**基于用户聚类的协同过滤推荐算法研究**

**Collaborative filtering recommendation algorithm based on clustering basal users**

[1 引言 2](#_Toc479252079)

[1.1 研究背景及意义 2](#_Toc479252080)

[1.2 国内外研究现状 3](#_Toc479252081)

[1.2.1 国内研究现状 3](#_Toc479252082)

[1.2.2 国外研究现状 4](#_Toc479252083)

[1.3 研究内容和目标 4](#_Toc479252084)

[1.4 论文组织结构 5](#_Toc479252085)

[1.5 本章小结 6](#_Toc479252086)

[2 相关技术研究 6](#_Toc479252087)

[2.1 协同过滤算法 6](#_Toc479252088)

[2.1.1 基于用户的协同过滤算法 6](#_Toc479252089)

[2.1.2 基于项目的协同过滤算法 8](#_Toc479252090)

[2.1.3 协同过滤算法存在的问题及解决方案 9](#_Toc479252091)

[2.2 相似度度量 10](#_Toc479252092)

[2.2.1 余弦相似性度量 10](#_Toc479252093)

[2.2.2 修正的余弦相似性度量 11](#_Toc479252094)

[2.2.3 Pearson相关系数 11](#_Toc479252095)

[2.3 聚类算法 11](#_Toc479252096)

[2.3.1 Canopy算法 11](#_Toc479252097)

[2.3.2 K-means算法 13](#_Toc479252098)

[2.4 本章小结 13](#_Toc479252099)

[3 基于改进的用户聚类的协同过滤推荐算法 13](#_Toc479252100)

[3.1 离线用户聚类模块 13](#_Toc479252101)

[3.1.1 Canopy确定聚类数目和初始聚类中心 14](#_Toc479252102)

[3.1.2 k-Means基本用户聚类 14](#_Toc479252103)

[3.2 在线搜索最近邻居并产生推荐模块 15](#_Toc479252104)

[3.2.1 问题描述与算法思想 15](#_Toc479252105)

[3.2.2 融合用户评分差异度的Pearson相似度 17](#_Toc479252106)

[3.2.3 融合用户评分差异度的推荐算法设计 18](#_Toc479252107)

[3.3 实验结果与分析 19](#_Toc479252108)

[3.3.1 实验环境及实验数据 19](#_Toc479252109)

[3.3.2 度量标准 20](#_Toc479252110)

[3.3.3 实验结果分析 20](#_Toc479252111)

[4 电影推荐系统设计与实现 20](#_Toc479252112)

[4.1 系统需求分析 20](#_Toc479252113)

[4.2 系统设计 20](#_Toc479252114)

[4.2.1 系统架构设计 20](#_Toc479252115)

[4.2.2 系统功能模块设计 20](#_Toc479252116)

[4.2.3 系统数据库设计 20](#_Toc479252117)

[4.3 推荐系统实现 21](#_Toc479252118)

[4.3.1 系统实现环境 21](#_Toc479252119)

[4.3.2 系统功能实现 21](#_Toc479252120)

[4.4 本章小结 21](#_Toc479252121)

[5 结论与展望 21](#_Toc479252122)

[5.1 本文总结 21](#_Toc479252123)

[5.2 工作展望 21](#_Toc479252124)

[6 参考文献 21](#_Toc479252125)

**关键词；协同过滤；用户聚类；个性化推荐**

1. 引言
   1. 研究背景及意义

随着信息技术的迅速发展，互联网已经成为人们获取资源和交流信息的重要渠道。根据互联网信息中心（CNNIC）已完成的第39次《中国互联网发展状况统计报告》显示：截止2016年12月，中国网名规模达7.31亿，全年共计新增网名4299万人。互联网普及率为53.2%，较2015年底提升了2.9个百分点[1]。现如今，随着互联网的普及，网络资源不断丰富，互联网中的信息呈指数爆炸型增长。由于用户量以及信息量的增长使得用户很难在大量资源中检索到自己需要的或者适合自己的资源，从而出现了所谓的信息过载现象。信息过载是指人们接受的了太多的信息，却无法有效整合、组织及内化成自己需要的信息，以致影响到人们的工作、生活以及人际关系等[2]。因此，如何能够有效的筛选和过滤信息成为了当今互联网领域研究的重要问题。搜索引擎技术是帮助用户获取有用信息的技术之一，当用户输入检索的关键字后，搜索引擎会将这些关键字与互联网中的海量信息进行比对，从而将比较符合的信息返回给用户。搜索引擎在一定程度上减少了用户检索信息耗费的时间，并且提高了互联网的服务质量。但是搜索引擎检索出来的结果无法满足用户的个性化需求，不同的用户检索同一关键字返回的检索结果都是相同的。为了解决这一问题，出现了另一种筛选和过滤信息的技术，个性化推荐系统。

个性化推荐系统是缓解信息过载问题最有效的办法之一。推荐系统能够根据用户兴趣特点和购买行为，分析出用户的兴趣爱好和需求，从海量信息中找到用户感兴趣的内容，并推荐给用户。随着Web2.0时代的到来，个性化推荐系统受到越来越广泛的应用。在国内，大型的电子商务网站，如淘宝、当当等都使用推荐系统向客户推荐其可能感兴趣或者满意的商品。在国外，雅虎、亚马逊等网站中也可以看到它的身影。个性化推荐系统是建立在海量数据挖掘基础上的一种智能平台，它能够为用户提供完全个性化的决策支持和信息服务[3]。好的推荐系统可以大大提高用户的忠诚度，并为电子商务带来巨大的利益。目前，各大型电子商务网站，如当当、淘宝、天猫、亚马逊等都不同程度的使用了推荐系统，用以向用户推荐商品，提高经济效益。

当前常见的推荐系统可以分为基于内容过滤的推荐系统、基于协同过滤的推荐系统以及混合推荐系统。基于内容过滤的推荐系统，顾名思义，是通过分析资源及用户档案等内容产生推荐结果。与基于内容过滤的推荐系统不同，基于协同过滤的推荐系统所做的，不是从数据中分析出信息，而是建立一套有效的评价反馈机制，让用户形成良好的反馈，让用户本身对产品进行推荐。也就是说，用户得到的推荐，也许根本不是从数据中挖掘出来的，而是由其他用户贡献的。在这里，用户群体，也成为了个性化推荐产品的一部分。

* 1. 国内外研究现状

20世纪90年代中期，推荐系统的概念首次被提出。后来信息技术的进步，电子商务和网上交易的蓬勃发展为推荐系统提供了良好的契机和发展平台，推荐系统得到了飞速的发展[4]。由于巨大的应用需求和使用价值，推荐系统因此得到广泛的关注，许多国内外学者也纷纷将推荐系统作为研究对象。

* + 1. 国内研究现状

国内目前的研究多集中在推荐系统的理论和技术方面，并取得了丰硕的成果。针对评价数据的稀疏性，文献[9]提出一种优化的协同过滤推荐算法，文献[10]提出一种基于项目评分预测的协同过滤推荐算法，采用一种新颖的相似性度量方法计算目标用户的最近邻居。针对算法的可扩充性，文献[11]提出基于项目聚类的协同过滤推荐算法，显著的提高了推荐系统的推荐质量。以上各种推荐算法都各有各的优缺点，因此许多研究者提出结合多种过滤技术，以克服各自的优缺点。例如文献[12]提出基于内容和合作模式的信息推荐机制，文献[13][14]提出基于语义相似性的资源协同过滤技术。文献[15]对稀疏问题和冷启动问题做了系统研究，文献[16]研究了基于神经网络和模糊逻辑的智能推荐系统。文献[17]将推荐系统的应用领域扩展到数字图书馆。

* + 1. 国外研究现状

国外在这方面的研究起步比较早。在用户模型方面，文献[5]提出使用智能代理技术来分析顾客的需求，建立用户模型，并据此提供推荐服务。在推荐算法方面，协同过滤技术是应用最广泛的个性化推荐技术之一，基于用户之间的相似性，即邻居用户产生推荐结果。随着推荐系统规模的扩大，实时性研究逐渐成为热点。文献[6]使用奇异值分解技术减少特征向量空间的维数，提高最近邻居的搜索速度。文献[7]对用户最近邻居和项采用不同权重的方法来提高推荐质量。由于传统的协同过滤技术根据用户显式评分产生推荐结果，用户使用不方便，很多研究者提出可以通过web挖掘技术获取用户隐式评分，由此，各种数据挖掘技术也开始应用到推荐系统中。文献[8]中提出了一种利用神经网络和遗传K-means算法通过分析用户在电子商务网站的浏览路径来获取用户偏好的方法。各种挖掘技术如关联规则挖掘、聚类挖掘等被应用于web日志的分析中，以提高推荐系统的精度。在推荐系统的应用方面，出现了一些研究型推荐系统实例，如GroupLens，自动协同过滤推荐系统，用于新闻信息推荐。GroupLens系统通过用户的评分信息自动搜索用户的最近邻，然后根据评分信息产生最终的推荐结果，适合于用户量比较大的场合。MovieLens也是一种自动协同过滤推荐系统，用于推荐电影。与GroupLens不同，MovieLens是一个基于web的推荐，系统通过浏览器方式进行用户评分数据收集和推荐结果显示，用户使用方便。

* 1. 研究内容和目标

本文深入研究了个性化推荐系统中的相关算法，并重点介绍了传统协同过滤推荐算法的工作原理及面临的主要问题和相关解决方案。针对传统相似度计算方法在数据稀疏的情况下推荐不准确问题，提出一种融合了用户间评分差异的相似度计算方法。

本文针对推荐系统中应用比较广泛的协同过滤推荐算法展开研究，主要做了以下几个方面的工作：

1. 分析协同过滤推荐算法的发展现状，讨论现有推荐算法存在的不足及其解决方案。
2. 分析传统协同过滤推荐算法产生推荐的三个过程，针对Pearson相似度计算方法存在的问题提出一种融合了用户评分差异度的改进Pearson相似度计算方法。传统Pearson相似度计算用户相似性时，忽略用户对项目的评分差异，导致获取的目标用户最近邻居并非符合实际。本文提出的算法结合传统相似度计算方法，
3. 采用MovieLens数据集做实验，比较本文提出的改进算法与传统协同过滤推荐算法的推荐效果，验证本文提出的改进算法的可行性。
4. 基于融合用户差异度的改进算法设计并实现了电影推荐系统，该系统从用户和管理员的角度出发，为用户提供了电影推荐、用户注册、用户登录、用户评分、电影信息查询等功能。为管理员提供了电影编辑、用户管理、评论管理等功能。该系统分析了电影系统中各个功能模块并设计了数据库，最后结合Html+CSS+Javascript+PHP技术实现了电影推荐系统。
   1. 论文组织结构

论文共分为五个章节，根据上述论文的主要工作，对本文的组织结构安排如下：

第一章，本章介绍了本课题的选题背景和意义，分析了课题的研究背景及国内外研究现状，并阐述了论文的主要研究内容和目标，本章的最后对论文的组织结构进行了说明。

第二章，本章介绍了传统协同过滤推荐算法和K-means聚类技术的相关概念，并分析了现有的协同过滤推荐算法存在的问题及解决方案。

第三章，本章阐述了K-means聚类算法及其在协同过滤推荐算法中的应用，提出了一种改进的基于用户聚类的协同过滤推荐算法，

第五章，系统实现。该章节详细介绍了系统各个模块的实现细节，并提供了关键部分的结果演示。

第六章，结论与展望。本章对全文的内容进行了总结，描述了本课题所完成的任务，并指出本课题中存在的不足以及将来的研究方向。

* 1. 本章小结

本文首先阐述了论文的研究背景和意义。随后介绍了推荐算法的国内外研究现状。接着对本文的主要研究内容进行了简单介绍。最后描述了本文的基本结构。

1. 相关技术研究
   1. 协同过滤算法

协同过滤推荐算法是诞生最早，并且较为著名的推荐算法。协同过滤的概念是在1992年由Goldberg、Nicols、Oki及Terry首次提出的，应用于Tapestry系统[20]，该系统仅适用较小用户群（比如一个单位内部），而且对用户有过多要求（比如，要求用户显式的给出评价）。作为协同过滤推荐系统的雏形，Tapestry展示了一种新的推荐思想，但存在许多技术上的不足。其后，出现了基于评分的自动协同过滤推荐系统，例如，推荐新闻和电影的GroupLens。

作为当前研究最多、应用最广泛的推荐算法，协同过滤推荐算法的主要功能是预测和推荐，算法通过对用户历史行为数据的挖掘发现用户偏好，基于不同的偏好对用户进行群组划分并推荐品味相似的商品。具体的推荐原理如图1所示。



图1[21] 经典协同推荐原理

一般来说，协同过滤算法可以分为两类：基于用户的协同过滤推荐算法和基于项目的协同过滤推荐算法。

* + 1. 基于用户的协同过滤算法

基于用户的协同过滤推荐技术是推荐系统中应用最为广泛的技术，基于用户的协同过滤推荐算法基于这样一个假设，即兴趣偏好相同或相似的用户，他们的需求也相似。算法首先找出与目标用户兴趣相似的邻居集合，然后利用K个最近邻居对项目的加权评分来预测目标用户对项目的评分，进而为目标用户产生Top-N推荐。

寻找最近邻居：在这一阶段，主要完成对目标用户最近邻居的查找。通过计算目标用户与其他用户之间的相似度，算出与目标用户最相似的“最近邻居集”。即对目标用户产生一个以相似度

协同过滤推荐算法是从用户的行为和偏好中发现规律的，并且基于此为用户产生推荐列表。因此，如何收集用户的偏好信息成为系统推荐效果最基础的决定因素。用户可以通过很多方式提供自己的偏好信息，用户对物品或者信息的偏好，根据其应用本身的不同，可以包括用户对物品的评分，用户查看物品的记录，用户的购买信息等。表1中列举了部分可提供偏好信息的方式，这些用户的偏好信息可以分为两类：显式的用户反馈和隐式的用户反馈。

表1 用户行为和用户偏好

|  |  |  |  |
| --- | --- | --- | --- |
| 用户行为 | 类型 | 特征 | 作用 |
| 评分 | 显式 | 整数量化的偏好，可能取值是[0,n]；n一般取值为5或者10 | 通过用户对项目的评分，可以精确的得到用户的偏好 |
| 购买 | 显式 | 布尔量化的偏好，取值是0或1 | 用户的是否购买能够很明确的反应他对该项目的偏好 |
| 投票 | 显式 | 布尔量化的偏好，取值是0或1 | 通过用户对项目的投票，可以精确的得到用户的偏好 |
| 转发 | 显式 | 布尔量化的偏好，取值是0或1 | 通过用户对项目的转发，可以精确地得到用户的偏好 |
| 收藏 | 显式 | 布尔量化的偏好，取值是0或1 | 通过用户对项目的收藏，可以精确的得到用户的偏好 |
| 点击流（查看） | 隐式 | 一组用户的点击，可通过分析，得到用户的偏好 | 用户的点击一定程度上反映了用户的注意力，所以它可以从一定程度上反应用户的喜好 |
| 页面停留时间 | 隐式 | 一组时间信息，噪音大，需要进行去噪，分析，得到偏好 | 用户在页面停留的时间一定程度上反应了用户的注意力和喜好，但是噪音大，不好利用 |

协同过滤推荐算法首先通过显式或隐式的方式收集用户的评分、评价行为等，然后对收集到的数据进行数据清理、转换和录入，最终形成用户对各种项目的评价矩阵，如表2所示。其中：*Rij*代表第*i*个用户*ui*对项目*Ij*的评分。一般来说，。分数越高，用户对该项目的认可程度越高。

表2 用户评分矩阵

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用户 | 商品 | | | | | |
| Item1 | Item2 | … | Item*j* | … | Item*n* |
| User1 | 3 | 5 | … | 2 | … | / |
| User2 | / | 4 | … | 5 |  | 4 |
| … | … | … | … | … | … | … |
| User*i* | 4 | 3 | … | / | … | 3 |
| … | … | … | … | … | … | … |
| User*m* | 5 | / | … | 4 | … | 5 |

* + 1. 基于项目的协同过滤算法

基于项目的协同过滤推荐也是协同过滤推荐的一种，该算法基于一种假设，即当前用户对多用户评分相同或相似的项目评分也相似。该算法首先找出与项目i 评分相似的邻居项目，然后利用K个最近邻居项目的加权来预测目标用户对项目i的评分，进而为用户产生推荐。

当我们通过分析用户的偏好信息得到评分矩阵后，我们可以根据用户喜好计算相似用户和物品，然后基于相似用户或者物品对目标用户进行推荐，这就是最典型的协同过滤的两个分支：基于用户的协同过滤和基于项目的协同过滤。

* **相似邻居的计算**

介绍完相似度的计算方法，下面我们看看如何根据相似度找到用户-项目的邻居，常用的挑选邻居的原则可以分为两类：固定数量的邻居和基于相似度门槛的邻居。

1. 固定数量的邻居

使用这种方法寻找邻居用户集时不论邻居的远近，只取最近的K个作为目标用户邻居。如图(a)所示，假设要计算3的5-邻居，那么根据点之间的距离，我们取最近的五个点，分别是点1，点2，点5，点6和点8。但是很明显的可以看出，这种方法对于孤立的点的计算效果不好，因为要取固定个数的邻居，当它附近没有足够多比较相似的点，就被迫取一些不太相似的点作为邻居，这就影响了邻居相似的程度，比如图中，点6和点3并不是很相似。

1. 基于相似度门槛的邻居

与计算固定邻居的原则不同，基于相似度门槛的邻居计算是对邻居的远近进行最大值的限制，落在以当前点为中心，距离K的区域内的所有点都作为当前点的邻居，这种方法计算得到的邻居数目个数不确定，但相似度不会出现较大的误差。如图中，从点3出发，计算相似度在K内的邻居，得到点1，点2，点5和点8，这种方法计算出的邻居的相似程度比前一种优，尤其是对孤立点的处理。

图 相似邻居计算示意图

* + 1. 协同过滤算法存在的问题及解决方案

随着个性化推荐系统在应用中的迅猛发展，协同过滤推荐技术的应用越来越广泛，通过对协同过滤算法的深入研究，发现其在实际应用中还存在很多的问题。这些问题主要表现为数据稀疏问题、冷启动问题以及可扩展性问题。

（1）数据稀疏问题

稀疏性问题是造成协同过滤推荐系统推荐质量下降的一个首要问题。协同过滤算法依靠用户对资源项目的评价产生推荐。如果用户对资源项目的评价不够多，那么依靠少量的评价无法产生精确地推荐，这就是协同过滤所遇到的评价数据的稀疏问题。

实际上，在一个大型的电子商务网站如淘宝、京东等中，陈列的资源项目往往达成百上千万，而一个普通用户所评价的项目数目仅占网站总商品数量的微乎其微的一部分，因此，造成评分矩阵非常的稀疏。此外，数据的高维性是引起评价数据稀疏性的一个重要原因。这些问题使得基于协同过滤算法的应用受到局限。

（2）冷启动问题

冷启动问题是协同过滤算法中比较经典的一个问题，该问题一直影响着协同过滤推荐算法的推荐质量，广义上的冷启动问题包括项目冷启动和用户冷启动。

协同过滤系统是依靠用户对项目的评价产生推荐。在基于项目的协同过滤系统中，当系统中加入一个新的项目，该项目没有被任何的用户评价过，那么根据协同过滤算法，这个项目就永远也不会有被推荐的机会。那么对于这个项目而言，推荐系统就失去了作用。这就是推荐系统冷启动的问题。还有一种情况出现在基于用户的协同过滤推荐系统中，当一个新的用户加入时，该用户最初向系统提供的自己的信息，包括兴趣爱好、评分等非常有限，因此系统无法找到该用户的最近邻居，从而也无法进行推荐。无论是项目冷启动还是用户冷启动问题，都是由于用户的评价行为不够产生的。其中，项目的冷启动问题更具有代表性。对于一个电子商务推荐系统而言，总是不断的有新项目加入到系统中来，如何使这些没有用户评价过的项目能够获得推荐的机会，是每个推荐系统都要面临的问题。

（3）可扩展性问题

随着计算机网络技术的迅速发展，带动电子商务系统的规模不断扩大，用户量和项目数也随之不断增加，这使得协同过滤算法的复杂度剧增，传统协同过滤算法会呈现出缺乏可扩展性问题。

许多在线系统在考虑用户的购买和评价历史记录后，需迅速为用户做出推荐，传统协同过滤算法在最近邻居搜索时需要扫描整个数据集，在一个用户和商品数量均以万计的电子商务推荐系统中，计算量随着用户和项目的不断增多变得越来越大，同时为数以万计的用户提供实时的推荐变得越来越困难，以致无法满足推荐的实时性要求，此时算法即使再精确对推荐系统而言也是徒劳。因此，如何使得协同过滤算法在用户和项目数量不断增加的情况下尽量降低在线推荐系统所需的时间成为协同过滤算法研究的一个重要课题。

* 1. 相似度度量

相似度计算是协同过滤推荐算法中最关键的一步。关于相似度的计算，现在的几种基本方法都是基于向量的，其实就是计算两个向量的距离，距离越近相似度越大。在用户-项目矩阵中，我们可以把用户对项目的偏好作为一个向量来计算用户之间的相似度，或者将所有用户对某个项目的偏好作为一个向量来计算物品之间的相似度。相似性的度量方法有很多种，不同的度量方法的应用范围也不一样，传统的相似度计算方法有三种[19]。本文以基于用户的协同过滤算法为例说明相似性的计算方法，基于项目的协同过滤算法的相似性计算与其类似。

* + 1. 余弦相似性度量

把用户评分看做n维项目空间上的向量，用户间的相似性通过向量间的余弦夹角度量，设用户*i*和用户*j*在n维项目空间上的评分分别表示为向量，，则用户*i*和用户*j*之间的相似性为：

* + 1. 修正的余弦相似性度量

余弦相似性度量方法中没有考虑不同用户的评分尺度问题，修正的预先相似性度量方法通过减去用户对项目的平均评分来改善上述缺陷，设经用户*i*和用户*j*共同评分的项目集合用*Iij*表示，*Ii*和*Ij*分别表示经用户*i*和用户*j*评分的项目集合，则用户*i*和用户*j*之间的相似性为：

其中，表示用户*i*对项目c的评分，和分别表示用户*i*和*j*对项目的平均评分。

* + 1. Pearson相关系数

设经用户*i*和用户*j*共同评分的项目集合用*Iij*表示，则用户*i*和用户*j*之间的相似性*sim(i,j)*通过Pearson相关系数度量：

其中，表示用户*i*对项目c的评分，和分别表示用户*i*和*j*对项目的平均评分。

总体来讲，三种方式均为基于向量的相似度计算方式，进行对象属性之间的严格匹配。

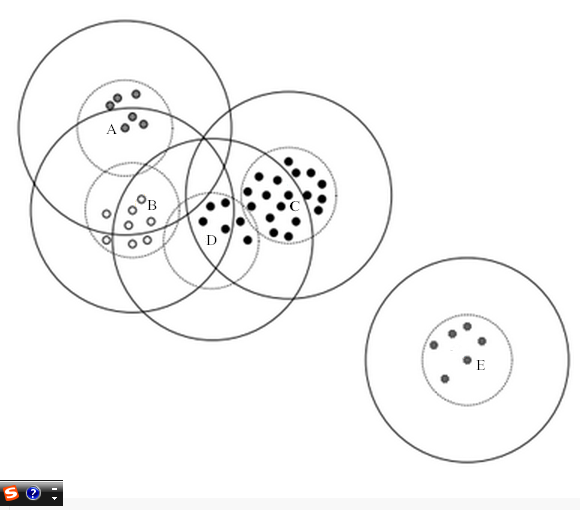
余弦相似性度量方法把用户评分看做一个向量，用向量的余弦夹角度量用户间的相似性，然而没有包含用户评分的统计特征；修正的余弦相似性方法在余弦相似性基础上，减去了用户对项目的平均评分，然而该方法更多体现的是用户之间的相关性而非相似性，相关性

* 1. 聚类算法
     1. Canopy算法

Canopy生成算法也被称为Canopy聚类，它是2000年提出来的，是一种快速近似的聚类算法。它的优势在于得到簇的速度非常快，只需一次遍历数据即可得到结果。该算法无法给出精确的簇结果，但是它可以给出最优的簇的数量。与传统的聚类算法不同，Canopy聚类的最大特点是不需要事先指定K值（即聚类中心的个数），因此具有很大的使用价值。

Canopy聚类算法是将对象分组到类的简单、快速、精确的方法。每个对象用多维特征空间里的一个点来表示。这个算法使用一个快速近似距离度量和两个距离阈值T1>T2来处理。基本的算法是，从集合中随机选取一个点开始并且从集合中删除这个点，创建一个包含这个点的Canopy，并在剩余的点集合上迭代。对于每个点，如果它距离这个点的距离小于T1，就将这个点加入这个聚集中。如果这个距离小于T2，就将这个点从集合中删除。这种方法可以防止一个现有的Canopy点成为新的Canopy中心。这个算法循环到初始集合为空为止，这样会得到若干Canopy，Canopy之间可以是重叠的，但不会存在某个对象不属于任何Canopy的情况。

图是一个Canopy的例子，其中包含5个数据中心向量。



图[21] Canopy聚类图

图中的数据向量用同样灰度值表示的属于同一个聚类。聚类中心向量A被随机选出，然后以A的数据向量创建一个Canopy，这个Canopy包括所有在其外圈（实线圈）的数据向量，而内圈（虚线圈）中的数据向量则不再作为中心向量的候选名单。

表 Canopy算法流程

|  |  |
| --- | --- |
| 1 | 将数据集向量化得到样本集合List，选择两个距离阈值：T1，T2，且T1>T2。（对应上图，实线T1，虚线T2） |
| 2 | 从List中任取一个样本点p，作为一个canopy，并从List中移除p。 |
| 3 | 计算List中所有点到p的距离dist。 |
| 4 | 若dist<T1，则将点p加入到这个canopy，作为弱关联。 |
| 5 | 若dist<T2，则将点p从List中移除，作为强关联。 |
| 6 | 重复步骤2~5，直至List为空结束。 |

Canopy一般用在K-means之前的粗聚类。考虑到K-means均值在使用上必须要确定K的大小，而往往数据集预先不能确定K值的大小，这样如何K值取的不合理会带来K-means的误差很大。选择利用Canopy聚类作为K-means的前奏比较科学。

* + 1. K-means算法
  1. 本章小结

1. 基于改进的用户聚类的协同过滤推荐算法

本文提出的改进的基于用户聚类的协同过滤推荐算法主要分为两部分：离线用户聚类模块和在线搜索最近邻并产生推荐模块。离线时，算法首先对基本用户聚类，产生若干用户聚类中心及用户类别所属矩阵。在线时，算法仅仅计算目标用户与各个聚类中心的相似性，从而求出目标用户所属类别，然后通过搜索离线时获得的类别所属矩阵找出与目标用户同类别的基本用户。通过计算目标用户与这些基本用户之间的相似性获得目标用户的最近邻居，再由这些最近邻居进行目标用户评分的预测，并产生推荐。离线用户聚类算法对基本用户进行聚类，使得每一类中的基本用户数目远远少于全部的基本用户数目，在线时系统只需要计算目标用户与其同类别的基本用户之间的相似性，因而，在线查找目标用户的最近邻居所需时间将大大缩短。

* 1. 离线用户聚类模块

本文算法对基本用户聚类的目的是产生基本用户的类别所属矩阵，使得在线时系统能够通过类别所属矩阵快速搜索到目标用户的最近邻居。

目前聚类方法很多，本文采用k-Means算法[22]的思想对基本用户进行聚类。由于使用k-Means算法时需要事先给出聚类数目，因而本文首先采用对基本用户进行预处理用以确定聚类数目。因此，本文提出的离线用户聚类模块分为两个阶段，阶段一，本文使用了Canopy聚类算法给出聚类数目和初始聚类中心。Canopy算法中首先计算基本用户两两之间的相似性，将相似性大于一定阈值的基本用户归于同一原始类别中，然后选取包含用户数量最多的前k个原始类别初始条件，通过计算它们的均值得到k个初始聚类中心。阶段二，使用k-Means聚类算法对基本用户进行聚类，其中的聚类数目k和初始聚类中心由阶段一所得。

* + 1. Canopy确定聚类数目和初始聚类中心

**算法1** Canopy算法获取聚类数目和初始聚类中心

**输入：**数据源*D = (U，I，R)*。

**输出：**聚类数目*k，k*个初始聚类中心*Cluster( k, n)*。

**方法：**

1. 计算基本用户两两之间的距离*dist( i, j)*。
2. 选择距离阈值T2。
3. 从数据源*D = (U，I，R)*中任选一个对象*p*，作为一个*canopy*，并从*D = (U，I，R)*中移除*p*。
4. 计算数据源*D = (U，I，R)*中所有对象到*p*的距离dist。
5. 若*dist <T2*，则将该对象加入这个*canopy*，并将其从数据源*D = (U，I，R)*中移除。
6. 重复步骤3~5，直至数据源*D = (U，I，R)*为空为止。
7. 选取包含用户量最多的前*k*个*canopy*作为初始条件。
8. 计算每个原始类别中的对象的均值作为初始的聚类中心。

本文使用Canopy算法预处理数据的目的是得到聚类数目*k*和初始聚类中心*Cluster( k, n)*，所以计算对象间相似性时本文选择了简单、计算代价较低的余弦相似性计算方法，对数据进行“粗”聚类，然后将相似的对象放在一个子集中，这个子集被叫做Canopy ，通过一系列计算得到若干Canopy，最后计算每个Canopy中对象的均值作为初始聚类中心。对于借助Canopy算法计算k-Means中的*k*值来说，Canopy算法中的距离阈值T1没有意义，只需要设定距离阈值T2 (T1 > T2) 即可，本文将T2设置为用户间平均相似度的一半。

* + 1. k-Means基本用户聚类

**算法2** k-Means基本用户聚类算法

**输入：**数据源*D = (U，I，R)*，聚类数目*k*，初始聚类中心*Cluster( k, n)*。

**输出：***k*个用户聚类中心*Cluster( k, n)*，类别所属矩阵*Idx( m, 1)*。

**方法：**

1. 计算每个对象与各个聚类中心之间的相似性，将对象分配到相似性最大的聚类中。
2. 所有对象分配完后，计算同一聚类中的所有用户的平均评分作为该聚类的聚类中心。
3. 与前一次计算得到的k个聚类中心比较，如果聚类中心发生变化，转到第(1)步，否则转到第(4)步。
4. 当聚类中心不发生变化时停止并输出聚类结果。

由算法2获得用户聚类中心*Cluster( k, n)*以及用户类别所属矩阵*Idx( m, 1)*。

用户聚类中心*Cluster( k, n)*中，*k*行代表*k*个用户聚类中心，*n*列代表*n*个项目，第*i*行第*j*列的元素*cij*代表用户聚类中心*i*对项目*j*的评分，其实用户聚类*i*中所有用户对项目*j*评分的均值。

基本用户的类别所属矩阵*Idx( m, 1)*中，*m*行代表每个基本用户，1列代表用户所属类别，第*i*行第1列的元素*Idx( i,* 1*)*代表用户*i*所属的类别。经过聚类，每个类别中的用户数目远远小于基本用户数目。

* 1. 在线搜索最近邻居并产生推荐模块

本文提出的算法在上述离线处理结果的基础上，首先计算目标用户与各个聚类中心之间的相似性，获得目标用户所属各个聚类程度的向量，将目标用户分配到相似性最大的聚类中，然后搜索类别所属矩阵*Idx*(*m,* 1)，找到与目标同类别的基本用户，计算目标用户与这些基本用户之间的相似性，确定目标用户的最近邻居。

* + 1. 问题描述与算法思想

传统的相似度计算方法在协同过滤技术中存在一定的弊端，如：在数据高维稀疏的情况下，用户之间关注圈交集(共同评分项目)的规模大多偏小且不一致，传统的相似性度量方法容易过分的快打或者缩小用户间的真实相似性[23]；受数据稀疏等影响，推荐精度较低[23]；Pearson相关系数必须满足数据之间的线性关系以及残差相互独立且均值为0等假设[23]。当这些条件不满足时，其计算准确度将会降低。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户 |  |  | 项目 | |
| *I*1 | *I*2 | *I*3 | *I*4 |
| *u*1 | 2 | 1 | 2 | 1 |
| *u*2 | 5 | 4 | 5 | 4 |
| *u*3 | 4 | 5 | 4 | 5 |
| *u*4 | 2 | 1 | 2 | 2 |
| *u*5 | 2 | 0 | 0 | 0 |

若使用Pearson相似度计算User1与其他用户之间的相似度则有：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *u*1 | *u*2 | *u*3 | *u*4 | *u*5 |
| *u*1 | 1 | 1 | -1 | 0.5774 | 1 |

从表可以看出，对于用户*u*1和*u*2，它们共有4个共同评分的项目，用Pearson相关系数计算*u*1和*u*2的相似性*sim*(*u*1, *u*2)=1，完全正相关，相似度最高，而实际上，*u*1的评分整体上偏低，*u*2的评分整体偏高，所以它们的相似度并没有那么高。对于用户*u*1和*u*3，相似性*sim*(*u*1, *u*2)= -1，完全负相关，相似度最低，虽然*u*1和*u*3的普遍评分偏低，但它们的相似度并没有那么低。对于判断*u*4和*u*5谁与*u*1更相似时，由于*u*1和*u*5只有一个共同评分项且评分一样，用Pearson相关系数计算得到*sim*(*u*1, *u*4)= 0.9045，*sim*(*u*1, *u*5)=1，而*u*4与*u*1有三个共同评分项目且评分一致，它们的相似度应该更高。

通过上面的分析可以看出，Pearson相关性分析方法存在以下几个问题：

1. 没有考虑共同评分项的数量对相似度的影响。

比如有两个用户他们共同观看了100部电影，虽然在电影的评分上不一定完全一样，但是他们之间的相似度应该要比只观看了两部相同电影的用户高。但是使用Pearson相关性计算时，结果并一定如此。若用户对观看的这两部电影的评分与目标用户相似或相近，那么，通过Pearson相关性计算得到的相似度一定会大于共同观看了100部电影的用户。可以看出Pearson系数只会对重叠的记录进行计算。

1. 如果用户间只有一个共同评分项，则相关性无法计算。

对于用户而言，如果两个用户间的共同评分项只有一个，那么从数学的角度上讲，至少有一组记录的标准差为0，这导致分母为0，无法计算相关性。

1. 如果用户间的评分完全相同，则无法计算相关性。

使用Pearson相关性计算时，如果两个用户间的共同评分项的所有评分都一样，则和（2）相同，至少会有一组记录的标准差为0，这导致分母为0，同样无法计算相关性。

1. Pearson系数对绝对数值不敏感

从表中可以看出，*u*1和*u*2虽然评分的项目完全相同，并且评分的趋势也相同，但是*u*1的整体评分偏低*， u*2的评分整体偏高，Pearosn相关性计算对绝对数值并不敏感。

所以对于Pearson相似度计算方法而言，无法准确的计算出某些用户评分之间的相似度。 针对Pearson系数对绝对数值不敏感的现象，本文提出一种改进的Pearson相似度计算方法，该方法融合了用户间评分的差异度。

* + 1. 融合用户评分差异度的Pearson相似度

本文提出一种改进的Pearson相似度计算方法，该方法使用用户间的评分差异度来修正原始的相似性计算。综合考虑用户评价对推荐效果的影响。

在文中的改进算法中需要计算用户间共同评分项的评分差异度，两个用户之间评分差异度越大，表明两个用户之间的相似度越低。用户间评分差异度计算步骤如下：

1. 假设用户*u*1和*u*2共同评分的项目集合为*I* = {*I*1，*I*2，…，*In*}，*u*1和*u*2共同评分为*u*1={，，…，}和*u*2={，，…，}，*u*1和*u*2的评分差异*d*(*u*1,*u*2)定义为：

*d*(*u*1,*u*2) = ()

=(*d*1，*d*2，…，*d*n)

(2) 使用欧几里得定理计算*u*1与*u*2之间的评分差异度，差异度

考虑到两个用户拥有的共同评分项数目n也会对相似度产生影响， n越大，相似度越大，所以加入1/*n*作为权重，新的差异度计算公式为：

式中：*n*为用户*u*1和*u*2共同评分集合大小；*di*为第*i*项评分的差值；由公式可知，取值范围为0到，

1. 将归一化到(0,1)

由于越大相似度越低，所以采用指数函数对做归一化操作，归一化之后，差异度的取值范围为0到1，值越大，用户间的相似度越高。

针对于表，计算用户*u*1与其他用户之间的评分差异度。首先计算用户*u*1与其他用户之间的评分差异矩阵，如表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户 | 商品 | | | | |
| *I*1 | *I*2 | *I*3 | *I*4 |
| *u*1 | 0 | 0 | 0 | 0 |
| *u*2 | -3 | -3 | -3 | -3 |
| *u*3 | -2 | -4 | -2 | -4 |
| *u*4 | 0 | 0 | 0 | -1 |
| *u*5 | 0 | / | / | / |

然后计算用户评分差异度，如表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *u*1 | *u*2 | *u*3 | *u*4 | *u*5 |
| *u*1 | 1 | 0.2231 | 0.2057 | 0.7788 | 1 |

如表所示，用户间评分差异度可以有效的改善Pearson相关性计算中对绝对数值不敏感的问题。对于用户间只有一个共同评分项的问题，如*u*1和*u*5，在实际情况中，因为项目的数量比较大，可以通过设置共同评分阈值来解决这个问题，如果共同评分项数目少于共同评分阈值，则认为两个用户之间没有可比性，将相似度置为0。

在本文的改进算法中需要在用户对项目的评分相似性*simpearson*(*i,j*)中融合用户之间的评分差异度，将两者进行线性组合，组合后的相似性作为用户之间的最终相似性。用户对项目的评分相似性基于用户-项目评分矩阵R，使用Pearson相关相似性公式计算。用户之间的评分差异度基于评分差异矩阵，使用公式（）计算。然后将用户对项目的评分相似性*simpearson*(*i,j*)和用户间的评分差异度进行线性组合，组合公式为：

*sim*(*i, j*) *= a simpearson*(*i, j*) *+* (1*-a*)

*a*为设定平衡因子，用来调节用户对项目的评分相似性和用户间评分差异度。

* + 1. 融合用户评分差异度的推荐算法设计

根据上述的相似度度量方法，将在线搜索目标用户最近邻居和产生推荐的算法描述如下：

**算法3** 在线搜索最近邻居算法

**输入：**目标用户评分向量，用户聚类中心矩阵*Cluster( k, n)*，类别所属矩阵*Idx*(*m*,1)。

**输出：**目标用户的*l*个最近邻居。

**方法：**

1. 计算目标用户与*k*个聚类中心之间的相似性，获得1*k*的向量()。
2. 将目标用户分配到相似性最大的聚类中。
3. 搜索类别所属矩阵*Idx*(*m*,1)，得到与目标用户同类别的所有基本用户。
4. 使用公式计算目标用户与其他用户之间的相似性。
5. 将相似性最大的前*l*个基本用户视为目标用户的最近邻居。

通过算法3得到目标用户的最近邻居后，下一步可以直接根据公式()产生对目标用户的推荐。因为聚类后，同一类别中的用户远远小于全部用户的数量，所以可以提高在线推荐的效率。

* 1. 实验结果与分析
     1. 实验环境及实验数据

实验平台使用Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz，4G内存，操作系统为Windows 7 旗舰版，所有程序均采用Matlab实现。

本次实验，我们采用美国Minnesota大学GroupLens项目组提供的数据集MovieLens（[http://www.grouplens.org/](http://movielens.umn.edu/)）。MovieLens是一个基于Web的研究型推荐系统，用于接收用户对电影的评分并提供相应的电影推荐列表。目前，该Web站点的用户已经超过43000人，可供用户评分的电影超过3500部。本次实验我们选取的是100k的数据集，该数据集中包含了943个用户对1682部电影的100 000条评分数据，其中每个用户至少对20部电影进行了评分。评分范围从1~5，表示用户对电影的喜爱程度，1表示最不喜欢，5表示最喜欢。

GroupLens项目组提供全部MovieLens数据集的同时，将其分为5个互不相交的子集，每次实验选择一对base数据集和test数据集，使用base数据集中的用户作为基本用户，对test数据集中的目标用户产生推荐测试。为了度量整个数据集的稀疏性，本文引入了稀疏度的概念，其定义了用户已评价的数据占整个数据集的比例。本文所用数据集的稀疏度为1-100000/（943\*1682）=93.7%，可见数据集相当稀疏。

* + 1. 度量标准

评价推荐系统质量的度量标准主要包括统计精度度量方法和决策支持精度度量方法[18]。统计精度度量方法中常用的是平均绝对偏差MAE（Mean Absolute Error）；决策支持精度度量方法中主要有召回率（Recall）、准确率（Precision）及ROC（Receiver Operating Characteristic）等三种方法。

* + 1. 实验结果分析

实验1：传统协同过滤推荐算法相似性计算公式有三种：余弦相似性、Pearson相关系数、修正的余弦相似性。三个相似性算法对比实验。（横坐标-邻居个数5-40，纵坐标-MAE）。找出最合适的相似性公式和最佳邻居个数。

实验2：邻居数目确定，改变共同评分项目阈值（横坐标-共同评分项目阈值，纵坐标-MAE）

实验2：使用K-means聚类，确定邻居数目，比较不同聚类数目下的推荐效果。（横坐标-聚类数目，纵坐标-MAE）

1. 电影推荐系统设计与实现
   1. 系统需求分析
   2. 系统设计
      1. 系统架构设计
      2. 系统功能模块设计
      3. 系统数据库设计
   3. 推荐系统实现
      1. 系统实现环境
      2. 系统功能实现
   4. 本章小结
2. 结论与展望
   1. 本文总结
   2. 工作展望
3. 参考文献
4. 中国互联网络发展状况调查统计报告[R]. 北京:中国互联网络信息中心, 2017.
5. 蔺丰奇, 刘益. 信息过载问题研究述评[J]. 情报理论与实践, 2007, 30(5):710-714.
6. 粱伟萍. 浅谈电子商务中的个性化推荐系统[J]. 网络与信息, 2011, 25(8):38-38.
7. 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009, 19(1):1-15.
8. Abbattista F, Degemmis M, Licchelli O, et al. Improving the usability of an e-commerce web site through personalization[J]. Recommendation and Personalization in eCommerce, 2002, 2: 20-29.
9. 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.
10. Yu K, Xu X, Ester M, et al. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach[J]. Knowledge and Information Systems, 2003, 5(2): 201-224.
11. Kuo R J, Liao J L, Tu C. Integration of ART2 neural network and genetic K-means algorithm for analyzing Web browsing paths in electronic commerce[J]. Decision Support Systems, 2005, 40(2): 355-374.
12. 周军锋, 汤显, 郭景峰. 一种优化的协同过滤推荐算法[J]. 计算机研究与发展, 2004, 41(10): 1842-1847.
13. 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).
14. 邓爱林, 左子叶, 朱扬勇. 基于项目聚类的协同过滤推荐算法[J]. 小型微型计算机系统, 2004, 25(9): 1665-1670.
15. 林鸿飞, 杨志豪, 赵晶. 基于内容和合作模式的信息推荐机制[J]. 中文信息学报, 2005, 19(1): 49-56.
16. 崔林, 宋瀚涛, 陆玉昌. 基于语义相似性的资源协同过滤技术研究[J]. 北京理工大学学報, 2005, 25(5): 402-405.
17. 陈冬林, 聂规划, 刘平峰. 基于网页语义相似性的商品隐性评分算法[J]. 系统工程理论与实践, 2006, 26(11): 98-102.
18. 孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 浙江: 浙江大学, 2005.
19. 曹渝昆. 基于神经网络和模糊逻辑的智能推荐系统研究[D]. 重庆: 重庆大学, 2006.
20. 高凤荣, 马文峰, 王珊. 数字图书馆个性化信息推荐系统研究[J]. 情报理论与实践, 2003, 26(4): 359-362.
21. 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.
22. 马宏伟, 张光卫, 李鹏. 协同过滤推荐算法综述[J]. 小型微型计算机系统, 2009 (7): 1282-1288.
23. 邱爽, 葛万成, 汪亮友, 等. 个性化推荐中基于用户协同过滤算法的优化[J]. 信息技术, 2016, 40(3): 62-63.

[21] <https://my.oschina.net/liangtee/blog/125407>

[22] 韩家炜, 坎伯. 数据挖掘: 概念与技术[J]. 北京: 机械工业出版社, 2001, 100: 103.

[23] 夏培勇. 个性化推荐技术中的协同过滤算法研究[D]. 青岛: 中国海洋大学, 2011.

致谢

时光荏苒，岁月如梭，一转眼间两年的研究生生活接近了尾声，这也同样意味着我的学生生涯即将结束。

感谢我的爸爸妈妈，谢谢你们一直在我背后默默的支持着我，谢谢你们为我的付出。这十多年来，为了能够让我们兄妹三人顺利完成学业，你们每天起早贪黑，辛苦的劳动，如今，我终于要毕业了，你们也可以放心了。

最后，向参加本论文评阅、答辩,并提出宝贵意见的老师致以诚挚的谢意!