

# 提取关键字改进协同过滤算法的研究与应用

李吉祺, 黄 刚

(南京邮电大学 计算机学院, 江苏 南京 210000)

**摘 要:**协同过滤算法在遇到数据稀疏性问题时,其相似度计算过程会受到很大的影响,导致推荐结果不准确,影响推荐系统用户体验。而影评网站的影评往往很好地概括了电影的特征,从影评网站的影评文字中可以使用关键字提取算法提取特征来进行电影间的相似性计算。TF-IDF 是一种高效而常用的关键词提取技术,其通过特定文档中词的相对频率和整个文档语料库中该词的反比例进行比较,最终得出该篇文章的关键字。利用文本信息提取关键字,进而通过文章的关键字进行文章的相似度计算,可以有效地改进评价矩阵稀疏的问题。通过爬取电影的评价文字来进行关键字提取,改进评分矩阵较稀疏的电影的相似度计算,可以弥补稀疏矩阵的缺陷。实验结果表明,该算法有效提高了准确率、召回率和覆盖率,证明了算法的可行性。

**关键词:**推荐系统;协同过滤;稀疏矩阵;词频与逆文本频率指数;混合推荐

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2019)06-0154-05

doi:10.3969/j.issn.1673-629X.2019.06.032

## Research and Application of Improved Collaborative Filtering Algorithm of Keyword Extraction

LI Ji-qi, HUANG Gang

(School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210000, China)

**Abstract:**When the collaborative filtering algorithm is influenced by data sparsity, its similarity calculation process will be greatly affected, resulting in inaccurate recommendation and affecting the user experience of the recommendation system. The movie reviews on movie review websites often summarize the characteristics of movie, where keyword extraction algorithm can be used to extract features to calculate the similarity between movies. The TF-IDF is an efficient and commonly used keyword extraction technique, which compares the relative frequency of words in a specific document with the inverse proportion of the words in the entire document, and finally derives the keywords of the article. Using text information to extract keywords and then calculating the similarity of articles through the keyword words of the article can effectively improve the sparse evaluation matrix. To make up for the defects of the sparse matrix, the keyword can be extracted by crawling the movie reviews of the movie. Experiment shows that the proposed algorithm, which is proved to be feasible, can effectively improve the accuracy, recall rate and coverage.

**Key words:** recommendation system; collaborative filtering; sparse matrix; TF-IDF; mixed recommendation

## 0 引言

推荐系统是一个古老的问题,也是大数据与人工智能的一个非常好的落脚点。传统协同过滤算法至今仍然在各种场景下发挥着巨大作用,其一般基于评分矩阵的每一列或者每一行的评分值来进行相似度计算,最终依据一定的推荐策略给出推荐列表。协同过滤算法虽然经过前人无数次的改进,但是在面对评价信息较少,或者没有评价信息的时候,很难发挥其作用。因此,挖掘出物品背后更多的信息来弥补传统协

同过滤算法的不足是一个很重要的议题。

## 1 背景知识与问题

近年来,互联网的快速发展带来了一个很严重的问题:信息过载。在网络中人们面临着太多的数据,已经无法高效率地筛选出有价值的数据了。但是从另外一方面来说,互联网信息量的指数级增长,也意味着可以挖掘的数据正以惊人的速度增长。但是,很多信息并没有做到恰当的“映射”,甚至包括很多应当信任可

收稿日期:2018-07-15

修回日期:2018-11-21

网络出版时间:2019-03--06

基金项目:国家自然科学基金(61171053);南京邮电大学基金(SG1107)

作者简介:李吉祺(1994-),男,硕士研究生,研究方向为数据挖掘与推荐系统;黄 刚,教授,研究方向为计算机软件理论及应用。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190306.0938.056.html>

以利用的数据。

解决信息过载问题一直是研究的重点,前人也提出了很多解决方案。自动化的推荐系统就是一个很好的帮用户过滤信息的方案。

推荐系统在互联网经济的很多方面扮演着至关重要的角色,例如社交网络、物品推荐(电影、音乐等)。京东、天猫和豆瓣等许多互联网,都已经在推荐系统领域深耕多年,采用推荐技术来估计客户的潜在偏好,针对性地向用户推荐相关产品。由于其在用户满意度和商业上的出色表现,推荐系统已经对实体经济产生了巨大影响。根据推荐系统计算的数据类型,以及在推荐系统中使用数据的方法,可以将其分成基于内容的推荐算法(content-based, CB),协同过滤(collaborative filtering, CF)和混合算法。例如基于内容的推荐算法广泛应用于推荐系统设计和使用中,其使用物品的内容创建特征和属性来匹配给用户推荐的物品。系统中的物品将依次与以前用户喜欢的项目进行比较,然后推荐给用户相似度最高的项目<sup>[1]</sup>。所以,基于内容的推荐算法的主要问题就是推荐系统需要去学习用户的偏好,然后才能计算出其他物品与之的相关性。

协同过滤算法是目前推荐系统中最流行的算法,其利用过去从用户行为中收集的大量数据来预测用户会喜欢哪些项目,但并不需要去分析物品的内容,取而代之的是计算用户和物品之间的关系。如表1所示,这些关系依靠于一个以评分作为反馈的矩阵,每个元素代表特定用户对特定物品的评分。矩阵纵坐标是物品编号,横坐标是用户编号,已经有数值的即为该用户对该物品的打分,分数为1到5分,分数越高表明用户越喜爱这个物品。协同过滤算法的任务就是通过计算用户-物品评分矩阵每一行或者每一列的相似性,来预测该矩阵中的缺失值。

表1 评分矩阵示例

	user <sub>1</sub>	user <sub>2</sub>	user <sub>3</sub>	user <sub>4</sub>	user <sub>5</sub>
item <sub>1</sub>	2	5	5	5	2
item <sub>2</sub>	3	4	5	5	3
item <sub>3</sub>	5	2	2	4	3
item <sub>4</sub>	4	1	3	3	4
item <sub>5</sub>	2	5	4	3	Null

但是在实际应用时,协同过滤算法存在很严重的稀疏性和冷启动(cold start)的问题。例如Netflix Prize大赛,世界著名的电影推荐大赛,其数据集中大约有48万多用户提供的关于18 000多部电影的共1亿多个评分,但是其评分数只占用了矩阵的大约1%。使用如此稀疏的评分矩阵来计算物品或者用户之间的相似度,一定程度上来说是非常危险的,矩阵越稀疏,

数据带给结果的随机性就越大。另外一个问题就是冷启动问题。当一个物品刚刚上市的时候,推荐系统中并没有对应的评分矩阵来计算其与其他物品的相似性。协同过滤算法需要大量来自用户对物品的评分才能对物品进行有效的推荐,如果推荐系统中可用的评分很少,推荐系统就无法为新用户和新物品做出准确的判断。

关键字提取可以作为一段文字的绝佳入口,发挥很大的作用<sup>[2]</sup>。比如在豆瓣等影评网站,用户通过各种电影的评价文字可以发掘出很多的信息。然而很多内容评价网站的评价关键字,仍然是用户自己或者依靠管理员手动添加上去的,这无疑给关键字本身的获取带来了一定的困难。TF-IDF(term frequency-inverse document frequency, 词频与逆文本指数)是一种常用的关键字加权提取技术,其主要计算的就是一个词对于该文档以及文档集合的重要性。

为了解决推荐矩阵的稀疏性问题,文中提出了一种提取影评关键字来辅助改进传统协同过滤推荐系统的模型。

2 相关工作

2.1 协同过滤算法

2.1.1 相似度计算过程

基于前文的描述,基于物品的协同过滤算法主要分为两大步骤<sup>[1-7]</sup>:

- (1) 计算系统内所有物品的相似度;
- (2) 根据用户历史评分物品记录找到高分物品,然后推荐高分物品的相似物品。

而物品间相似度的计算方法又分为以下几种:

(1) 根据基于物品的协同过滤算法的设计初衷(购买了该物品的用户同时也购买其他物品),定义物品相似度计算公式如下:

$$\text{sim}(i,j)=\frac{|N(i)\cap N(j)|}{|N(i)|}$$

(1)

其中,  $N(i)$  表示喜欢物品  $i$  的用户数;  $N(i) \cap N(j)$  表示同时喜欢物品  $i$  和物品  $j$  的用户数。

但是,式1却面临着推荐系统中“长尾问题”的困扰,即很多热门电影是大部分人都会喜欢的,所以热门物品间会被计算为高度相似的物品,这显然会影响推荐系统的准确性。所以,可以将该公式改进为:

$$\text{sim}(i,j)=\frac{|N(i)\cap N(j)|}{\sqrt{|N(i)||N(j)|}}$$

(2)

在该公式下,给热门物品进行了降权,即惩罚了物品  $j$  的权重,因此可以在一定程度上避免热门物品对物品相似度计算的影响<sup>[8-9]</sup>。

- (2) 余弦相似度。

$$\text{sim}(i, j) = \cos(i, j) = \frac{\sum_{n \in N_{ij}} R_{i,n} R_{j,n}}{\sqrt{\sum_{n \in N_i} (R_{i,n})^2} \sqrt{\sum_{n \in N_j} (R_{j,n})^2}} \quad (3)$$

其中,  $R_{i,n}$  和  $R_{j,n}$  分别表示用户  $n$  对物品  $i$  和物品  $j$  的评分;  $N_{ij}$  表示对物品  $i$  和物品  $j$  的共同评分用户集合。

如果运用余弦相似度计算公式, 物品的所有评分数据就会被看作是  $N$  维用户空间上的向量, 物品相似度的计算就是通过向量间的余弦夹角来衡量的, 夹角越大, 物品的相似度就越低。

(3) 改进的余弦相似度。

$$\text{sim}(i, j) = \frac{\sum_{n \in N_{ij}} (R_{i,n} - \bar{R}_n)(R_{j,n} - \bar{R}_n)}{\sqrt{\sum_{n \in N_i} (R_{i,n} - \bar{R}_n)^2} \sqrt{\sum_{n \in N_j} (R_{j,n} - \bar{R}_n)^2}} \quad (4)$$

其中,  $\text{sim}(i, j)$  表示物品  $i$  和物品  $j$  的相似性;  $N_i$  和  $N_j$  分别表示物品  $i$  和物品  $j$  的评分用户集合;  $\bar{R}_n$  表示用户  $n$  的平均评分。

这样改进的目的是标准化评分, 以防不同用户的评分习惯不同, 比如一些用户打分均分偏高, 可能会影响相似度的计算<sup>[10-12]</sup>。

### 2.1.2 推荐生成过程

计算出物品之间的相似度之后, 基于物品的协同过滤推荐算法将根据用户的高分物品, 寻找其高度相似的物品, 从而得到 TOP-N 推荐结果集<sup>[13-14]</sup>。

首先, 算法需要构建一个用户-物品的倒排列表, 即列出每个用户高度喜欢的物品。然后根据喜爱的物品与物品列表中物品的相似度计算得到最终推荐分数, 即预测兴趣度, 公式如下<sup>[15]</sup>:

$$H_{ui} = \sum_{i \in N(u) \cap S(j, K)} \text{sim}(i, j) R_{ui} \quad (5)$$

其中,  $N(u)$  是用户喜欢的物品集合;  $S(j, K)$  是与物品  $j$  最相似的  $K$  个物品集合。

根本意义是与用户的历史喜欢物品越相似, 越可能是用户的潜在喜欢物品<sup>[15-16]</sup>。

## 2.2 关键字提取算法与文本处理

### 2.2.1 关键字提取算法

TF-IDF 是一种常见的用于信息检索和数据挖掘的加权技术, 也是很好理解的关键字提取算法。本质上讲, TF-IDF 就是通过特定文档中词的相对频率和整个文档语料库中该词的反比例进行比较, 即其计算的就是某单词在特定文档中的相关系数。TF-IDF 的公式如下:

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \quad (6)$$

其中,  $D$  为给定的已经整理的文档集合,  $|D|$  是语料库的大小;  $w$  为特定的单词, 另外还有文档  $d \in D$ ;  $f_{w,d}$  为  $w$  出现在  $d$  中的次数;  $f_{w,D}$  为  $D$  中包含  $w$  的文档数。

根据  $f_{w,D}$  和  $f_{w,d}$  权值的不同, 最终  $w_d$  的结果不尽相同, 因此文中的不同  $w_d$  的词汇带来不同的意义。

TF-IDF 算法是建立于这样一个假设: 如果一个单词能够很有效地区分出这篇文章与其他文章, 那么这个单词应该在本文中高频次的出现, 而在整个文档集中的其他文档中出现较少。所以 TF 词频是一个很好地衡量是否是同类文本的权值<sup>[17]</sup>。另外, 一个单词如果其出现的文本频率越低, 它区别不同类别文本的能力也会变得越强, 所以该算法引入了 IDF 逆文本频率的概念<sup>[18]</sup>。

但是, 传统的 TF-IDF 仍然存在各种各样的不足。根据 IDF 的公式, 某些特征词的 IDF 值会很低, 意味着该特征词可能不具有代表性。但是在实际情况中, 如果某一个词汇在某一类文本中大规模的出现, 则说明该词汇能够很好地表示出该类文本的内容取向, 像这样的词汇应该给予其很高的权重值, 用来计算文本之间的相似性, 从来推荐相似特征的电影。所以传统 TF-IDF 公式会出现两个缺陷: 部分不能代表文本内容的低频词汇, 其 IDF 值可能相对很高; 一些真正能够代表文本内容的关键词的 IDF 值却非常低。所以要对传统 TF-IDF 公式进行一定的改进<sup>[19-20]</sup>。

首先, 需要引入归一化的思想。因为根据原公式, 文本的长短会对 TF-IDF 的最终结果产生影响, 所以要对其权值进行归一化处理, 使得最终结果大于等于 0 且小于等于 1, 公式如下:

$$W_d = \frac{f_{w,d} * \log(|D|/f_{w,D} + 0.01)}{\sqrt{\sum_{d=1}^D (f_{w,d})^2 * [\log(|D|/f_{w,D} + 0.01)]^2}} \quad (7)$$

其次, 一个词在全文的跨度也可以在一定程度上表现出该词的重要性。跨的句子越多, 在全文的代表性就越高。传统 TF-IDF 的权值计算过程中, 局部关键词很有可能因为在局部的高频率出现而降低了提取关键词的准确性, 所以可以进一步改进 TF-IDF 公式:

$$W_d = \frac{f_{w,d} * \log(|D|/f_{w,D} + 0.01) * \frac{l_d}{L}}{\sqrt{\sum_{d=1}^D (f_{w,d})^2 * [\log(|D|/f_{w,D} + 0.01)]^2 * \frac{l_d}{L}}} \quad (8)$$

其中,  $l_i$  表示句子出现的句子数;  $L$  表示句子总数。



### 2.2.2 文本处理

#### (1) 停止词。

停止词(stop words)就是一些在某种语言中大范围使用的词,但是对分析文本并没有太大意义,比如说“我们”“可能”“所以”等。所以要对这些停止词进行一些处理,但面对不同类型的需求的时候,对停止词的处理方法是不尽相同的。比如聚类算法中就可能要减少停止词的权值,信息检索的时候就不会检索停止词,而文中需要直接删除停止词,以防某些停止词的出现干扰文本 TF-IDF 算法的计算结果。同时停止词的过滤可以降低系统处理语句的量,减少非内容信息词的干扰。当然停止词的使用必须谨慎,以免丢失关键的文本信息。

#### (2) 同义词。

英文和中文一样,会包含大量的同义词,而推荐系统计算相似性时,不同的单词可能会被理解成不同的意义,从而增加了维度,但是因为是同义词,它们应当代表的是同一维度。因为其本身含有的意义应该是相同的或者相近的,所以为了提高推荐系统的精确性和计算效率,必须将同义词进行精确的替换,替换为某一个指定的词。

## 3 实验

### 3.1 实验数据

实验采用著名的 MovieLens 电影数据集。该数据集是由明尼苏达大学 GroupLens 项目组创办的一个开源的站点,包含了很多开源数据集,文中使用其中的 MovieLens 1M Dataset,包含了“评分”、“用户”和“电影”四张表(英文)。评分表包括用户 ID,电影 ID,对应评分以及时间戳。用户表包括用户 ID,性别,年龄段,职业以及邮编。电影表是电影 ID,电影名与年代,以及电影分类(例如喜剧,浪漫等)。

为突出该实验方案解决稀疏矩阵相似性计算问题的能力,进行了人工预处理,去掉了部分评分,使得评分矩阵的某些维度更加稀疏。预处理后共有 3 883 部电影,80 万个评分数据以及 6 040 个用户。

另外实验采用的影评数据集采集自著名影评网站 IMDB(Internet movie database,互联网电影资料库),其创建于 1990 年,是一个专业且严肃的影评网站。根据 MovieLens 的电影名从各自的页面中爬取影评作为 TF-IDF 的数据集。

### 3.2 实验模型设计

为了验证 TF-IDF 在改进协同过滤算法的可行性,进行了几组对比实验,首先是传统的协同过滤算法实验。首先计算物品之间的相似度,然后根据物品之间的相似度和用户的历史打分物品给出最终的推荐列

表。其次就是使用 TF-IDF 的改进公式来改进传统协同过滤算法(CF)中“稀疏矩阵”的相似性计算(简称 TI-CF 算法)。该实验先使用 TF-IDF 算法抽取权值前 10 的关键词来计算物品相似度,然后取出 TOP-N 的物品,和传统相似度计算方法的 TOP-N 物品根据相似度大小(已经归一化)排名取合集 TOP-N,作为最后的推荐物品集合。

### 3.3 实验评价体系

文中使用准确率、召回率以及覆盖率评价该模型的推荐性能。设给用户  $u$  推荐  $N$  个物品的集合为  $R(u)$ ,令用户  $u$  在测试集上评分的物品集合为  $T(u)$ ,则准确率和召回率的公式为:

$$\text{Precision} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|} \quad (9)$$

$$\text{Recall} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|} \quad (10)$$

准确率反映了正确被评分的项目占推荐项目的比例,召回率反映了正确被评分的物品占用户实际喜欢的物品的比例。两者取值在 0 和 1 之间,数值越接近 1,准确率和覆盖率就越高。

覆盖率反映了推荐算法挖掘长尾的能力。长尾效应指的是这样一种现象:数量占少数的热门商品,往往贡献了网站的大部分流量。其最初由“连线”的总编辑克里斯·安德森(Chris Anderson)于 2004 年发表,主要表达诸如 Amazon 和 Netflix 此类的商业网站的盈利途径。其强调那些销量小而且及其庞大的商品能够给公司带来的收益,往往大大超过那些所谓的热门商品。如果推荐系统片面地推荐了热门商品,那么这个系统并没有分析出有意义的价值,真正的价值应该是发现冷门的但是很有商业潜力的商品。

$$\text{Coverage} = \frac{|\bigcup_{u \in U} R(u)|}{|I|} \quad (11)$$

### 3.4 实验结果

如图 1~3 所示,物品的相似性计算过程经过关键词提取技术的改进后,最终推荐结果在各种 TOP-N 取值下,其准确率、召回率和覆盖率均有一定的提升。

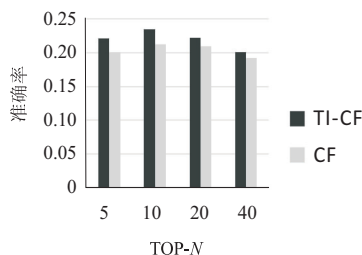


图1 准确率

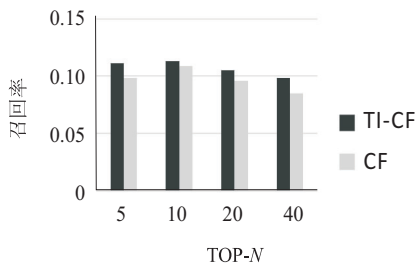


图 2 召回率

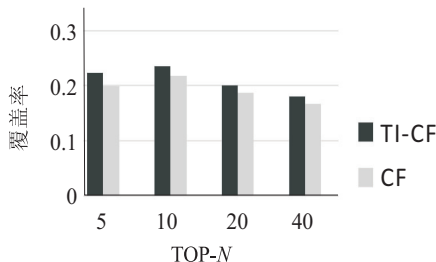


图 3 覆盖率

证明通过文本提取关键字的路径,可以成功挖掘到文本相关电影的内容特征,进而辅助相似度计算的过程,最终在一定程度上解决矩阵稀疏的问题,提升推荐系统用户体验,从而给电影推荐网站带来盈利。

## 4 结束语

传统协同过滤算法发展至今,已经有了很大的成就。但是面对冷启动以及数据稀疏问题时,精确率等评价标准会显著下降,这时就需要去挖掘其他信息来完善物品相似度的计算,各种电影的影评就是来源之一。如果是一个创业期电影网站,面对冷启动以及矩阵稀疏问题时,就可以从其电影的影评等文字数据中获得有效的信息,从而准确计算电影相似度,推荐相似类型电影。

TF-IDF 算法仍然有很多挖掘的潜力,深度学习在推荐系统领域的作用也正在逐渐展现<sup>[19]</sup>,值得进一步去研究。

### 参考文献:

- [1] 项 亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012: 51-58.
- [2] 张朝恒,何小卫,陈勇兵. 基于社交网络信息的协同过滤推荐算法[J]. 计算机技术与发展,2017,27(12):28-34.

- [3] 陈小礼. 基于最大团的协同过滤算法的研究与改进[D]. 武汉:武汉邮电科学研究院,2018.
- [4] 张应辉,司彩霞. 基于用户偏好和项目特征的协同过滤推荐算法[J]. 计算机技术与发展,2017,27(1):16-19.
- [5] 许征征. 个性化推荐系统中基于用户的协同过滤算法与系统架构的研究与优化[D]. 济南:山东大学,2017.
- [6] 吕成成. 基于用户项目属性偏好的协同过滤推荐算法[J]. 计算机技术与发展,2018,28(4):152-156,160.
- [7] 李 玲,王移芝. 融合信息熵和加权相似度的协同过滤算法研究[J]. 计算机技术与发展,2018,28(5):23-26,31.
- [8] 李 民. 基于智慧推荐的高校智慧图书馆服务模式研究[D]. 天津:天津理工大学,2017.
- [9] 刘 涛,刘 佐. 一种面向新文章的个性化推荐算法研究[J]. 控制工程,2018,25(6):999-1006.
- [10] 黄震华,张佳雯,田春岐,等. 基于排序学习的推荐算法研究综述[J]. 软件学报,2016,27(3):691-713.
- [11] 谭 昶,刘 洪,吴 乐,等. 推荐系统中典型用户群组的发现和应用[J]. 模式识别与人工智能,2015,28(5):462-471.
- [12] 李 慧. 社会网络环境下的个性化推荐算法研究[D]. 徐州:中国矿业大学,2016.
- [13] RICCI F, ROKACH L, SHAPIRA B. Recommender systems: introduction and challenges [M]//Recommender systems handbook. Boston, MA: Springer, 2015.
- [14] 刘文佳,张 骏. 改进的协同过滤算法在电影推荐系统中的应用[J]. 现代商贸工业,2018(17):59-62.
- [15] MANOGARAN G, VARATHARAJAN R, PRIYAN M K. Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system [J]. Multimedia Tools and Applications, 2018,77(4):4379-4399.
- [16] WANG Zhibo, LIAO Jilong, CAO Qing, et al. Friendbook: a semantic-based friend recommendation system for social networks[J]. IEEE Transactions on Mobile Computing, 2015, 14(3):538-551.
- [17] 石俊涛. 中文文本分类中卡方特征提取和对 TF-IDF 权重改进[D]. 成都:西华大学,2017.
- [18] 李 原. 中文文本分类中分词和特征选择方法研究[D]. 长春:吉林大学,2011.
- [19] 杨文龙. 基于 BP 神经网络的协同过滤推荐算法的研究与应用[D]. 武汉:武汉邮电科学研究院,2018.
- [20] 刘小慧,李长玲,冯志刚. 基于改进的 TF \* IDF 方法分析学科研究热点——以情报学为例[J]. 情报科学,2017,35(7):82-87.