学校代号  学 号

分 类 号 密 级

****

硕士学位论文

基于蝴蝶优化算法与聚类协同过滤电影推荐系统的研究与实现

学位申请人姓名 何雄

培 养 单 位 信息科学与工程学院

导师姓名及职称 戴牧红教授

学 科 专 业 软件工程

研 究 方 向 推荐系统

论文提交日期 2022年 2月10日

**The Research and Implementation Of Collaborative Filtering film Recommendation System Based On Butterfly Optimization Algorithm And Clustering**

**By**

**He Xiong**

## B.E.(Hunan University)2022

**A thesis submitted in partial satisfaction of the**

**requirements for the degree of**

**Master of Engineering**

**in**

**Information science and Engineering**

**in the**

**Graduate school**

**of**

**Hunan University**

**Supervisor**

**Professor Dai MuHong**

**March,2022**

湖　南　大　学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。 除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的 成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完 全意识到本声明的法律后果由本人承担。 作者签名： 日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有 关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以 将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复 制手段保存和汇编本学位论文。

本学位论文属于

１．保密□ ，在 年解密后适用本授权书。

2．不保密□。

(请在以上相应方框内打”√”)

作者签名： 日期： 年 月 日

导师签名： 日期： 年 月 日

Ⅰ

**摘 要**

进入大数据时代，数据呈现指数级增长。如何在繁杂的数据中，获取用户感兴趣的内容成为当前研究热点。在电影视频领域亦是如此，随着研究的深入，协同过滤推荐技术日益推广，成为应用最广的推荐技术。目前研究中方向集中体现在推荐质量和扩展性问题两个方面。

本文中，将K-means聚类算法与协同过滤算法想结合。改善了扩展性的问题的问题。针对K-means初始中心点敏感的的问题。将较新的群智能算法--蝴蝶优化算法与K-means算法相结合。缓解了初始化聚类随机带来的陷入局部最优问题。。对比其他相关推荐算法，结果表明，该算法在推荐准确度上有较大提高。

在验证文中推荐算法有效地基础上，将聚类协同过滤算法应用于电影系统中。该电影系统由信息管理模块、推荐模块、查询模块、数据收集模块及离线计算模块，通过应用文中所述算法，子模块实现了基于用户用户评分的协同过滤推荐。

关键词：协同过滤；聚类；群智能算法；电影推荐系统；

Ⅱ

**Abstract**

Entering the era of big data, all kinds of data show exponential growth.

**Key Words：** Collaborative filtering; Clustering; Swarm intelligence algorithm; Movie recommendation system.

Ⅲ

**目 录**

学位论文原创性声明和学位论文版权使用授权书………Ⅰ

摘要………………………………………………………Ⅱ

**Abstract**…………………………………………………Ⅲ

插图索引……………………………………………………Ⅶ

附表索引……………………………………………………Ⅷ

**第1章 绪论**…………………………………………………1

1.1 研究背景与意义……………………………………2

1.1.1 气相色谱法最新进展…………………………2

1.2 国内外研究现状……………………………………3

**第2章 相关理论基础**………………………………………1

2.1 聚类算法……………………………………2

2.1.1 聚类分析概念…………………………2

2.1.2 聚类算法要求…………………………2

2.1.3 聚类算法………………………………2

2.1.4 聚类评价指标…………………………2

**结论**…………………………………………………………61

**参考文献**……………………………………………………62

**附录A**（攻读学位期间所发表的学术论文目录）………63

致谢…………………………………………………………64

Ⅵ

**插图索引**

图1.1 气相色谱装置框图…………………………………2

Ⅶ

**附表索引**

表2.1 柱温与样品沸点的关系…………………………… 15

表2.2 9.26氧化液及其还原产物的色谱数据…………… 32

Ⅷ

**第1章　绪 论**

**1.1 研究背景和意义**

中国互联网信息中心（CNNIC）发布的<<第46次中国互联网网络发展状况统计报告>> [1] 显示，截止2020年6月，我国网民规模达9.40亿，较2020年3月增长3625万，互联网普及率达67.0%，较2020年3月提升2.5个百分点。我国手机网民规模达9.32亿，较2020年3月增长3546万，网民使用手机上网的比例达99.2%，较2020年3月基本持平。

为了给这些互联网用户群体提供资讯服务，人们随处可见到各种各样的应用，根据上文的CNNIC报告显示，截止2020年6月底，我们国内市场上检测到的APP数量359万款，由此带来的网络数据的规模也日益膨胀。希捷科技赞助、国际数据公司（IDC）发布了《数据时代2025》[1]白皮书。白皮书显示，未来数据增长速度惊人，全球数据领域将从 2019 年的 45 ZB 增长到 2025 年的 175 ZB。数据的来源以及由数据飙升引起的应用趋势也会产生变化，这是数据未来的大势所趋。从数据中挖掘隐藏的巨大的价值，也必将促进社会的进步。

目前的社交、视频、新闻以及电子商务等互联网系统都是采用推荐系统给用户提供更好的体验。推荐系统的核心及充分考虑用户个性化的需求，即结合用户行为偏好、历史数据等，使用合适算法从海量数据中，将用户喜爱的内容推送给到用户，达到“千人千面”的目的。它能有效的提升用户忠诚度、粘性、增加流量，并提供满意度。同时也能够更深入了解用户，便于进行更多产品形态的开发和变现。

**1.2 国内外研究现状**

自从上个世纪90年代起，“推荐系统”被首先提出，最早是Tapestry首次提出协同过滤，解决电子邮件分类过滤问题。而其后协同过滤作为推荐系统的重要技术开始发展开来。其后，亚马逊等企业纷纷引入该算法，也就将推荐系统在多个领域开枝散叶。经过了经过了二十多年的沉淀与发展，推荐系统取得了瞩目的成果。

基于协同过滤的推荐。一般有基于用户的协同过滤和基于物品的协同过滤。不需要预先获取用户和新闻的特征数据，仅仅依赖于用户的历史行为数据对用户建模分析，从而给出推荐。具体是：根据用户－项目评分矩阵，寻找与目标用户（待推荐用户）有相似偏好的其他用户，即为目标用户的最近邻居，根据最近邻居进行评分预测，形成推荐。

但是在整个用户空间搜索目标用户的最近邻居非常耗时，并且存在选择邻居不合理的情况，进而导致推荐质量和推荐效率不高。于是基于聚类的协同过滤改进算法被提出。聚类技术的引入，使得在为了寻找近邻计算相似度的过程中，不用比较全局用户而只需比较同一个簇内的用户，减少了寻找相似用户时间。且聚类得到的同一个簇中对象具有较高的相似度。可以降低算法在计算相似度时的复杂性，并且增加系统的可扩展性。近些年许多学者对基于聚类的协同过滤算法提出了改进方案，Wang等人[3]设计了启发式聚类模型，根据数据集的特性为目标用户划分邻居，提高了推荐效率，仿真实验证明了这一算法的可行性。Li等人[4]提出了一种类别偏好Canopy-K-means聚类的协同过滤推荐算法，设计类别偏好比率，改善了推荐效率。

研究显示群智能算法能改善中心点敏感问题，本文将对一种新型群智能算法做优化，并基于聚类算法完成推荐系统。

**1.3 论文主要内容和组织结构**

论文分为以下五个章节：

第一章为绪论，主要介绍群智能算法、聚类算法和推荐系统的研究背景和意义，以及国内外的研究成果。

第二章主要介绍协同过滤算法，聚类算法中的 K-means 聚类和群智能算法中的蝴蝶优化算法的相关理论基础。

第三章主要介绍对蝴蝶优化算法的改进和加入改进蝴蝶优化算法的 K-means 聚类算法 EEBK-means。

第四章主要是设计部分，通过需求分析，搭建了整个系统，融入了我们设计的推荐算法。

第五章主要是总结和展望部分，总结文中不足，并探讨今后工作进一步研究方向。

1. **相关理论基础**

本章主要介绍目前推荐算法的种类和评价指标，详细分析了电影推荐系统需要的协同过滤算法和聚类算法做了介绍，为后文设计和实现中改进算法做前期铺垫。

**2.1 推荐算法**

**2.1.1 基于协同过滤的推荐算法**

协同过滤推荐算法主要是利用用户历史行为如评分等，根据用户信息与推荐物品信息进行分析，依据评分矩阵，完成推荐。依据此为基础对用户进行推荐。我们对基于用户的协同过滤算法做介绍。算法分为一下步骤：

1. 根据用户历史行为信息构建用户-项目评分矩阵（以单一评分为例）

假设对项目进行评分的n个用户id集合为，和m个项目的id集合为。用表示用户-项目评分矩阵，如表 2.1 所示。使用元素, 来表示用户对项目的评分，其范围为0到5之间的整数。评分值由小到大，评分越高表示兴趣越大。

表3 用户评分矩阵

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 2 | 5 | 3 |
|  | 0 | 4 | 0 |
|  | 1 | 2 | 1 |

1. 用户相似度的计算

推荐算法中，需要计算用户间的相似程度，这时候就需要用到相似度公式来获取用户最近邻集合。在电影推荐算法中有很多的相似度公式，本文使用的 Pearson相似度[37]是根据目标用户和其近邻用户共同评分的项目集合进行计算。计算公式如下：

其中，表示用户间相似度，和分别表示用户和用户对所有项目评分的平均值，表示用户对项目ｉ的评分。。

1. 评分预测推荐

得到用户间的相似度后，根据相似性从高到底排序，选取K个目标用户组成最近邻集合，从而进行评分预测，将需要推荐的项目进行比较，将预测值最高的电影项目推荐给用户。预测公式如下：

其中，表示用户的预测评分, 表示用户的最近邻集。

**2.1.2 基于模型的电影推荐算法**

基于模型的电影推荐算法并不依赖初始数据，而是通过建立模型来对目标用户进行推荐。其中的模型原理依托于各种机器学习的算法模型，对数据进行离线训练，为用户在线进行推荐。下面介绍几种常见的基于模型的推荐算法及其算法思想可以归类概括为如下：

（1）关联规则推荐算法模型

模型首先在用户项目数据中，找到频繁项目集，对频繁集项目进行计算，找到频繁 N 项集。如果用户购买的项目在频繁项集中，那么我们将对频繁项集中其他未购买的项目按照支持低于、置信度和提升度等指标进行评分预测，最终推荐给用户。

1. 聚类推荐算法模型

基于聚类的推荐算法和上一节中所讲的算法较类似。首先按照距离公式计算用户或者项目的距离进行聚类。基于用户聚类的话，在推荐之前，首先对用户进行聚类，来减少数据维度过大导致的时空消耗，之后对用户完推荐。基于项目的聚类则是将用户喜欢的项目分类，完成推荐。

（3）矩阵分解推荐算法模型

模型先将矩阵进行分解，然后对空白数据进行补足，以达到预测评分的效果。矩阵补全常使用的方法是 QR 分解、SVD 分解及其改进算法等。这些算法可以提高矩阵分解在稀疏矩阵下的推荐准确度。

**2.1.3 推荐算法评测方法指标**

推荐系统评估[47]是检验一个推荐系统推荐结果好坏的标准。针对不同的推荐系统，不同的数据集和不同的评估方式都不同，评估方法有两种：（1）用户在线评测方法一般是通过平台将推荐结果推送给用户，然后通过用户反馈的信息，去判断观察用户是否真正喜欢并产生交互行为，最终判断算法的好坏。（2）离线评估方法则不需要用户实时数据，将用户数据收集下来，在后台进行训练学习，得到结果后在下次用户上线时进行推荐，通过推荐的准确度来判断一个算法的好坏。推荐系统评测指标如图 2.3 所示。

下面给出几种推荐系统的评测指标：

（1）准确率（Precision）

准确率反应的是用户对推荐结果的满意程度。设项目个数为N，项目集为R(u)，用户标记为满意的项目集为T(u)，计算公式如 2-(3)所示：

（2）召回率（Recall）

召回率又称为查全率，它是指推荐系统最终输出给用户的推荐列表中包含多少用户已经评分过的项目，计算公式如 2-(4)所示：

（3）覆盖率（Coverage）

该指标计算的是推荐算法中产生的推荐列表中项目个数占总项目个数I的比例，

计算公式如 2-(5)所示：

（4）均方根误差（Root mean square error，RMSE）

RMSE 评估指标首先将数据集分成两部分，一部分用来训练，一部分用来测试。对训练集中数据进行训练，获得预测评分，在训练模型上预测得到的评分用表示，真实评分用表示，用户-项目评分用表示。通过计算两者的误差，来监测推荐结果的好坏。计算公式如 2-(6)所示：

（5）平均绝对误差（Mean absolute error，MAE）

很多学者都会选择 MAE 指标来评估自己的推荐系统，MAE 反映的是训练集和测试集的平均误差结果。在实验中，测试集中商品数量用N表示，计算公式如 2-(7)所示：

**2.1 聚类推荐算法**

聚类的核心思想是指按照某种指标将相近的对象聚集到不同的类中的过程，这种划分过程是无监督的，特征相似的对象会被分到同一类群，相差较大的就被分到其他类群。聚类算法的算法过程是：设定初始聚类的簇数和中心点，计算数据与中心点的距离，将与中心距离近或相似度高的样本分到一个聚类簇里，最终划分为几个簇。因此，推荐系统对于搜寻目标用户最近邻之前可以利用聚类技术，将用户按照特征属性的不同进行分簇。

目前聚类里面对计算样本特征进行分类使用最多方法是选择测量数据样本间的距离。对于样本不同的属性特征，就用相对应之方法进行相似度的计算。因为通常判断相似程度取决于样本间的距离。其中，表示一个数据样本，指代两个样本和间的距离，𝑑 维向量中每个样本的特征为 =( , , ⋯ , )(𝑘 = 1,2, ⋯ , 𝑑)指的是𝑑个属性特征的具体值。常用的计算距离方式[55]有：曼哈顿距离、欧氏距离以及明考斯基距离。

**2.1.1 聚类算法分类**

作为机器学习中的分类模型，聚类算法因其算法原理简单、适应性强，算法不断被研究者们的优化更新，按照聚类中计算的步骤以及底层设计的不同，大致分为以下三类[55]，我们对其进行介绍并给出其算法原理。

（1）基于划分的聚类方法

该方法是利用事先规定的特征属性将数据集中的数据对象按照距离或其他规则划分到不同的簇类中。首先在算法开始之前，首先需要对聚类簇的数目聚类的中心点进行初始化，之后算法根据规则将每个数据依次划分到对应的簇类，直到算法满足终止条件。聚类算法 K-means 算法的算法原理是在算法开始之前需要对聚类中心点和聚类簇数进行初始化，对每个数据进行分簇，每次迭代都重新选取簇类中的中心点。K-means 算法一般选取欧氏距离作为目标函数。K-means 算法在开始执行时我们无法知道数据大致有哪些类，甚至不知道会被划分为哪一类，所以 k 值的选择就非常关键；其次聚类算法的初始中心点是算法开始时随机初始化得到的，所以聚类效果的好坏将直接由中心点的选取所决定。

（2）基于密度的聚类方法

由于基于划分的聚类方法多数以距离作为划分簇类的依据，所以聚类后的结果多数以球形居多。而复杂的数据集往往需要聚类形成各种各样不同的形状，为满足不同的要求，基于密度的聚类应运而生。该种聚类方法的原理是按照数据点的密集程度作为划分簇类的依据，比较常见的是 DBSCAN 算法。算法在聚类前，不需要特意指定簇数，而是将邻域半径内密度达到某一阈值的区域划分成一簇，噪声点会形成特殊形状的簇，更方便区分。算法需要提前预设邻域半径𝑟和密度阈值𝜌。

（3）基于层次的聚类方法

基于层次的聚类方法是计算数据属性的相似度作为划分依据来对数据集分簇。层次聚类可以分为两种：一种是基于凝聚的聚类，每个数据都是单独一类，计算类与类间的相似度，将高相似度的簇合并，直至归为一类，是自下而上的。另一种是基于分裂的聚类，他是自上而下的，首先计算簇内数据的相似度，判断簇内中相似度的差异，若差异过大，则进行簇类分裂产生两个小簇类，直至达到最大迭代次数或者所有的簇类中都只剩唯一数据时结束。基于凝聚的聚类算法 AGNES 算法适用比较广，不局限于数据本身的形状。但是若遇到的数据量太大，则会消耗更多的时间，且在聚类过程中合并过程不可逆，因此比较局限。

**2.1.4 聚类评价指标**

聚类评价指标是用来评价聚类效果的好坏，也是聚类算法的好坏的重要依据[29]。而聚类算法的目的是把数据对象划分为一个个簇类，同一簇内元素之间的属性相似度越高，不同簇间元素之间的相似度越小，则聚类效果越好。基于这一思想，研究人员引入了多种聚类评价指标的计算方式。另外由于目标函数也是算法

在聚类过程中的聚类效果好坏的体现，因此我们可以将目标函数设定为聚类评价指标。

常用的内部评价指标轮廓指数（SI）、邓恩指数（DVI）和戴维森堡丁指数（DBI）。

（1）SI 主要测量每个簇自身距离和其它最近簇的距离比值：

（2-1）

SI(i) = （2-2）

其中  （2-3）

（2-4）

dis分别表示簇类和 任意两个不同簇类数据的平均距离和簇类中任意两个数据的平均距离。SI 指数越高说明同一簇类内数据对象的距离越小，不同簇类间数据对象的距离越大，聚类效果也就越好，但是该指标的计算复杂度较高。

（2）DBI 主要测量任意两个簇类的类内平均距离之和与对应质心间距的比值最大值：

（2-5）

其中 （2-6）

表示簇类的簇心。DBI 指数越小说明同一簇类内数据对象的距离越小，不同簇类间数据对象的距离越大，也就是说聚类效果越好。但是环状分布的簇类得到的该指标数值上会相对较大。

（3）DVI 主要测量任意两个簇之间的类间最短距离和任意簇的类内最大距离的比值：

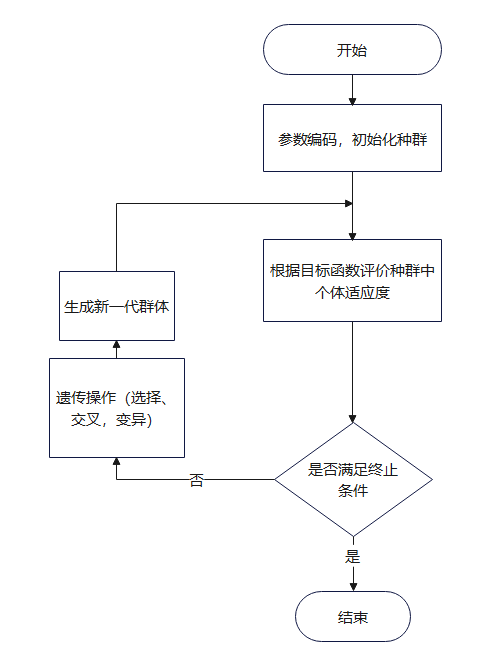
（2-7）

DVI 值越大说明同一簇类内数据对象的距离越小，不同簇类间数据对象的距离越大，也就说明聚类效果越好。但是具有离散类的簇类结果计算得到的该指标数值会相对较大。

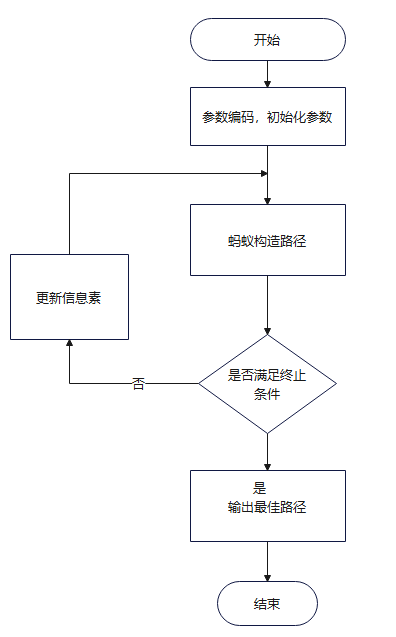
**2.2 群智能算法**

群体智能是研究人员通过观察和模拟群居生物的群体行为提出的一些用于解决传统问题和实际应用问题的智能计算或优化方法[6,31]。人类在大自然中的生物种群之间体现出的智能行为下得到了启迪，以此提出了一种新的人工智能方式，即为群体智能；也就是群体中的多个简单个体之间可以通过相互合作做出一系列更加复杂的群体行为表现。这个由很多主体组成的群体被称之为智能群体的前提是群体必须能够体现出学习性、自主性和自适应性等复杂的行为特征。遗传算法（GA）、粒子群优化（PSO）算法和蚁群（ACO）算法就是目前学术界常用的群体智能算法。

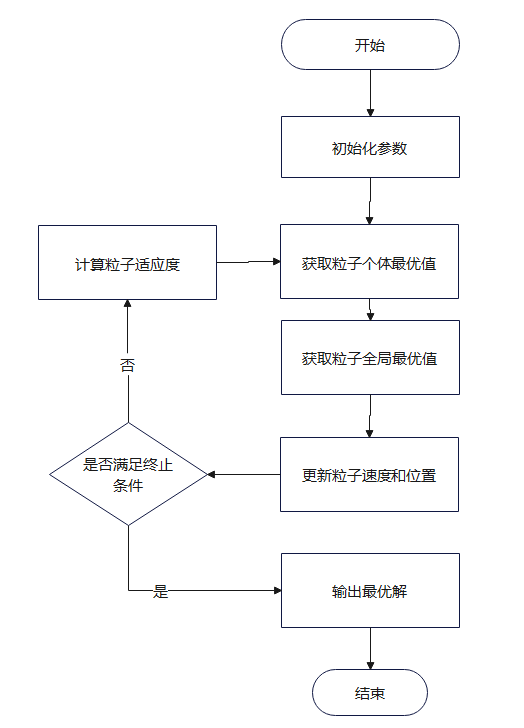
**2.2.1 遗传算法**

遗传算法（GA）是 20 世纪 60 年代由美国学者 J.H.Holland 提出的一种模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型[32]。它通过模拟大自然中生物群体优胜劣汰的演变过程不断优化问题解，并逐渐逼近最优解。在问题解集搜索过程中，算法具有较强的鲁棒性和适应性。该算法属于启发式算法，算法对于问题的优化过程是来自于生物遗传物质的操作包括三种基本的遗传算子：选择、交叉和变异算子。选择算子，算法根据个体的适应值去判断个体的好坏，然后根据选择算子择优选取种群中的个体。交叉算子，是模仿自然界的生物交配形成新个体的行为过程。保持物种的多样性。变异算子，随着自然环境的不断变化，自然界中的生物群体会发生一定概率的基因突变来适应变化的环境。算法根据变异因子使每个个体都有一定的概率发生突变，加速种群的收敛速度。遗传算法易与其它算法结合，但是算法涉及参数较多，且过于依赖初始种群。

（2）蚁群算法

蚁群优化算法[52]在发表后就应用在了数学领域中。其原理是来自蚂蚁组成的群体觅食行为，每只蚂蚁都会分泌一种信息物质，并且释放在路过的路径上供其他蚂蚁辨识。当其余蚂蚁进行路径选择时就会被这些信息物质所吸引，再通过识别每条路径上信息物质的浓度来选择最优行径方向。所以就会形成一种正反馈，信息物质越多的路径，会吸引越多的蚂蚁，然后留下更多信息物质，逐渐这条路径上的信息物质浓度将会更高，随着信息物质的挥发，会有更过的蚂蚁跟从。这种反馈机制就会帮助蚁群在面临多条路径的时候，通过识别信息物质的变化，最终找到一条最短最优路径。简化这种行为过程：离食物更短的路径上，会有更多的蚂蚁经过，则留下信息物质的浓度就会变高，会吸引其他蚂蚁选择该路径，随之形成正反馈过程，就会找到最佳路径。

3）粒子群算法

粒子群优化算法[53]是在一个区域内，有一群不知道食物的位置的鸟在随机地搜寻食物。鸟群会根据自身的适应值，和其他鸟交换信息，来分析当前的自身位置是不是离食物更近，并且找到周围食物较多的鸟，通过搜寻其附近区域去快速有效地找到食物。算法是通过将鸟群中的每一只鸟作为一个粒子，粒子群中的每个粒子都拥有两种属性，速度属性决定粒子移动的快慢和移动的朝向，位置属性用粒子当前所在的坐标点表示。粒子搜索到的所有解中最优的个体极值即为粒子群的最优解。算法原理简单，使用操作起来比较方便等优点。但算法收敛时粒子都会向最优粒子移动，从而趋于一致损失掉复杂性。

**2.2.1 蝴蝶算法**

Sankalap Arora等人[5]于2018年提出蝴蝶优化算法(（Butterfly Optimization Algorithm，BOA）)，由于其简单性和有效性，在研究社区颇为流行，文中通过基准测试函数和三个经典问题，试验表明，该算法比其他元启发算法具有更高的效率。蝴蝶算法是一种元启发式仿生群智能优化算法，它模拟蝴蝶的觅食行为。

思想的条件假设如下:

1) 所有的蝴蝶都应该散发出某种香味, 使蝴蝶能够互相吸引。

2) 每只蝴蝶都会随机移动, 或朝着散发出更多香味的最佳蝴蝶移动。

3) 蝴蝶的刺激强度受目标函数值的影响或决定。

当蝴蝶能感觉到其他任何蝴蝶的香味时并朝它移动, 在该算法中, 该阶段称为全局搜索. 在另一种情况下, 当蝴蝶不能感觉周围的香味时, 然后它会随机移动这个阶段称为局部搜索. 利用转换概率控制全局和局部搜索过程, 其迭代公式为

式 (1)中的f是香味感知量, c是感觉形式,I是刺激强度,a是香味,通常 a和c的取值范围为[0, 1]之间;式(2)中的表示第i只蝴蝶在第t+1代的位置, 的随机数, 表示全局最优解, 表示第i只蝴蝶的香味感知量; 式(3)中的, , 分别表示第i,j,k只蝴蝶在第t代的位置;式(4)中的为常数, 表示第t代的值,Ngen 为最大迭代次数。

基于上述描述, 蝴蝶优化算法的实施步骤如下:

Step 1. 初始化种群规模n及转换概率p等参数。

Step 2. 利用式(5)计算每只蝴蝶的适应度值, 并求

出当前最优值和最优解 Best。

Step 3. 利用式(1)计算香味感知量, 如果 rand<p,

则利用式(2)计算,否则利用式(3)计算。

Step 4. 利用式(5)重新计算每只蝴蝶的适应度值

, if <, 则替换之前的最优值和最优解。

Step 5. 利用式(4)更新c

Step 6. 判断是否达到最大迭代次数, 如果是输出最优值和最优解, 否则跳至 Step 3。

**第3章　改进蝴蝶优化聚类算法**

**3.1蝴蝶优化算法的改进**

**3.1.1 指数收敛因子**

为了增强BOA算法的探索能力和提供收敛精度，引入线性收敛因子(Linear convergence factor)α在蝴蝶算法的全局位置更新处，为其设计的迭代前期α较大可以增强全局探索能力且递减速度快，而在迭代后期α收敛到较小值且递减速度变缓慢，以实现前期加速收敛，后期保证收敛精度在较高水平。Α随着迭代次数由2减小到0。文献收敛因子和黄金指引机制的蝴蝶优化算法，设计的公式是：

式中：t是当前迭代次数，是最大迭代次数。

本文引入一种新的基于指数变化的收敛因子(Exponential connective factor)更新方式。



**3.1.2 融入差分进化策略和精英策略**

差分进化是一类基于群体的自适应全局优化算法，与遗传算法类似，不过不采用二进制编码，而是实数编码。它的进化也包括变异，交叉和选择。

变异操作：在每次迭代中其变异过程是通过种群中选择3个不同的个体，通过下公式产生新个体：

其中，

交叉操作：

选择操作：经过变异和交叉生成的个体和目标个体通过，更新种群：

精英策略

在最优解附近产生符合正态分布的随机数，使用如下公式：

其中

基于指数收敛因子的蝴蝶算法融入差分进化策略（differential evolution）和精英(elite)策略。为了加强种群个体交互信息，做出了优化。

为了验证改进的指数收敛因子的有效性，实验首先进行了比较经典的蝴蝶优化算法，线性收敛因子蝴蝶算法和指数收敛因子蝴蝶优化算法在求解Sphere（单峰函数，公式）和Rastrigin函数（多峰函数）最小值问题过程中的优化结果。种群数量50，最大迭代次数500。动态转换概率0.8，C取0.01，α取0.1。结果如下，结果为算法独立运行30次取平均值。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | BOA算法 | LBOA算法 | EBOA算法 | EDEIBOA算法 |
| Sphere函数 | 0.026949 | 0.039431 | 1.255e-09 | 9.69674e-18 |
| Alpine函数 | 1.679137 | 0.839685 | 0.0001138 | 4.803655e-05 |

算法的改进算法的时间复杂度分析算法的种群规模设置为 N，最大迭代次数为 maxiter，搜索空间的维度为 dim，增加 差分进化策略，相当于增加一个内层循环( 由改进算法的流程图可以直接看出) ，则增加的运算量为 O( maxiter × dim × N) ，相当于增加了一个内层循环，根据改进算法的寻优效率来说，增加的计算量是可以接受的。

可以看到我们最终的最优的算法。相对于最初的基本蝴蝶算法。对于Sphere函数精度提升了16个数量级，对于Alpine函数精度也提升了5个数量级。

验证了EDEIBOA算法的有效性之后，我们在进一步验证与之结合的K-means 算法在不同数据集中的聚类结果表现。

|  |  |  |
| --- | --- | --- |
| 数据集 | 数据个数 | 属性个数 |
| Iris | 150 | 4 |
| wine | 178 | 13 |
| Absenteeism\_at\_work | 740 | 21 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | DBI指数 | K-means | BOAK | EK |
| Iris | 最优值 | 0.4042 | 0.3747 | 0.3747 |
| 最差值 | 1.1313 | 0.4648 | 0.3827 |
| 平均值 | 0.6671 | 0.3855 | 0.3782 |
| Wine | 最优值 | 0.5380 | 0.2959 | 0.3842 |
| 最差值 | 0.7181 | 0.5300 | 0.2724 |
| 平均值 | 0.6692 | 0.4014 | 0.3252 |
| Absenteeism\_at\_work | 最优值 | 1.0237 | 1.0602 | 0.8907 |
| 最差值 | 1.3118 | 1.2151 | 1.2043 |
| 平均值 | 1.1755 | 1.1786 | 1.0796 |

3.4.2 实验结果分析

为了比较本实验中方法的性能，在推荐指标的选择上，选择用 RMSE（均方根误差）

作为评价指标。RMSE 指标广泛应用于评估推荐系统中，具体来说，RMSE 提供了有关用

户选择的真实内容和预测的可能性内容之间的差异。RMSE 值越小表明推荐系统性能越好。计算公式如(3-7)：

其中𝑝,𝑖是用户𝑢对内容𝑖的预测评分，𝑟,𝑖表示实际评分，而𝑁表示对此内容的评分总分。在实验中我们将本文提出的基于降维和聚类的协同过滤算法与基于 K 均值算法的协同过滤算法和基于 K 近邻的协同过滤算法进行比较，邻居范围设定为 10-50。

实验结果

1. 确定最佳聚类个数C

为了确定聚类个数对于算法的影响，在不同的聚类个数C值下完成实验，将对应MAE的结果记录如下所示。

根据上图所示，当C为11时，此时的MAE取值最小，在其左右两侧，均呈现慢慢上升的趋势。可以认为，聚类个数为11时候，EEBK-CF算法有比较好的效果。

1. 确定最佳最近邻个数K

在确定聚类个数C为11的情况下，考虑最近邻个数选取对算法的影响。选取了两种协同过滤算法与本文算法进行对比，包括基于用户的协同过滤算法（UB-CF）和基于K-means的协同过滤算法以及包括本文的基于蝴蝶优化算法的协同过滤算法(EEBK-CF)。聚类个数C=11。实验结果如下图所示。

从图中不难看出，最近邻为30时，此时推荐效果较好，在此值两侧，推荐效果均下降。通过实验对比分析，本文算法本文的算法在电影推荐数据集中，MAE取值最小，推荐准确率最高。本文使用的蝴蝶算法与聚类协同过滤想结合，解决了中心点敏感问题，使得选择最近邻效果更好。

如表 3-10 所示。从表中可以看出本文提出的算法在三个数据集上的表现都不错，都要比另外两个基于传统的聚类算法的效果要好。

kmeans与传统的协同过滤算法UCF，基于K-means的协同过滤算法，

**第3章　基于蝴蝶优化聚类的协同过滤电影推荐系统**

结合上文推荐算法，本章将对基于蝴蝶优化聚类的协同过滤电影推荐系统进行设计和实现。行文先完成，功能需求分析，再对系统做功能模块设计，最终对数据库结构做设计，完善系统。

**3.1功能需求设计**

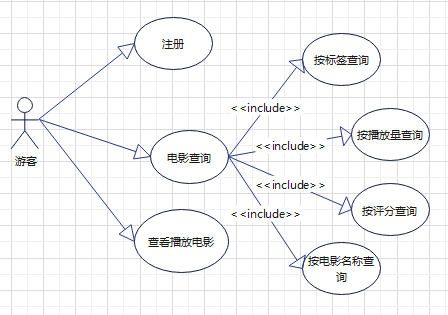
功能需求分析就是要明确该系统所含有功能。系统的使用对象不同，提供的功能也会

有所不同。根据系统的使用对象，对系统进行功能需求分析。

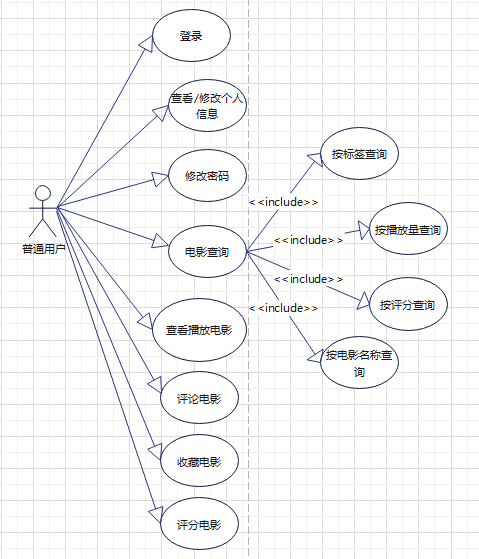
(1) 游客

我们将未登录的用户称之为游客。当游客使用该系统时，系统内并没有该用户的历史信息，所以系统对该类用户只会提供两种推荐列表，即热门电影列表和最新电影列表。此

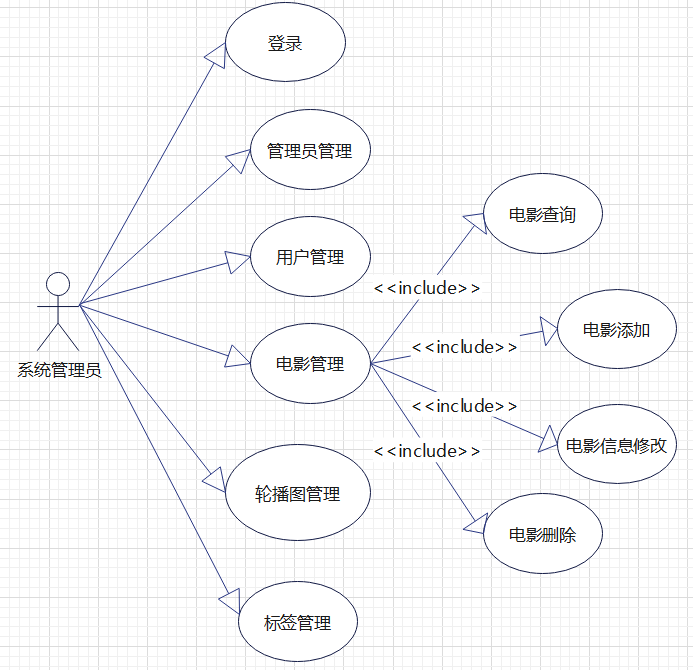
外，该用户还可以进行分类电影的查询以及相关电影信息的查看，但不能进行电影的观看

以及对电影的评论、评分等操作。 游客用例图如图 4.1。

(2) 普通用户

我们将已注册的用户称之为普通用户。普通用户可以使用已注册的账号、密码登录电影推荐系统，登陆成功后，系统会对该类用户提供三种推荐列表，即多样化电影列表、热门电影列表和最新电影列表。此外，用户还可以进行个人信息的查看以及修改，电影信息的查看以及相关电影的查询，进行电影的观看，并且可以对电影发表个人评论，用户还可以根据个人对电影的喜爱程度对电影进行打分，分数为 1-5 之间的整数，分数越高，表明该用户对此电影越喜欢。图 4.2 是普通用户的用例图。

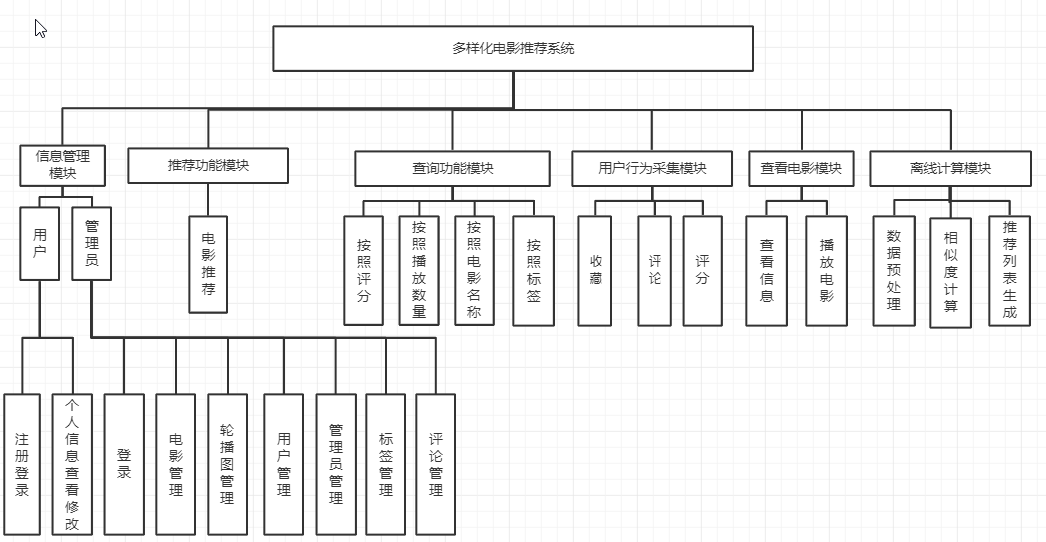
(3) 系统管理员

系统管理员是系统内不可缺少的角色。管理员登录系统，进入后台管理界面，其主要任务是电影信息的管理。对于新上映的电影，管理员需要收集相关影讯信息，主要是电影剧情介绍、导演、主要演员、上映日期、电影类型等信息，并将电影添加到数据库；对于数据库内已有的电影，管理员可以执行查看、修改、删除等操作。此外，管理员还拥有系统说明管理、系统前台轮播图管理等操作。图 4.3 是系统管理员的用例图。

**开发技术**

数据库设计采用Mysql数据库，Python作为后端语言，易于开发。后端框架采用Flask，开箱即用。使用MVC的开发模式，前端采用Bootstrap美化样式，采用JQuery简化Javascript编程。

**3.1功能模块设计**

功能模块设计根据功能需求分析，从而设计系统所有功能模块。

(1) 信息管理模块

信息管理模块包含用户信息管理和管理员信息管理两大部分。用户信息管理部分包含用户注册及登录，用户个人信息查看与修改。管理员信息管理部分则包括管理员的登录、，电影管理、用户管理，管理员管理，标签管理及评论管理管理等。

(2) 推荐功能模块

电影推荐系统的核心模块即为推荐功能模块，该模块通过对已登录用户历史电影评分记录的分析，挖掘出用户的兴趣爱好，进而对用户推荐多样化的 Top-N 电影列表。该模块也会根据电影的上映时间，向用户进行最新电影的推荐。此外，根据电影收到的评分情况，该模块会向用户提供热门电影的推荐。

(3) 查询功能模块

查询功能模块主要为用户提供根据标签，电影名称，标签和评分进行搜索的功能。该模块和推荐模块相合作，为用户提供更好的服务。

(4) 用户行为采集模块

采集用户对电影的行为，这里主要是用户可以评论和收藏电影。

(5) 查看电影模块

该模块主要实现查看和播放电影需求。用户可以查看某部电影的名称、片长、地区、评分、播放数量、简介、标签等信息，以及用户对该电影的评论。用户也可以选择播放电影。

(6) 离线计算模块

要实现推荐模块，则离不开离线计算模块。首先，离线计算模块会对原始数据进行预处理，然后挖掘用户相似项等操作。使用离线计算，可以很大程度上节省推荐花费的时间，提高推荐的效率。

3.4.2 数据库设计

本文的系统使用 Mysql 数据库保存信息，将收集到的用户信息和推荐算法生成的推荐结果都保存在数据库中。数据库中包括四张表来保存信息：User 表保存客户基本信息；Movie 表中保存影视内容基本信息；Ratings 表中保存影视内容属性及其评分等信息；Recommend 表中保存预测结果。

数据库设计

1. 用户信息表User表

用户信息表包括用户 id、用户姓名、用户密码、邮箱、手机号、简介、头像和用户评论外键，电影收藏外键

，如表 4-1 所示。

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | Bigint10) | 用户id,主键，自增 |
| Name | Varchar(100) | 用户名称 |
| Pwd | Varchar(100) | 密码 |
| Email | Varchar(100) | 邮箱 |
| Phone | Varchar(11) | 手机 |
| Info | Text | 简介 |
| Face | Varchar(255) | 头像 |
| Addtime | Datetime | 注册时间 |

电影信息表记录电影的所有属性，其表的设计如下表所示：

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | Bigint(10) | 电影id,主键，自增 |
| Title | Varchar(255) | 电影名称 |
| Url | Varchar(100) | 电影链接 |
| Info | Text | 简介 |
| Logo | Varchar(255) | 标志 |
| Star | SmallInt(6) | 评分 |
| Area | Varchar(255) | 地区 |
| Release\_time | Date | 上映时间 |
| Length | Varchar(100) | 影片长度 |
| Playnum | Bigint(20) | 播放量 |

管理员信息表，用户存储管理员的基本信息，其设计如下表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | Bigint(10) | 管理员id,主键，自增 |
| Name | Varchar(255) | 电影名称 |
| Pwd | Varchar(100) | 电影链接 |

标签表，用于存储系统电影的属于所有标签。其中标签Id为标签表主键，其设计如表所示

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | Bigint(10) | 标签id,主键，自增 |
| Name | Varchar(255) | 标签名称 |
| Addtime | Datetime | 添加时间 |

电影评论表，用户记录用户对电影的评价，包括用户Id和电影Id为外键。具体字段设计如下表所示。

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | int(11) | 评论id,主键，自增 |
| Content | Text | 内容 |
| Movie\_id | Int(11) | 电影Id |
| User\_id | Int(11) | 用户id |
| Addtime | Datetime | 添加时间 |

轮播图表，存储轮播图，其设计表如下所示。

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | int(11) | 轮播图id,主键，自增 |
| Title | Varchar(255) | 标题 |
| Logo | Int(11) | 轮播图 |
| Addtime | Datetime | 添加时间 |

用户评分表，其设计如下表所示。

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | int(11) | 评分Id,主键，自增 |
| Movie\_id | Int(11) | 电影Id |
| User\_id | Int(11) | 用户Id |
| star | Smallint(6) | 添加时间 |

电影收藏表，其设计如下表所示。

|  |  |  |
| --- | --- | --- |
| 字段名称 | 类型 | 注释 |
| Id | int(11) | 收藏Id,主键，自增 |
| Movie\_id | Int(11) | 电影Id |
| User\_id | Int(11) | 用户Id |
| Addtime | Datetime | 添加时间 |

4.5 系统功能实现

基于 3.2 部分对多样化推荐方法的介绍，以及本章 4.2 节详细的需求分析、4.4 节对系统的总体的设计，本小节对系统主要功能进行了实现，包括：用户注册、登录功能、用户个人信息管理功能、用户评论功能、用户搜索功能、管理员登录功能、管理员对电影信息的管理功能、管理员对类别添加功能以及管理员对前台轮播图管理功能和推荐功能等。

4.5.1 用户注册、登录功能的实现

用户登录界面是进入系统的唯一入口。对于没有账号的用户，可以通过注册界面填写注册信息，从而获得账号、密码，并在登陆界面输入用户账号以及密码，进而与数据库中信息进行匹配，若匹配成功，则完成登录操作，用户会进入前台系统主页，若匹配不成功，

则返回错误提示信息。用户注册、登录界面如图 4.9、4.10 所示。



4.5.2 管理员登录功能的实现

****数据库中存入管理员信息，管理员在登陆界面输入账号、密码，提交后会转到数据库进行管理员信息表的查询操作，若表中查询到的信息与管理员输入信息匹配成功，则进入系统后台主页，反之，系统会返回错误提示信息。管理员登陆界面如图 4.11 所示。

4.5.3 用户个人信息查看、修改功能的实现 、

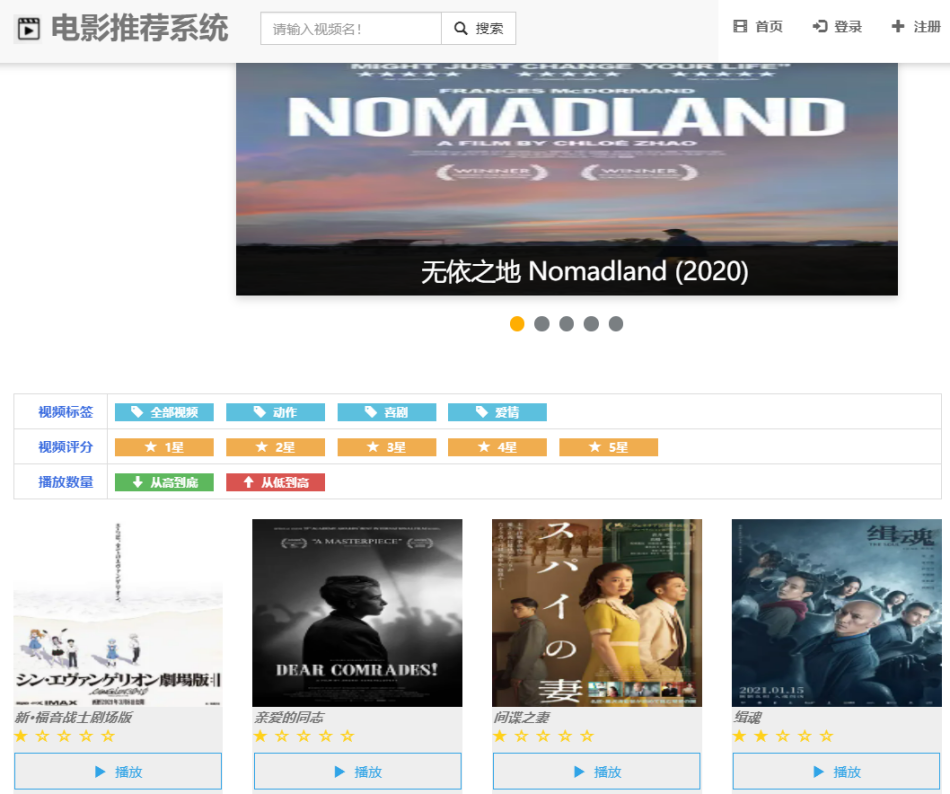
用户登录成功后，可以点击个人头像信息处，自由选择个人信息的查看或修改。在个人信息修改界面，用户可以对账号的密码、个人姓名、手机号、年龄、性别和个人头像，以及密保等信息进行修改。当修改完成并保存后，新信息会存储到数据库中，下次登录使用最新密码即可，其界面实现如图 4.12 所示。





4.5.4 用户搜索功能的实现

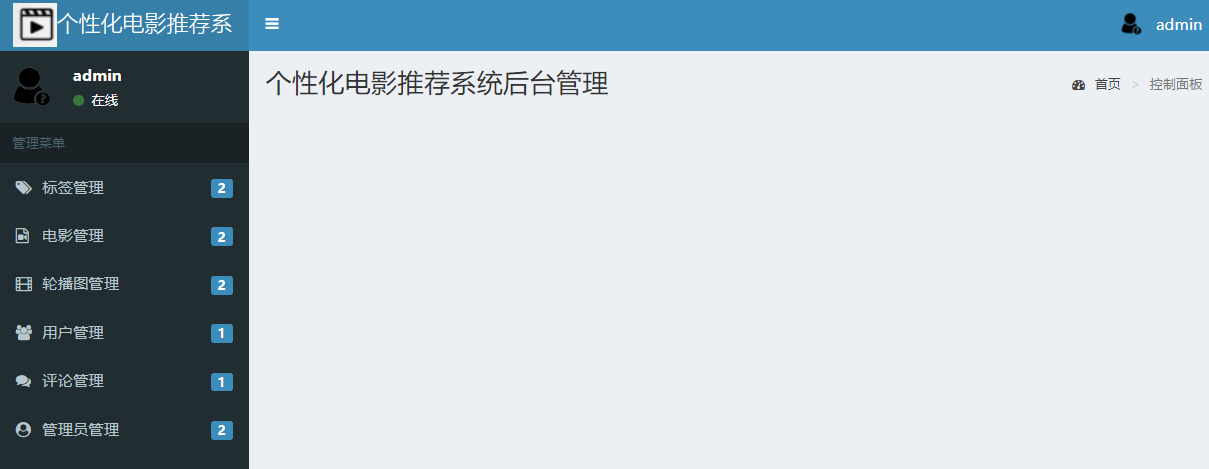
当用户想看某类型的电影时，可以在搜索框内输入电影类别，从而向用户返回符合该类型的电影列表以供其选择；当用户想观看某部特定电影时，可以根据电影的名称进行电影搜索，从而在该电影信息界面选择播放源进行电影的观看。其界面实现如图 4.13、4.14、4.15 所示。





4.5.5 管理员对电影信息管理功能的实现

管理员登陆系统后，首先会进入系统后台设置标签，其界面如图 4.16 所示。

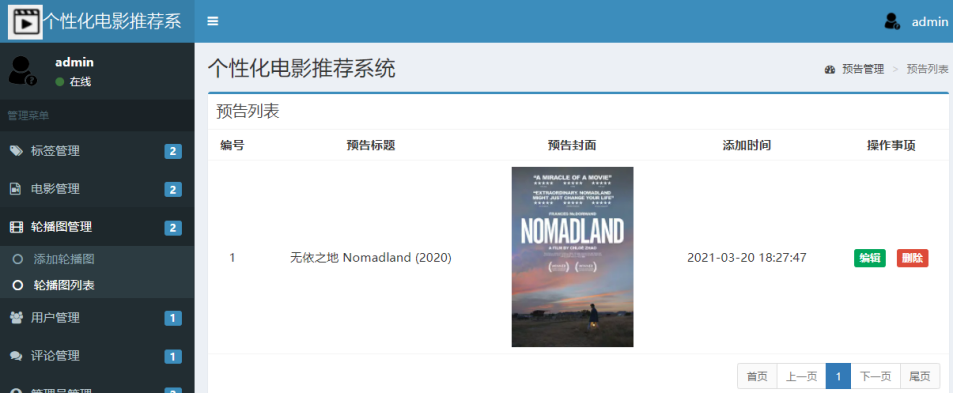


管理员可以后台做电影管理，添加，编辑和删除。



管理员对前端轮播图管理功能的实现

为了前端界面美化，在后台系统，可以对轮播图做管理，进行添加，编辑和删除

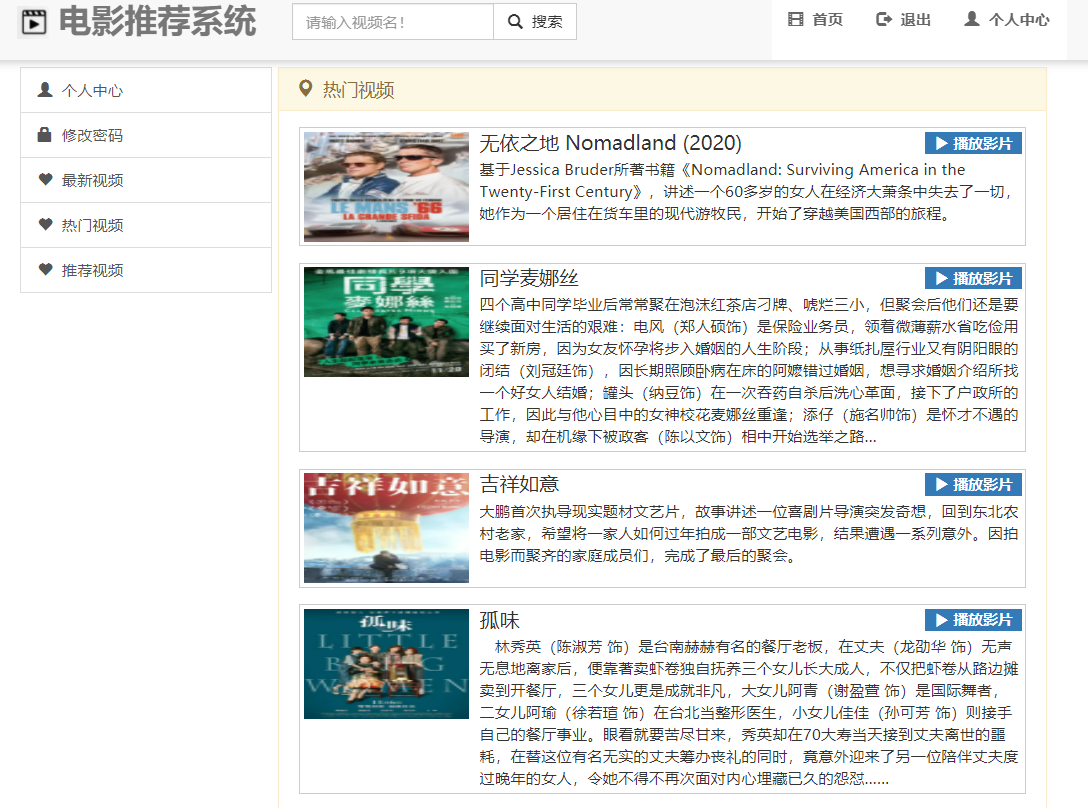
推荐功能的实现

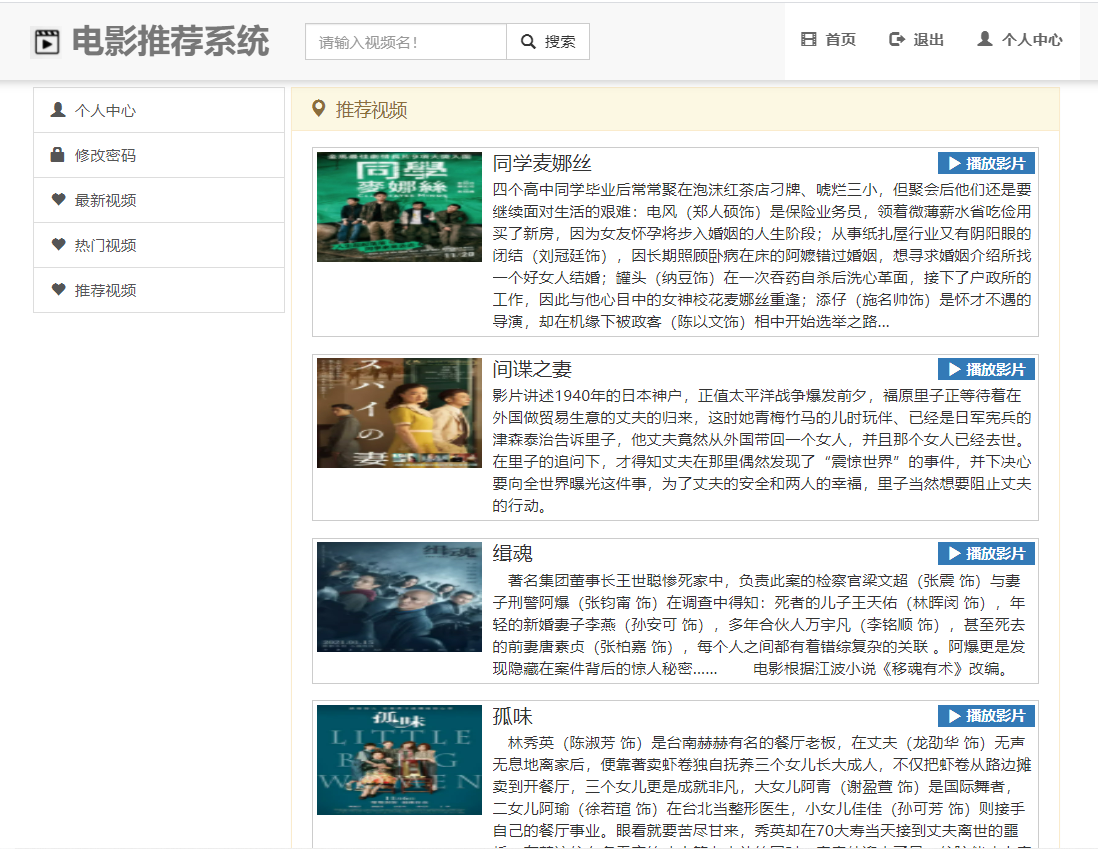
根据



**图 4.30 多样化电影推荐图**

**4.6**





**系统测试**

多样化电影推荐系统开发完成后，需要对其进行测试。功能测试是系统测试的重要环节之一。利用功能测试，可验证系统功能是否符合用例图的设计，其功能是否达到用户要求。文中我们选用的功能测试方法包括页面链接检查、相关性检查、search 检查等，对用户主要功能和管理员主要功能进行了测试。

表 4.7 是管理员主要功能的测试结果。

表 4.7 管理员功能测试结果

测试功能 测试次数 成功次数 失败次数 未运行次数 是否通过

登录验证 30 30 0 0 是

电影搜索 30 30 0 0 是

电影添加 40 40 0 0 是

电影删除 20 20 0 0 是

电影修改 10 10 0 0 是

类别添加 30 30 0 0 是

类别删除 20 20 0 0 是

轮播图添加 20 20 0 0 是

轮播图删除 20 20 0 0 是

表 4.8 是对用户主要功能的测试结果。

表 4.8 用户功能测试结果

测试功能 测试次数 成功次数 失败次数 未运行次数 是否通过

注册验证 30 30 0 0 是

登录验证 30 30 0 0 是

依据类别 查

询电影

20 20 0 0 是

依据片名 查

询电影

20 20 0 0 是

个人信息 修

改

10 10 0 0 是

电影评论 20 20 0 0 是

电影评分 30 30 0 0 是

多样化推 荐

列表查看

20 20 0 0 是

**第 5 章 总结与展望**

随着互联网的发展，网络中的信息呈爆炸式增长，这虽然可以满足用户的信息需求，但也带来了“信息过载”的问题。错综复杂的信息让用户眼花缭乱，以至于用户无法快速发现自己所需要的信息。对于这一问题，推荐技术是最为有效的解决方案。通过不同的推荐方法，推荐技术可以自动筛除掉一些垃圾信息，向用户推荐其感兴趣的信息，不仅提高了用户信息搜索的效率，而且丰富了用户体验。

协同过滤方法，凭借着“易解释、领域无关性”等特点成为了最为常用的一种推荐方法。然而，对于传统的协同过滤方法来说，在进行相似项或相似用户查找时常常需要大量的计算，这就会使得时间消耗增多，推荐效率较差，系统扩展性不高。与此同时，常用的协同推荐方法中一般只考虑推荐的准确度，并没有考虑推荐列表中项与项之间的相似性问题，由此带来推荐结果冗余化，使得用户满意度降低，从而影响用户对推荐系统的使用感。针对上述问题，本文用以下方法进行了解决：

(1) 为提高传统协同过滤方法低效率推荐以及冗余推荐问题，本文利用局部敏感哈希技术实现协同过滤，对数据库中的项进行项目索引值的计算，该过程离线即可完成。当系统需要查找与用户历史项目相似的项时，只需利用用户历史项目的索引值与数据库中其它项的索引值进行匹配即可，经过此过程，相似搜索的时间复杂度即可由 O(m\*n)降至 O(1)，系统可快速找到候选项，从而推荐效率也得到提高。在为用户生成推荐列表时，使用评分多样性对列表进行二次优化，除去列表内相似影片，在保证推荐准确度的情况下，尽可能为用户推荐多样的电影，使用户满意度有所提高。

(2) 为了验证上述两点的有效性，本文选用 MovieLens 数据集，通过设计对比实验，根据常用的推荐系统指标，对实验结果进行分析，证明了该方法是可行的。 (3) 基于 B/S 架构模式，结合上述研究成果，设计并实现了多样化电影推荐系统，通过对系统进行可行性分析、功能性需求与非功能性需求分析，以及对系统的总体的设计，使用Python 编言对系统的功能模块进行实现。 虽然本文提出的方法可以在保证推荐效率和推荐准确度的同时提高推荐多样性，且设计的多样化电影推荐系统能够独立运作。但是由于本人能力问题，改进的算法与实现的系统都有相应的不足之处，在未来的研究中需进一步弥补。总体来说，可以概括如下：

(1) 本文利用改进蝴蝶算法改进k-means聚类实现协同过滤，提供k-means聚类精度。

(2) 对于多样化方法，我们考虑的维度较为单一，并没有考虑电影的多个维度，例如导演、演员等，在之后的工作中，我们可以进一步考虑电影的其它维度，并考虑使用加权函数将电影的多个维度进行结合，以衡量电影间的相似性。此外，我们并没有考虑每个用户对推荐列表多样化的要求，比如有些用户可能并不太希望列表多样，这些都是我们以后工作中可以进一步完善的地方。 由于时间与能力有限，本文所设计实现的电影推荐系统，有些模块功能可能并不够完善，页面可能不够美观。除此之外，当用户量过大时，该系统可能面临服务器负载过大问题。在之后的研究中，需继续对系统进行完善。

**参 考 文 献**

[1] 赵晓，庞劲风，杨春光.浅析我国锦纶6帘子布生产现状.合成纤维工业，2001，24（1）：43-46

[2]  王德诚.我国轮胎帘子布供求现状与发展前景.中国化纤信息，1998，（9）：14-16

[3] 中国科学院大连化学物理研究所.气相色谱法. 北京：科学出版社，1972，44-46

[4] Crippen R C. Indentification of Oragnic Compounds With the Aid of Gas Chromatography. New York: Mc Graw Hill, 1993,36-39

1. Pecsok R L, Shieds L D, Cairns T, et al. Modern Methods of Chemical Analysis, New York: John Wiley & Sons, 1976,67-70

附录A 攻读学位期间所发表的学术论文目录

-6-

-10-

-8-

致 谢

时光飞逝，白驹过隙，转眼三年研究生生活即将告一段落。但这段学习的时光，我也在不断成长，我感受到这是一种财富。

感谢戴牡红老师， 在学业方面的指导，戴老师对我学习期间，给了我充分的自由，让我可以依照自己的兴趣，学习并做一些自己的研究，我的论文离不开他指导。也感谢学校可以给我们宽松的环境，能够快乐学习，感谢其他老师的付出，感谢同学的关爱。

也感谢一直默默支持我的家人。他们是我一直学习的坚强后盾，让我没有后顾之忧。他们的无私的爱让我感到力量满满。

最后，感谢所有评阅专家于百忙之中评阅此文。