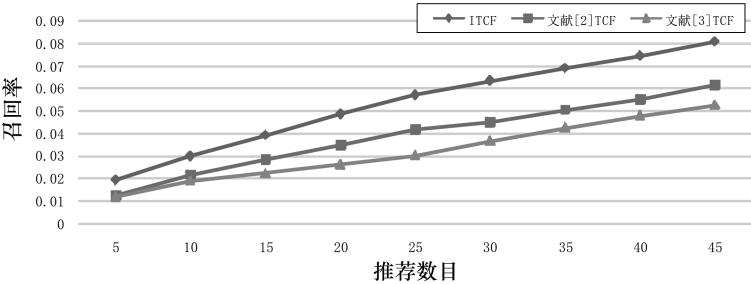


图 2 不同算法对 data 1 的召回率

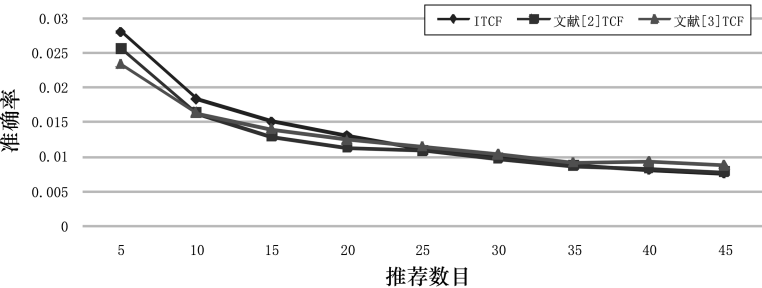


图 3 不同算法对 data 2 的准确率

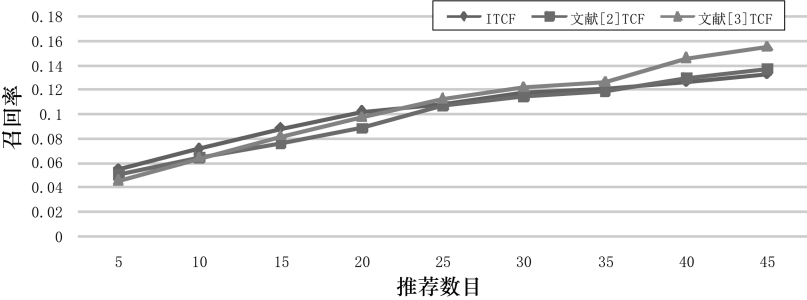


图 4 不同算法对 data 2 的召回率

由图 1 和图 2 可以看出,在稠密的数据集中,ITCF 的平均准确率和平均召回率比文献[2]和文献



[3]的平均准确率和平均召回率分别提升约 2.0%和 1.7%。由图 3 和图 4 可以看出,在稀疏的数据集中,当  $5 \leq N \leq 20$  时,ITCF 的准确率和召回率均高于其他两种 TCF 算法,其中准确率约提升 0.2%,召回率约提升 0.8%; $N > 20$  时,ITCF 算法的准确率和召回率均略低于文献[3]的 TCF 算法,但与文献[2]的算法相近。其原因是当  $N > 20$  时,数据集中的很多项目已不存在相似标签,因此导致 ITCF 的准确率和召回率降低。

Data 1 的准确率和召回率比 data 2 的准确率、召回率提升得更高,其原因是由于 data 1 利用用户相似度和标签相似度扩充了用户-标签矩阵,因此在一定程度上缓解了数据稀少问题,进而提高了准确率和召回率。而 data 2 因用户历史行为数据过于稀少、训练样本不足,因此导致标签相似度普遍较低,进而使得其准确率和召回率提升幅度低于 data 1。

## 4 结论

研究表明,本文提出的基于标签和协同过滤的改进推荐算法(ITCF)与文献[2-3]相比,其准确率和召回率均有所提高,因此本文方法对因标签过少而影响推荐质量的问题具有一定程度地缓解作用。本文在研究中未考虑用户兴趣的动态变化情况,在推荐实时性方面有所欠缺,因此我们在今后的研究中将引入时间因子,建立随时间变化的动态用户兴趣模型,以更好地完善本文模型。

## 参考文献:

- [1] 黄泽明. 基于主题模型的学术论文推荐系统研究[D]. 辽宁:大连海事大学,2013.
- [2] 刘健. 基于标签和协同过滤的个性化推荐算法[J]. 计算机与现代化,2016(2):62-65.
- [3] 蔡强,韩东梅,李海生,等. 基于标签和协同过滤的个性化资源推荐[J]. 计算机科学,2014,41(1):69-71.
- [4] 郭彩云,王会进. 改进的基于标签的协同过滤算法[J]. 计算机工程与应用,2016,52(8):56-61.
- [5] 李昊阳,符云清. 基于标签聚类与项目主题的协同过滤推荐算法[J]. 计算机科学,2018,45(4):247-251.
- [6] PITSILIS Georgios, WANG Wei. Harnessing the power of social bookmarking for improving tag-based recommendations[J]. Computers in Human Behavior, 2015,50:239-251.
- [7] ZHENG Xiaolin, WANG Menghan, CHEN Chaochao, et al. EXPLORE: EXPLainable item-tag CO-REcommen-dation[J]. Information Sciences, 2019,474:170-186.
- [8] PARRA-SANTANDER D, BRUSILOVSKY P. Improving collaborative filtering in social tagging systems for the recommendation of scientific articles[C]//Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Piscataway, USA: IEEE, 2010:136-142.
- [9] WANG Chong, BLEI D M. Collaborative topic modeling for recommending scientific articles[C]//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2011:448-456.
- [10] 项亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012:108.
- [11] 查叶飞. 基于可信机制及用户偏好模型的推荐技术的研究与应用[D]. 南京:东南大学,2015.