

# 电影推荐系统中基于图的协同过滤算法

郑 策<sup>1 2</sup>, 尤佳莉<sup>1 2</sup>

(1. 中国科学院声学研究所国家网络新媒体工程技术研究中心, 北京 100190; 2. 中国科学院大学, 北京 100190)

**摘要:** 在视频服务领域, 通常使用传统的协同过滤算法来解决评分数据较为稀疏的问题, 而算法的视频相似度计算仅利用评分矩阵, 从而造成推荐准确度较低。针对视频资源中的电影这一应用场景提出一种基于图的协同过滤算法。结合电影属性与用户偏好的关联性, 将电影信息中类型、导演和演员等信息进行图元素的映射, 融合图结构特点来计算影片资源的相似度。用该方法替代传统协同过滤算法中仅利用评分矩阵的相似度计算方法, 在一定程度上缓解了由于数据稀疏性影响推荐准确度的问题。实验验证了该方法的有效性。

**关键词:** 关联性分析; 协同过滤算法; 图结构; 个性化推荐

中图分类号: TP391

文献标识码: A

doi: 10.3969/j.issn.1006-2475.2019.11.008

## A Graph-based Collaborative Filtering Algorithm in Movie Recommendation System

ZHENG Ce<sup>1 2</sup>, YOU Jia-Li<sup>1 2</sup>

(1. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Science, Beijing 100190, China; 2. University of Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** In order to solve the problem of sparse scoring data in the field of video service, the traditional collaborative filtering algorithm is usually used, but the video similarity calculation of the algorithm only uses score matrix, which results in low recommendation accuracy. In this paper, a graph based collaborative filtering algorithm is proposed for the scene of the movie in video resources. Combining the correlation between movie attributes and user preferences, the map elements of film information such as types, directors and actors are mapped, and the similarity of film resources is calculated by combining the features of graph structure. This method replaces the similarity calculation method of scoring matrix in traditional collaborative filtering algorithm, which alleviates the problem that the recommendation accuracy is affected by the sparse data. Experiment verifies the effectiveness of the proposed algorithm.

**Key words:** correlation analysis; collaborative filtering algorithm; graph structure; personalized recommendation

## 0 引 言

当前互联网信息越来越多, 呈现指数增长的趋势。视频服务网站是互联网的重要组成部分, 往往都存着数以万计的电影资源<sup>[1]</sup>, 用户打开电影网站时也许没有明确的目标, 使得查找时所涉及的电影资源数量仍然巨大, 并且很难轻松获得符合自己兴趣的新资源。在这种情况下, 通过对用户行为信息和电影资源信息进行关联性分析来预测并推荐与用户喜好相符的电影, 会很大程度上增加用户的观看兴趣。视频服务已经成为用户浏览互联网时最关注的服务之一, 因而, 提升在线视频服务的用户体验具有重要的研究

价值<sup>[2]</sup>。而视频服务中的电影推荐系统可以通过提高推荐算法的准确度来增加用户的点击率、购买率或观看时间等, 从而也给视频服务商带来更多的经济收益。但是传统协同过滤技术仅使用用户对物品的评分矩阵, 没有充分利用用户和电影资源的其他多种特征, 而且由于评分矩阵非常稀疏, 导致推荐系统的推荐准确率严重下降<sup>[3]</sup>。

关联性分析是一种简单并且实用的分析方法, 是通过发现大量数据中存在的某种关联性或相关性, 从而可以描述事物中某一些属性同时出现的规律或者模式。关联性分析方法已经在很多领域应用, 例如购物篮分析、发现毒蘑菇的相似特点以及 Twitter 源记

收稿日期: 2019-03-13; 修回日期: 2019-03-25

基金项目: 中国科学院先导专项课题(XDC02010701); 中国科学院青年创新促进会项目(Y529111601)

作者简介: 郑策(1994-), 女, 河南南阳人, 硕士研究生, 研究方向: 大数据分析, 基于图计算的视频服务分析, E-mail: zhengce@dsp.ac.cn; 尤佳莉(1982-), 女, 副研究员, 博士, 研究方向: 网络边缘智能, 大数据分析和未来网络技术, E-mail: youjl@dsp.ac.cn。

录公共词等<sup>[4-5]</sup>。本文通过将关联性分析应用到视频服务领域,针对现有协同过滤算法由于评分数据的稀疏性使得推荐算法准确度较低的问题,结合图结构特点,通过改进电影资源相似度计算来进一步提升推荐算法准确度的研究。

## 1 相关推荐算法的介绍

内容推荐问题已经被研究了很多年,常见的推荐算法主要有:基于热度的算法、基于协同过滤的算法、基于模型的算法、基于标签的算法<sup>[6-7]</sup>以及混合型算法。这几种推荐算法的具体描述如下:

### 1) 基于热度的算法<sup>[8]</sup>。

基于热度的算法是按照某种热度,根据日均页面浏览量、独立访客或者转发率等数据对用户进行推荐<sup>[9]</sup>。这种算法通常适合新注册的用户,例如把热榜上的影片资源推荐给影迷。但是,它无法提供针对性的个性化推荐给相应的用户。

### 2) 基于协同过滤的算法。

协同过滤(Collaborative Filtering, CF)算法是推荐系统中最为经典的算法,包括基于用户的协同过滤算法(User-CF)和基于项目的协同过滤算法(Item-CF)<sup>[10]</sup>。协同过滤算法可以通过获得用户的观看行为数据作为隐性反馈或者评分记录的数据作为显性反馈,利用一定的算法预测出用户没有观看过但可能喜欢的电影并达到推荐给用户观看的目的<sup>[11]</sup>。

协同过滤算法利用了用户和商品的交互信息,即用户的行为记录。通常,用户对商品的行为反馈可以分为显性反馈和隐性反馈,显性反馈涉及用户个人隐私难以获得以及实际操作中这类信息较为稀疏,因此这也成为了算法的瓶颈之一<sup>[12]</sup>。

### 3) 基于模型的算法。

基于模型的协同过滤算法一般使用的是机器学习的方式<sup>[13]</sup>,通过样本中用户的喜好信息来训练得到推荐模型,并根据实时的用户喜好信息进行预测和推荐,常用的方法包括LSI<sup>[14]</sup>、贝叶斯网络等<sup>[15]</sup>。

### 4) 基于标签的算法。

基于标签的推荐算法也较为常见,比如豆瓣、京东等平台的评论信息<sup>[16]</sup>。该算法所需的标签信息的主体分为普通用户和专家学者,同时需要获得大量用户的显性反馈行为<sup>[17]</sup>,因此在数据准备阶段相对繁琐。

### 5) 混合型的算法。

在现实应用中,其实很少直接用某种单一的算法来做推荐系统<sup>[18]</sup>。该算法是将多种推荐技术进行混合来弥补各算法的缺点<sup>[19]</sup>,综合各算法的优点以达

到更好的推荐效果。最常见的是将协同过滤算法和其他技术相结合,以克服冷启动<sup>[20]</sup>以及数据稀疏性<sup>[21]</sup>等问题。

通过对以上常见的几种推荐算法的优缺点分析,可以发现对于目前广泛存在的关系型数据,这些算法基本都没有深入挖掘大规模关系型数据背后所隐含的关联信息。本文将传统协同过滤算法和关联性分析技术相结合,针对视频服务领域中存在大量关系型数据但这类数据中有效信息较为稀疏,造成了各类推荐算法对于关系型数据的推荐质量较低的问题,提出改进算法——基于图的协同过滤推荐算法。

## 2 基于图的协同过滤推荐算法

### 2.1 问题的提出

传统的基于用户的协同过滤算法是根据评分矩阵建立用户的偏好模型来帮助用户发掘新的可能感兴趣的电影类型,其主要处理过程如图1所示。

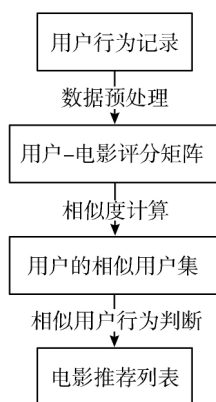


图1 传统的基于用户的协同过滤算法流程

由于传统协同过滤算法对属性关系的刻画较少,特别是针对关系型数据,难以描述复杂属性潜在的依赖和相关性。在此情况下,可以发现图结构可以很好地表示关系型数据:图由顶点和边组成,表示为图 $G=(V, E)$ ,其中 $V$ 表示顶点的集合, $E$ 表示边的集合,边可以代表顶点数据之间的关系。对于视频网站上的影片信息,如导演、演员等信息可以表示为图中的顶点,它们之间的关系可以表示为图中的边。因此,本文提出一个基于图的协同过滤算法。在该方法中,电影数据以图的形式存储,根据用户的属性信息建立用户兴趣模型,并利用顶点和边的关系多维度来挖掘出电影属性与用户偏好之间存在的关联性,更加准确地获得用户的兴趣特征,最终得到每个用户当前兴趣的个性化推荐列表,该算法具体处理过程如图2所示。

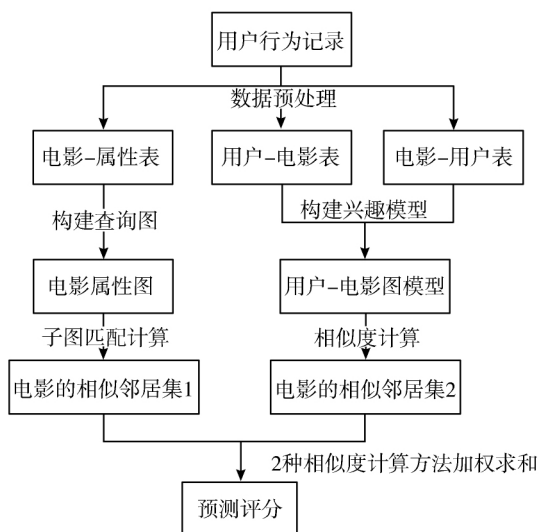


图2 基于图的协同过滤算法流程图

在这里,用户兴趣模型的构建是对用户间,以及用户和电影内容间关系的刻画和打分,也是推荐算法性能优劣的关键。因此,本文提出一种融合关联性分析的用户兴趣模型构建方法(见2.2节),通过图结构构建用户属性和内容属性间的关联关系,以及用户和内容间的兴趣强弱关系;同时,基于图结构采用子图匹配方法计算电影资源间的相似关系,作为电影内容间相似性的衡量;最后通过2个模型的融合,更加深入地挖掘电影属性与用户之间的隐含关联性,从而进行电影内容的推荐(见2.3节)。

## 2.2 融合关联性分析构建用户兴趣模型

用户兴趣模型的构建主要用于预测用户的兴趣偏好或行为倾向,融合关联性分析建立用户兴趣模型是基于图的协同过滤算法的关键,其主要的构建步骤包括:

1) 从数据库中获得电影的类型、上映时间、导演、演员这4类属性信息。

2) 将电影的4类属性信息通过归一化处理提取有效且统一的信息作为特征信息。

3) 统计出存在用户行为记录的电影资源中4类属性值出现的次数,挖掘用户喜好。表1为数据库中电影资源的演员属性记录示例,其中actor代表演员属性, $N$ 代表电影资源中该演员出现的次数, $N$ 值越大,则表示该值所对应的演员越符合该用户的偏好。

表1 演员属性记录( $k \in Z$ )

actor	$a_1$	$a_2$	...	$a_k$
$N$	$n_1$	$n_2$	...	$n_k$

4) 选择出现不小于次数为 $N$ 的 $M$ 个电影资源的属性进行组合,在本文中即表示在用户行为记录中出现次数为20的电影类型、上映时间、导演、演员4个

属性( $N=20, M=4$ )进行两两组合,得到用户对电影内容各属性组合的兴趣记录。

5) 通过步骤4中获得的用户兴趣记录得到各组合属性的兴趣向量。其中,本文认为当 $N$ 大于某一值时才能由兴趣记录放入兴趣向量中,并将该条用户兴趣记录标记为有效。用户兴趣向量中每一个值均对应一条有效的兴趣记录。设用户兴趣向量为 $v_1, v_2, \dots, v_i$ ,那么用户兴趣模型可表示为 $V = \{v_1, v_2, \dots, v_i\}, i \in Z$ 。

$$\begin{cases} v_1 = (t_{11}, t_{12}, \dots, t_{1K_1}) \\ v_2 = (t_{21}, t_{22}, \dots, t_{2K_2}) \\ \vdots \\ v_i = (t_{i1}, t_{i2}, \dots, t_{iK_i}) \end{cases} \quad (1)$$

式(1)中 $i$ 表示兴趣向量的个数; $K$ 表示选取的 $M$ 个电影属性的组合数,本文中 $M=4$ ,则 $K=6$ 。

用户兴趣模型的构建过程中需要提前设定 $N$ 值和 $M$ 值,然后根据实验的数据集规模进行调整。

## 2.3 基于图结构特点进行电影资源的相似度计算

一般地,传统的协同过滤算法采用余弦相似度等传统的相似度计算方法来计算用户或者项目的相似度,但是传统的相似度计算方法还有待改进<sup>[22-23]</sup>,目前主要面临的问题有:通常计算项目之间的相似性是通过项目所获得的评分向量计算得到的,而实际上有评分的项目只是很少的一部分,这导致了历史记录里评分数据的极端稀疏性<sup>[24]</sup>,影响推荐准确度。为了提高电影推荐系统的推荐准确度,本文提出的基于图的协同过滤算法中利用图结构特点,将用户和电影资源作为图的顶点,用顶点之间的边来表示关系。利用顶点间的路径信息以及根据电影的属性图的子图匹配方法计算项目之间的相似度,加权求和得到最终的相似度排名并预测用户对项目的评分。

其中,基于相似度计算的推荐算法步骤如下:

1) 采用图结构来记录信息并计算电影资源之间的相似度,得到电影的相似邻居集1,其中图模型中每条边所连接的2个顶点为类型互斥;而对于图模型中2个图顶点的相似度计算取决于:连接2个图顶点的路径数目、图顶点之间路径的长度以及连接图顶点的边数。

在计算电影资源之间的相似度时,连接2个图顶点的路径数越多,关联性越强则代表相似度越高;路径越短,关联性越强则代表相似度越高;对于连接边数较多的图顶点,对相似度计算贡献不大,需要适当降低其权重。

综上所述,设当代表电影的图顶点之间经过的非电影的图顶点的个数为 $n(n \in Z)$ ,那么基于图的协

同过滤推荐算法的电影资源的相似度计算公式可表示为:

$n = 1$  时:

$$\text{sim}(m_i, m_j) = s \times \sum_{path=1}^k (1/|u_{path}|) \quad (2)$$

$n > 1$  时:

$$\text{sim}(m_i, m_j) = s \times \sum_{path=1}^k (|m_{path}| / (|u_{1path}| + |u_{2path}| + \dots + |u_{npath}|)) \quad (3)$$

本文所述电影图顶点指顶点标签为电影,电影数据的图模型中标签为 Movie 的图顶点记为电影图顶点;非电影图顶点则表示除了这类图顶点以外的其他类型的图顶点。

上述公式(2)和公式(3)中  $k$  表示  $u_i$  和  $u_j$  之间共有  $k$  条经过非电影图顶点的路径,  $|m_{path}|$  表示第  $path$  条路径上经过的电影图顶点的度数之和,即通过 2 个待计算相似度的电影图顶点之间通过电影图顶点连接的边数之和,连接的边数越多 2 个电影的关联性越强,则认为相似性越高;  $|u_{npath}|$  表示第  $path$  条路径上的第  $n$  个非电影图顶点的度,公式(3)中  $|u_{1path}| + |u_{2path}| + \dots + |u_{npath}|$  表示代表 2 个电影图顶点之间经过的非电影图顶点的度数之和,因为 2 个图顶点之间非电影图顶点的度数之和越大,则经过的图顶点越多即表示路径越长,那么 2 个电影的相似度则越低。 $s = \log(|V|)$ ,  $|V|$  表示 2 个电影图顶点之间的非电影的图顶点集合的大小,2 个电影图顶点之间相连接的非电影图顶点集合越大,表示它们相似度越高。

2) 构建电影资源属性图模型,将电影类型、上映时间、导演、演员与电影的关系映射为图结构中图顶点和边的关系,用步骤 1 中的相似度计算方式计算得到电影资源的相似邻居集 1,相似度表示为  $\text{sim}_1(m_i, m_j)$ 。

3) 根据 2.2 节构建的用户兴趣模型,建立用户-电影关系图模型,将用户对电影的评分关系映射为图结构中图顶点和边的关系,将其加入到图顶点集合和边集合中,利用步骤 1 中的方法计算得到用户评分较高的相似电影列表,从而得到电影资源的相似邻居集 2,相似度表示为  $\text{sim}_2(m_i, m_j)$ 。

将 2 个相似度加权求和,并根据计算所得的电影相似度得分,最终得到推荐列表。

$$P = \text{sim}_1(m_i, m_j) \times (1 - \eta) + \text{sim}_2(m_i, m_j) \times \eta \quad (4)$$

其中  $\text{sim}_1(m_i, m_j)$  和  $\text{sim}_2(m_i, m_j)$  分别代表电影属性图 and 用户-电影关系图中的相似度计算结果,  $\eta$  代表权重,  $P$  表示每部电影资源最终的相似度得分。本文对共计 11 种不同组合  $(1 - \eta, \eta)$  进行测试,取值分别为  $(0, 1)$ ,  $(0.1, 0.9)$ ,  $(0.2, 0.8)$ ,  $\dots$ ,  $(1, 0)$ , 结果表明当取值为  $(0.8, 0.2)$  时,推荐准确度最高,故在系

统实现中  $\eta$  取值为 0.2,此时推荐算法的推荐准确度提升最大。

通过公式(2)~公式(4)可以看出,本文所提出的电影资源的相似度计算方法在原来只考虑用户评分的基础上又参考了电影资源本身的属性信息之间的相似性,这使得评分数据存在稀疏性的问题得到了一定的缓解,同时还有利于提高相似度计算的有效性。

其中,基于图结构的电影相似度  $\text{sim}_1(m_i, m_j)$  计算部分的伪代码如图 3 所示。

#### Algorithm 1 Movie Similarity Calculation

**Input:** Graph vertex ID of movie  $m_i$  and  $m_j$

**Output:**  $\text{sim}(m_i, m_j)$

```

1: function GETSCORE(int movie_1, int movie_2)
2:   pathList ← findPath(movie_1, movie_2)
3:   for path ∈ pathList do
4:     nodeList = getMiddleNode(path);
5:     for node ∈ nodeList do
6:       m_degree+ = movieNode.getDegree();
7:       u_degree+ = notMovieNode.getDegree();
8:     end for
9:     if n == 1 then
10:      similarity+ = 1/u_degree;
11:     else
12:       if n > 1 then
13:         similarity+ = m_degree * 1/u_degree;
14:       end if
15:     end if
16:   end for
17:   return similarity * s;
18: end function

```

图 3 电影相似度  $\text{sim}_1(m_i, m_j)$  计算部分的伪代码

## 3 实验结果与分析

### 3.1 实验数据准备

本文以视频资源中的电影作为实验对象。为验证基于图的协同过滤算法的推荐质量,实验采用推荐算法领域的经典标准数据集——MovieLens 公开电影数据集,该数据集包含约 100000 条用户对电影的评分记录和每部电影的类型、相关导演、演员等信息,一共包含约 1000 个用户以及 2000 部电影资源。对数据集中的数据进行预处理,过滤掉其中电影信息不完整的记录,即不同时包括电影类型(Genre)、上映年份(Year)、导演(Director)以及演员(Actor)这 4 类信息;删除年代久远的电影信息的记录,剩余约 20000 条记录用于实验。

在基于图的协同过滤算法中以图的形式表示电影数据。其中,图模型的构建是基于 Neo4j 图数据库来实现,该图数据库可以将类似于文中的用户-电影、电影-属性、用户-属性这类关系型数据转换成图顶点和边的形式存储;同时,它自带几种简单的图算法,支持相关的图算法的操作,可以将本文提出的基于图的协同过滤算法结合 Neo4j 图数据库来完成算法中涉及图算法的操作。实验数据集中电影数据的图模型示例如图 4 所示。

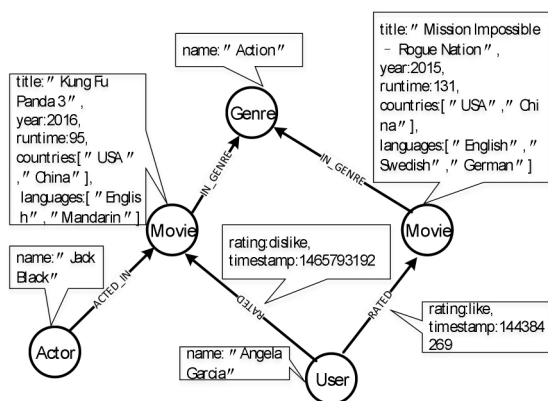


图4 电影数据的图模型(组合:类型+演员)

特别地,由于本文所选的实验数据集中没有用户的性别、年龄等属性信息,所以图4所示的图模型中未出现这类信息,但是下一步的研究是要完善用户的这类属性信息,也作为图顶点来加入图模型中,以丰富算法的输入信息。

### 3.2 电影推荐系统的评价指标

本文选择推荐列表准确度、召回率以及这两者的综合评价指标作为电影推荐实验结果的评价指标。这类指标的推荐结果是列表的形式,分别如公式(5)~公式(7)所示,并忽略具体评分的场景。通过对用户的历史行为记录的分析得到个性化的电影列表,进而通过测试集来判断用户对于推荐的项目是否感兴趣。

准确度:

$$P_{ui} = |rec(u_i) \cap real(u_i)| / |rec(u_i)| \quad (5)$$

召回率:

$$R_{ui} = |rec(u_i) \cap real(u_i)| / |real(u_i)| \quad (6)$$

综合评价指标:

$$F1-Measure = (2 \times P_{ui} \times R_{ui}) / (P_{ui} + R_{ui}) \quad (7)$$

其中  $rec(u_i)$  表示通过推荐算法向用户  $u_i$  推荐的电影列表,  $real(u_i)$  代表在测试集上用户  $u_i$  实际喜欢的电影列表。

### 3.3 算法性能比较与分析

本文在 MovieLens 电影数据集上进行基于图的协同过滤推荐算法和传统的协同过滤算法的对比实验。在电影数据集中随机抽取 80% 的数据作为实验的训练集  $S$ , 剩下 20% 的数据作为测试集  $T$ ; 将  $S$  的数据作为历史记录, 采用 2 种算法分别对  $T$  中用户对影片的喜好进行预测并生成推荐列表 Top- $k$ , 这里  $k$  取 10。

实验中, 分别取电影属性信息: 电影类型、上映年份、导演以及演员两两组合, 即类型+年份、类型+导演、类型+演员、年份+导演、年份+演员以及导演+演员共 6 种情况。其中, 实验的基准算法具体为本文 2.1 节所介绍的传统的基于用户的协同过滤算法, 该

算法在计算相似度时不考虑电影的属性信息, 只考虑用户的评分情况 (0~10 分), 本文将评分小于 5 的记为不喜欢, 大于或等于 5 的记为喜欢, 其输入参数仅为用户对影片的评分矩阵; 影片相似度则通过余弦相似度计算方法来计算。设用户对影片  $a$  的评分矩阵为  $[x_1, y_1]$ , 对影片  $b$  的评分矩阵为  $[x_2, y_2]$ , 则该方法的相似度计算如公式 (8):

$$\cos \theta = (x_1 \times x_2 + y_1 \times y_2) / (\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}) \quad (8)$$

实验过程中, 上述 2 种对比算法分别针对所选取的每种电影属性的组合各自重复实验 5 次, 记录实验数据并分别计算推荐列表中的准确率  $P_{ui}$ 、召回率  $R_{ui}$  以及综合评价指标  $F1-Measure$ , 并求得 5 次实验的平均值, 如图 5~图 7 所示。

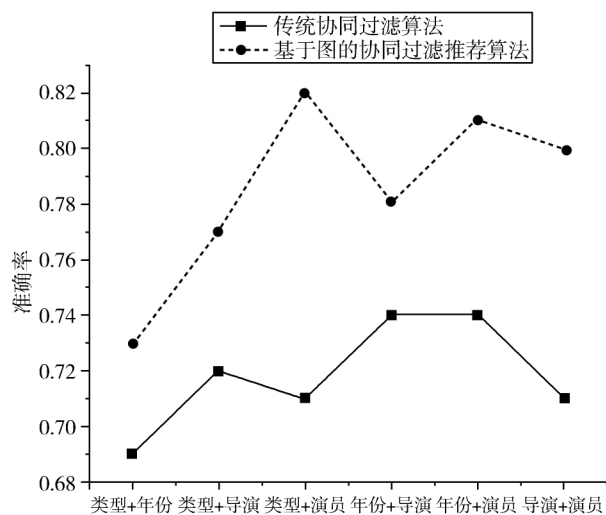
图5 算法的准确率  $P_{ui}$  对比

图5表示传统协同过滤算法和本文提出的基于图的协同过滤算法在电影数据集中存在大量关系型数据的情况下, 对于推荐算法的准确度的性能比较。从图5中可以看出本文提出的算法相较于传统的协同过滤算法而言, 推荐准确率在一定程度上有所提升, 平均提升约为 9.49%; 针对基于图的协同过滤算法来说, 当算法中的电影属性组合包含演员时, 相对其他组合类型即包含导演、包含类型以及包含年份而言推荐准确率更高, 分别高出 10.36%、8.31% 以及 12.65%。

图6表示上述 2 种推荐算法的召回率的性能比较。从图6中可以看出本文提出的算法相较于传统的协同过滤算法而言召回率也有所提升, 平均提升约为 11%; 针对基于图的协同过滤算法来说, 当算法中的电影属性组合为导演+演员时, 相对实验中其他组合类型, 即类型+年份、类型+导演、类型+演员、年份+导演和年份+演员这些组合而言召回率更高。分别高出 10.43%、6.12%、9.97%、10.69% 以及 3.92%。

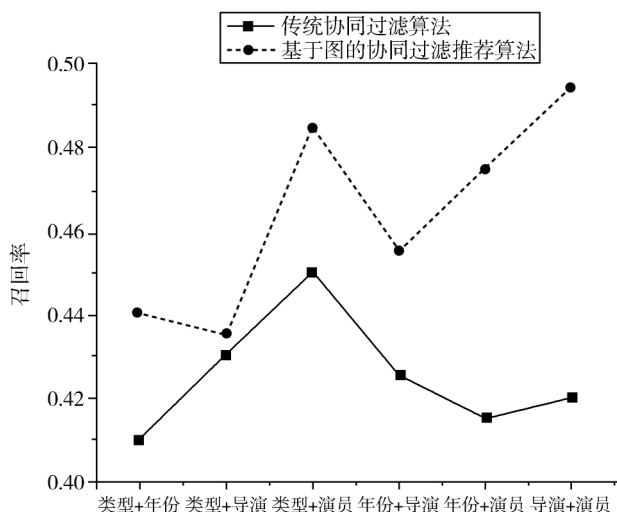
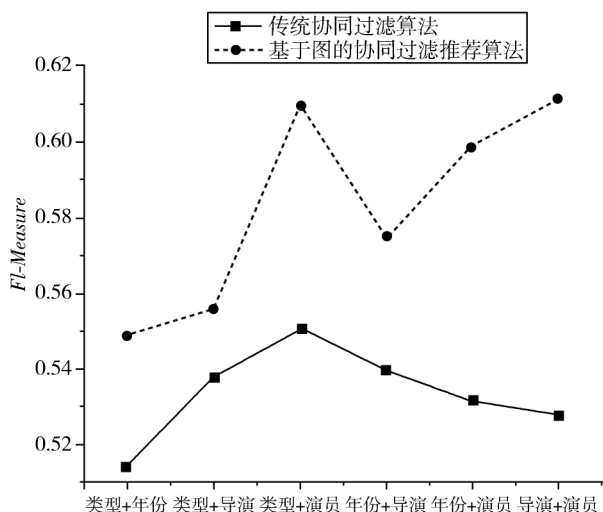
图6 算法的召回率  $R_m$  对比图7 算法的综合评价指标  $F1-Measure$  对比

图7表示上述2种推荐算法的综合评价指标的比较。从图7中可以看出,本文提出的算法相较于传统的协同过滤算法而言总体有所提升,平均提升约为9.26%;当算法中的电影属性组合为导演+演员时,相对实验中其他组合类型,即类型+年份、类型+导演、类型+演员、年份+导演和年份+演员这些组合而言提升更高,分别高出9.13%、12.63%、5.23%、9.43%以及3.27%。

综上所述,本文提出的基于图的协同过滤算法相对于传统的协同过滤算法而言,在电影信息的内容推荐场景下,算法的推荐质量整体有所提升,并当电影属性图模型选择的组合包含演员属性以及其中组合为导演+演员时推荐质量提升最大。

## 4 结束语

推荐系统数据库的评分数据稀疏,对电影推荐的质量有所限制<sup>[25]</sup>。本文结合关联性分析提出了一种基于图的协同过滤推荐算法,该算法通过图结构中顶

点和边的关系来表示描述不同事物的数据,借助图中路径长度、路径数以及顶点的度来计算相似度,并以图结构的形式来存储和记录数据以及数据之间的关系,一定程度上缓解了因为数据稀疏性而使得推荐准确度下降的问题,并在文中进行了验证。

下一步研究中,笔者计划选择在本文提出的算法中加入更多的组合类型进行对比,例如导演、演员和类型等大于2种电影属性的组合,并且考虑在系统中增加时间组件,利用在较短时间内同一用户前一次与后一次的查询请求具有较强关联性进行深入研究。

## 参考文献

- [1] HAN W T, MIAO Y S, LI K W, et al. Chronos: A graph engine for temporal graph analysis [C]// Proceedings of the 9th European Conference on Computer Systems. 2014: 1-6.
- [2] 卓煜, 尤佳莉, 王劲林. 爱奇艺视频网站服务特性分析 [J]. 网络新媒体技术, 2017, 6(2): 1-6.
- [3] 肖青秀, 汤鲲. 基于双层注意力机制的深度学习电影推荐系统 [J]. 计算机与现代化, 2018(11): 109-114.
- [4] LI Y Y, ZHENG Y F, KANG J, et al. Designing a learning recommender system by incorporating resource association analysis and social interaction computing [C]// State-of-the-art and Future Directions of Smart Learning. Springer, 2016: 137-143.
- [5] 陈天昊. 互联网电影推荐方法的研究与实现 [D]. 合肥: 中国科学技术大学, 2014.
- [6] 贺超波, 付志文, 刘海, 等. 基于兴趣社区的学习资源推荐模式设计 [J]. 中国教育信息化, 2017, 13(7): 51-54.
- [7] YUAN P P, XIE C F, LIU L, et al. PathGraph: A path centric graph processing system [J]. IEEE Transactions on Parallel & Distributed Systems, 2016: 2998-3012.
- [8] LIU S P, TU X H, LI R R. Unifying explicit and implicit feedback for top-N recommendation [C]// 2017 IEEE 2nd International Conference on Big Data Analysis. 2017: 35-39.
- [9] 陈汉福. 高校视频公开课点播平台智能推荐系统的设计与实现 [D]. 广州: 华南理工大学, 2016.
- [10] LOW Y, BICKSON D, GONZALEZ J, et al. Distributed GraphLab: A framework for machine learning and data mining in the cloud [J]. VLDB Endowment, 2012, 5(8): 716-727.
- [11] LEE J, ABU-EL-HAJA S. Large-scale content-only video recommendation [C]// 2017 IEEE International Conference on Computer Vision. 2017: 987-995.
- [12] SU H Y, LIN X F, YAN B, et al. The collaborative filtering algorithm with time weight based on MapReduce [C]// International Conference on Big Data Computing and Communications. 2015: 386-395.
- [13] SINCAN O M, YILDIRIM Z. A new similarity coefficient for a collaborative filtering algorithm [J]. Communications, 2017, 2(59): 41-54.

(下转第48页)

构和相关原理,并对 3 组实际数据展开实验,通过对本文的模型和经典软件可靠性模型、其他神经网络组合模型以及数据驱动模型在预测精度上的对比,验证了极限学习机算法在组合软件可靠性模型中的应用能够取得良好的效果。

#### 参考文献:

- [1] 陆民燕. 软件可靠性工程[M]. 北京: 国防工业出版社, 2011: 226-230.
  - [2] 张柯, 张德平, 汪帅. 软件可靠性混沌神经网络模型[J]. 计算机科学, 2014, 41(4): 172-177.
  - [3] LYU M R. Handbook of Software Reliability Engineering[M]. McGraw-Hill, 1996: 51-72.
  - [4] 刘遵, 郭立红, 肖辉, 等. 基于参数动态调整的动态模糊神经网络的软件可靠性增长模型[J]. 计算机科学, 2013, 40(2): 186-190.
  - [5] 刘遵, 郭立红, 肖辉, 等. 基于结合模拟退火算法的动态模糊神经网络的软件可靠性增长模型[J]. 吉林大学学报(工学版), 42(5): 1225-1230.
  - [6] 乔辉, 周雁舟, 邵楠, 等. 基于 AGA-LVQ 神经网络的软件可靠性预测模型研究[J]. 计算机科学, 2013, 40(1): 179-182.
  - [7] 熊小均, 梅登华. 基于改进型神经网络的软件可靠性模型[J]. 计算机工程, 2010, 36(22): 187-189.
  - [8] 孙志安, 裴晓黎, 宋昕, 等. 软件可靠性工程[M]. 北京: 北京航空航天大学出版社, 2009: 60-108.
  - [9] JELINSKI Z, MORANDA P. Software reliability research[M]// Statistical Computer Performance Evaluation. Academic Press, 1972: 465-484.
  - [10] GOEL A L, OKUMOTO K. Time-dependent error-detection rate model for software reliability and other performance measures[J]. IEEE Transactions on Reliability, 1979, 28(3): 206-211.
  - [11] LITTLEWOOD B, VERRALL J L. A Bayesian reliability growth model for computer software[J]. Journal of the Royal Statistical Society, 1973, 22(3): 332-346.
  - [12] 邹丰忠, 徐仁佐. 软件可靠性多模型综合评估[J]. 同济大学学报(自然科学版), 2002, 30(10): 1183-1185.
  - [13] IEEE Standards Software Reliability Working Group. IEEE Std 1633-2016 (Revision of IEEE Std 1633-2008) IEEE Recommended Practice on Software Reliability [EB/OL]. [2019-04-23]. <https://www.doc88.com/p-7324954906795.html>.
  - [14] LYU M R, NIKORA A. Applying reliability models more effectively[J]. IEEE Software, 1992, 9(4): 43-52.
  - [15] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 97-107.
  - [16] 贺智勇, 尹乾. 基于神经网络的级联软件可靠性模型[J]. 计算机工程与设计, 2009, 30(14): 3377-3379.
  - [17] 尹乾. 基于神经网络的软件可靠性模型[D]. 北京: 北京师范大学, 2006.
  - [18] 王高祖, 李伟华, 徐艳玲. 基于神经网络组合模型的软件可靠性评估研究[J]. 计算机仿真, 2010, 27(8): 176-180.
  - [19] 吴勤, 侯朝桢, 原菊梅. 软件可靠性多模型综合的预测方法[J]. 计算机工程, 2006, 32(23): 214-215.
  - [20] RUMELHART D E. Learning representations by back-propagating errors[J]. Nature, 1986, 23: 533-536.
  - [21] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.
  - [22] HUANG G, HUANG G B, SONG S, et al. Trends in extreme learning machines: A review[J]. Neural Networks, 2015, 61: 32-48.
  - [23] 张文博, 姬红兵. 融合极限学习机[J]. 电子与信息学报, 2013, 35(11): 2728-2732.
  - [24] ZHANG X, PHAM H. A software cost model with warranty cost, error removal times and risk costs[J]. IIE Transactions, 1998, 30(12): 1135-1142.
- 
- (上接第 43 页)
- [14] DING J, WANG Y B, WANG Q, et al. Football video recommendation system with automatic rating based on user behavior[C]// 2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics. 2017: 1-5.
  - [15] SHI C, LI Y T, ZHANG J W, et al. A survey of heterogeneous information network analysis[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(1): 17-37.
  - [16] SUN Y Z, HAN J W, YAN X F, et al. PathSim: Meta path-based top-k similarity search in heterogeneous information networks[J]. VLDB Endowment, 2011, 4(11): 992-1003.
  - [17] REMEZANI M, MORADI P, AKHLAGHIAN F. A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domain[J]. Physica A: Statistical Mechanics & Its Applications, 2014, 408(32): 72-84.
  - [18] MIAO Y S, HAN W T, LI K W, et al. Immortal graph: A system for storage and analysis of temporal graphs[J]. ACM Transactions on Storage, 2015, 11(3): 14.
  - [19] GONZALEZ J E, LOW Y, GU H, et al. PowerGraph: distributed graph-parallel computation on natural graphs[C]// Proceedings of USENIX Conference on Operating Systems Design and Implementation. 2012: 17-30.
  - [20] DELDJOO Y, ELAHI M, CREMONESI P, et al. Content-based video recommendation system based on stylistic visual features[J]. Journal on Data Semantics, 2016, 5(6): 99-113.
  - [21] WU G, SWAMINATHAN V, MITRA S, et al. Context-aware video recommendation based on session progress prediction[C]// 2017 IEEE International Conference on Multimedia and Expo. 2017: 1428-1433.
  - [22] BHATT C, COOPER M, ZHAO J. SeqSense: Video recommendation using topic sequence mining[C]// International Conference on Multimedia Modeling. 2018: 252-263.
  - [23] 张新慧. 面向网络学院学习者的学习路径推荐算法研究[D]. 西安: 西安交通大学, 2016.
  - [24] CHOI I Y, OH M G, KIM J K, et al. Collaborative filtering with facial expressions for online video recommendation[J]. International Journal of Information Management: The Journal for Information Professionals, 2016, 3(36): 397-402.
  - [25] 刘春霞, 陆建波, 武玲梅. 一种电影推荐模型: 解决冷启动问题[J]. 计算机与现代化, 2018(11): 83-87.