# 武汉理工大学毕业设计（论文）

# 基于协同过滤算法的电影推荐系统研究与设计

学院（系）：计算机学院

专业班级：计算机 zy1701 班

学生姓名：廖文华

指导教师：***

# 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，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

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障、使用学位论文的规定，同意学校保留并向有关学位论文管理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。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，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、保密口，在　　年解密后适用本授权书

2、不保密口　。

（请在以上相应方框内打“√”）

作者签名：　　年　　月　　日

导师签名：　　年　　月　　日

# 摘　要

随着信息技术的发展，当前产生和需要处理的信息呈几何式增长。推荐系统致力于为用户从海量的数据中推荐出用户潜在的感兴趣的内容，其往往能带来比提供传统搜索引擎更好的用户体验。

影视领域是当前推荐系统应用和研究的热点，但是很多电影推荐系统很难兼顾推荐的准确性，多样性和时效性，且冷启动问题也常常没有得到解决。本文设计了一个个性化电影推荐系统，通过离线推荐和在线推荐相结合，基于协同过滤的推荐和基于内容的推荐相结合，不仅解决了冷启动问题，而且使得推荐系统具有推荐效果好，效率高，推荐具有实时性，多样性等特点。

本文综合协同过滤和基于内容的推荐构建出离线推荐系统，在此基础上，又提出了一种基于偏好队列的在线推荐算法（ORBPQ），ORBPQ 能够快速捕捉并反映用户行为，为用户近期行为给出相应实时性推荐，使得推荐具有时效性。此外，本文使用混合推荐算法综合调度在线推荐，离线推荐，基于协同过滤的推荐和基于内容的推荐，通过用户评分数量和电影被评分数量分别确定在线推荐与离线推荐、协同过滤与基于内容推荐的推荐比例，解决了用户冷启动、项目冷启动问题。

综上，本文所提出的个性化电影推荐系统综合应用了各类推荐算法，使推荐兼顾准确性，多样性与时效性，同时解决了冷启动问题。

**关键词：**电影推荐系统；协同过滤；基于内容的推荐；在线推荐；冷启动问题

# Abstract

With the development of information technology, nowadays, the information that is generated and needs to be processed has increased geometrically. The recommendation system is dedicated to recommending potentially interesting content for users from a large amount of data, and it can often bring a better user experience than traditional search engines.

The field of film and television video is a hot spot in the application and research of current recommendation systems. However, many movie recommendation systems are difficult to balance the accuracy, diversity and timeliness of recommendation, and the cold start problem is often not solved. This paper designs a personalized movie recommendation system. Through the combination of offline recommendation and online recommendation, collaborative filtering-based recommendation and content-based recommendation, it not only solves the cold start problem, but also makes the recommendation system with good recommendation effect and high efficiency, recommendations have the characteristics of timeliness and diversity.

This paper integrates collaborative filtering and content-based recommendation to construct an offline recommendation system. On this basis, an online recommendation algorithm based on preference queue (ORBPQ) is proposed. ORBPQ can quickly capture and reflect user behavior, and provide feedback for users' recent behavior. The corresponding real-time recommendation is made to make the recommendation time-sensitive. In addition, this paper uses a hybrid recommendation algorithm to comprehensively schedule online recommendation, offline recommendation, collaborative filtering-based recommendation and content-based recommendation. The number of user ratings and the number of movie ratings are used to determine the ratio of online recommendation to offline recommendation and collaborative filtering to content-based recommendation, respectively. The recommended ratio is to solve the user cold start and project cold start problems.

In summary, the personalized movie recommendation system proposed in this article comprehensively applies various recommendation algorithms to make recommendations take into account accuracy, diversity and timeliness, and at the same time solve the cold start problem.

**Key Words：** Movie recommendation system; collaborative filtering; content-based recommendation; online recommendation; cold start problem

# 目　录

# 第1章　绪论

## 1.1 选题目的和意义

随着信息技术的发展，当今所产生和所需处理的数据呈几何式增长。视频网站和电影评论网站所能提供给用户的视频内容也快速增加，为用户从海量的视频内容中推荐出用户潜在的感兴趣的内容往往能带来比提供传统搜索引擎更好的用户体验。推荐系统致力于从海量的数据中挖掘出少量对用户有用的内容，做到“特别的爱，给特别的你”，使系统呈现出传统模式所不具备的千人千面的效果，因此，个性化推荐系统变为上世纪 90 年代以来最受欢迎的应用之一[1]。

本课题致力于推荐用户可能感兴趣的电影，指导用户在海量的电影数据中寻找到自己所爱，由此可以提高用户寻找电影的效率，提高用户体验。同时，为用户做出合理的推荐也可以避免用户只能看到系统中少量热门电影，对冷门电影无法做到合理访问的情形，这样不仅充分利用了资源，同时提高用户对系统的忠诚度，对于平台运营方也大有裨益。

协同过滤是推荐系统中常用的效果较好的算法，对于解决信息过载问题具有较好的能力，可以从大量的数据中为用户提供合适的建议，使数据的价值充分得到应用。

## 1.2 推荐系统的国内外研究现状

当今在电影领域，个性化推荐算法已不可或缺，并在国内外应用实践中都得到了广泛的应用[2 4]，如 You Tube 推荐系统推荐影片，要经过候选池、排名池两层网络，筛选出合乎用户感兴趣的影片；爱奇艺推荐系统分为召回阶段和排序阶段，从海量的视频库中找到与用户兴趣相符的影片[5]。

协同过滤算法是当前推荐系统最为常用的算法，他的思想是基于群体智慧，根据用户对电影的打分记录，从中挖掘出电影的隐式特征及用户的隐式偏好情况，基于用户会喜欢与自己以前喜欢的电影相关的电影或喜欢与其相似的“邻居”用户所喜爱的电影[6,7]，由此可得用户对电影的预测评分，在此基础上再有选择的为用户推荐电影。

协同过滤的基本思路是基于用户对电影的共同历史评分，推荐用户可能喜欢的电影。矩阵分解是实现协同过滤的常见有效方法，其通过是预测评分矩阵和原始评分记录矩阵之间的差距最小化，得到电影的隐含特征向量，用户的隐含特征向量，并可预测出对电影的评分。其预测评分往往具有准确性较高的优势[8 11]。

协同过滤算法可分为基于用户的协同过滤和基于物品的协同过滤。基于用户的推荐算法基于用户会喜欢其相似用户喜欢的电影的假设，应用矩阵分解得到的用户隐含特征向量可以计算用户之间的相似度，得到一定数量的最相似用户，在此基础之上基于一定规则通过相似用户的数据得到为用户推荐的电影列表。

基于物品的协同过滤是基于用户的协同过滤算法基础之上的发展，其基于的假设是用户会喜欢其之前喜欢过的内容[6]，基本思路是通过寻找相似物品为用户做出推荐。

基于内容[12,13]的推荐算法也常常被应用于个性化推荐系统，以弥补基于协同过滤算法的诸如物品冷启动问题的固有缺陷。其是最早用于推荐系统的算法之一，具有思想简单，推荐结果的可解释性强[7]的特点。

## 1.3 本论文章节安排

本文共分了 9 个章节，下面将简要介绍各个章节的主要内容：

第一章主要介绍了个性化电影推荐系统的选题目的及其意义，并进一步说明了本文要实现的电影推荐系统所涉及到的常见推荐算法的国内外研究现状。

第二章分析了本文的个性化推荐系统所应达到的目标。

第三章在第二章所提出的需求的基础上给出了系统的总体设计，对系统总体架构进行了详细介绍。

第四到第七章详细介绍了系统的各个核心模块。第四章介绍了同时包括了基于协同过滤和基于内容的推荐算法的离线推荐系统。第五章针对系统需要捕捉用户的近期行为并快速为用户给出满足时效性要求的推荐的需求，提出了一种基于偏好队列的在线推荐算法。第六章所提出的混合推荐系统则综合第四，第五章的推荐算法，力求对各类算法扬长避短，主要解决了冷启动问题。第七章介绍的排序系统则完成了从推荐电影的候选集中合理选择出最终推荐电影，以及去重，热门补足等功能。

第八章则介绍了在对本文所设计的电影推荐系统的实现过程中的一些参数设置情况以及最终成果展示。

第九章总结了全文的成果并对本文所提出的系统的不足之处的改进做出了思考。

# 第2章　需求分析

个性化电影推荐系统需要根据用户的浏览记录从海量的电影数据中为用户推荐若干部用户可能喜欢的电影。在这个过程中，需要正确分析用户的历史记录并评估用户的偏好信息，以此为基础为用户做出较好的推荐。分析可知，个性化推荐系统需要做到一下四个方面：

（1）系统需要在不过多询问用户的偏好信息的情况下采集用户信息而进一步评估用户偏好。

（2）给出的推荐列表也应在保证准确性的同时保证推荐的多样性以更多发掘用户兴趣。

（3）考虑到用户在不同阶段的喜好往往不同，即用户的推荐喜好会随时间变化，所以推荐系统还应保证推荐的时效性，这就要求推荐系统给出的推荐列表更多的考虑用户的近期浏览记录，并能够快速感知到用户的偏好变化。

（4）推荐系统还应保证性能，保证系统能够在规定时间内给出推荐序列。

综上所述，个性化推荐系统给出推荐应该保证准确性，多样性，时效性和高效率。

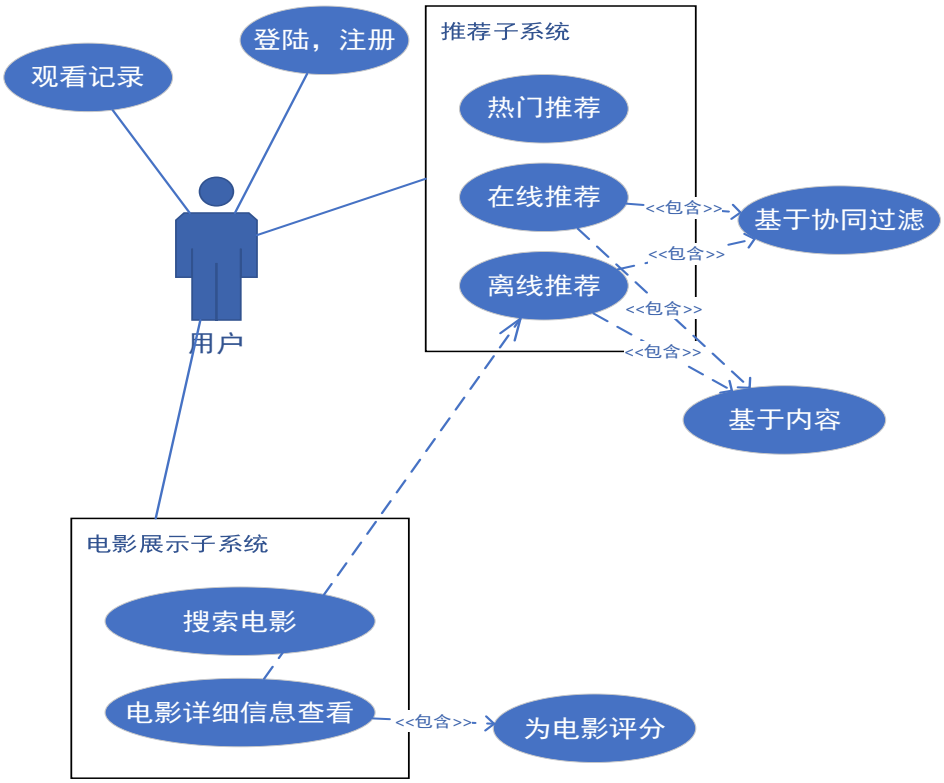


图 2.1 系统用例图

由此，如图 2.1 所示，个性化推荐系统具备了热门电影推荐，在线推荐和离线推荐等推荐功能。热门电影推荐是能够基于统计信息推荐出观看人数较多，评分较高的电影。离线推荐基于较长时间段内用户历史信息评估用户的喜好而给出推荐。与之不同的是，在线推荐是基于较短时间内用户的浏览记录快速给出推荐电影的，其能够快速适应用户的偏好变化并给出相应推荐。用户还可以查询电影的详细信息并为其评分，在电影详情展示页面还将调用推荐子系统，展示与该电影相似的电影。系统还应具备登陆，注册和观看记录查询等功能。

# 第3章　系统总体架构

系统的核心是推荐系统，构建出推荐系统所遵循的推荐模型又是重中之重。为了兼顾推荐的准确性和推荐的多样性[8 10]以便对用户喜好进行多方面的覆盖，系统需要构建起的推荐模型能够兼顾多方面指标。为此，推荐模型应该分为多个子系统，每个子系统或由一个独立的常用推荐模型，或由一个经过改进的经典推荐模型，或由多个推荐模型通过一定规则综合所组成。由此，推荐系统便可满足相应的推荐的准确性和内容的多样性。

对此，电影个性化推荐系统所具有的子模块有基于统计的定期热门电影推荐，有从用户历史观看信息中学习其内容并做出内容相似的相关电影推荐，有基于由于历史评价信息的协同过滤推荐，有基于最近喜爱的电影的实时性推荐，相关推荐将在综合排序后展出。

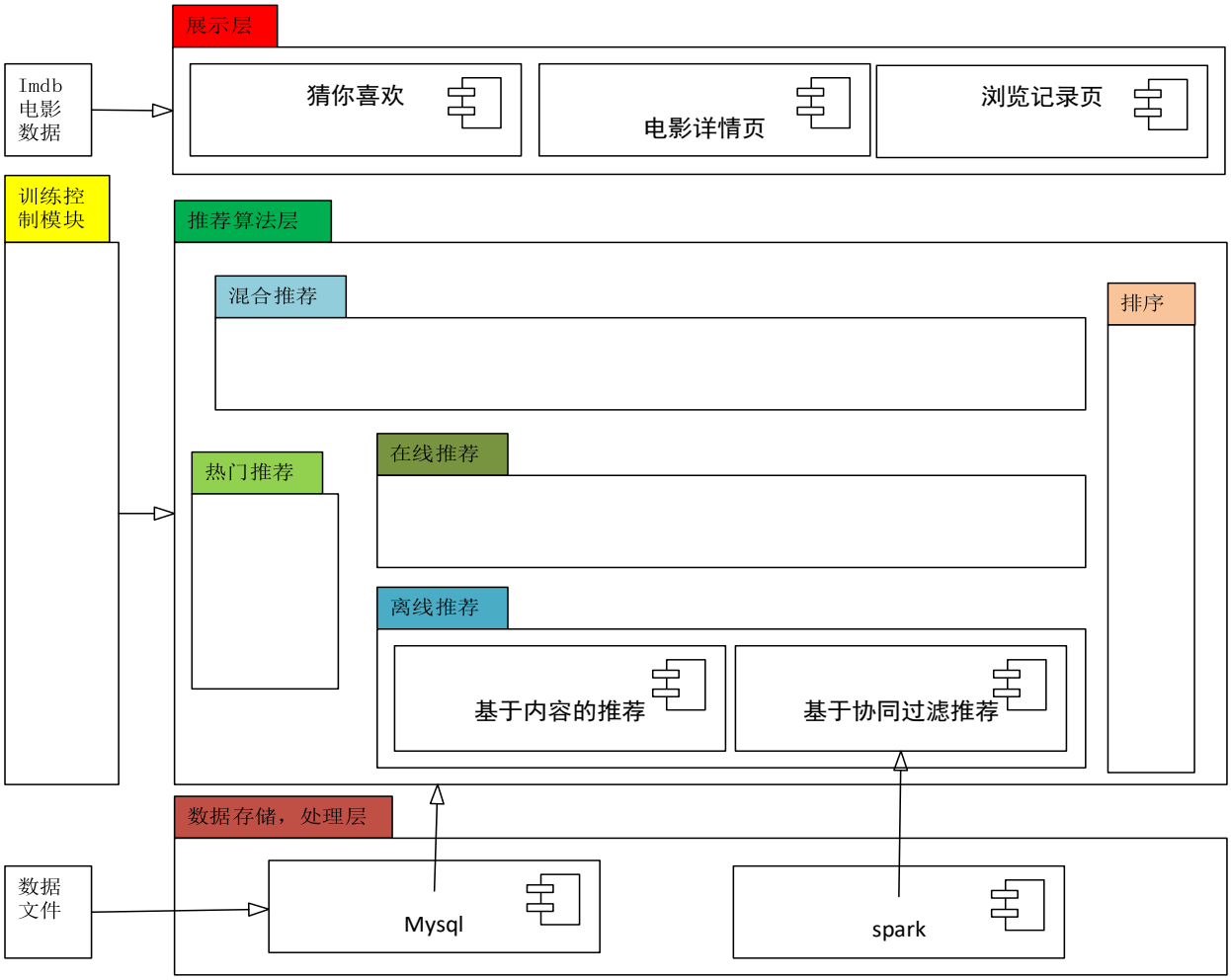


图 3.1　系统架构图

由此，本文设计的系统总体架构如图 3.1 所示，个性化电影推荐系统包括数据层，推荐层和展示层三层。在数据存储，处理层中，使用 Mysql 数据库管理系统中需要持久化的

数据，Spark 是专为大规模数据处理而设计的快速通用的计算引擎，本文使用 Spark 完成对协同过滤的训练。在推荐算法层中，除热门推荐，离线推荐和在线推荐子系统外，还包括混合推荐系统和排序系统。混合推荐系统综合各类推荐算法，以统一的接口留给外部调用。排序系统以一定策略对各类推荐算法给出的推荐结果进行选择，去重，排序和热门补足等一系列操作。在完成一段时间的数据积累后，可以使用训练控制模块在合适的时机重新训练各个推荐算法。展示层与用户体验密切相关，本文将为用户设计推荐主页面，电影详情页，浏览记录查看页面等用户界面。

# 第4章　离线推荐

## 4.1　基于协同过滤的推荐

### 4.1.1 协同过滤算法的建立

协同过滤算法是当前推荐系统最为常用的算法，其基本思路是基于用户对电影的共同历史评分，推荐用户可能喜欢的电影。基于交替最小二乘法（ALS）进行矩阵分解是实现协同过滤的常见有效方法，其通过最优化方法使预测评分矩阵和原始评分记录矩阵之间的差距最小化，从而得到电影的隐含特征向量，用户的隐含特征向量，并预测出对电影的评分。其预测评分往往具有准确性较高的优势。如图所示，ALS 算法将 $n\times m$ 维的评分矩阵 R 分解为一个 $n\times k$ 维的用户特征矩阵 $U$ 和 $k\times m$ 维的电影特征矩阵 $V$，所以易得预测评分矩阵中的元素 $\hat{r}_{ij}$ 满足下式。

$$\hat{r}_{ij} = u_i^T \bullet v_j \tag{4.1}$$

其中 $u_i$ 为 $k$ 维的用户喜好向量，$v_j$ 为 $k$ 维的电影特征向量，$\hat{r}_{ij}$ 为预测评分。

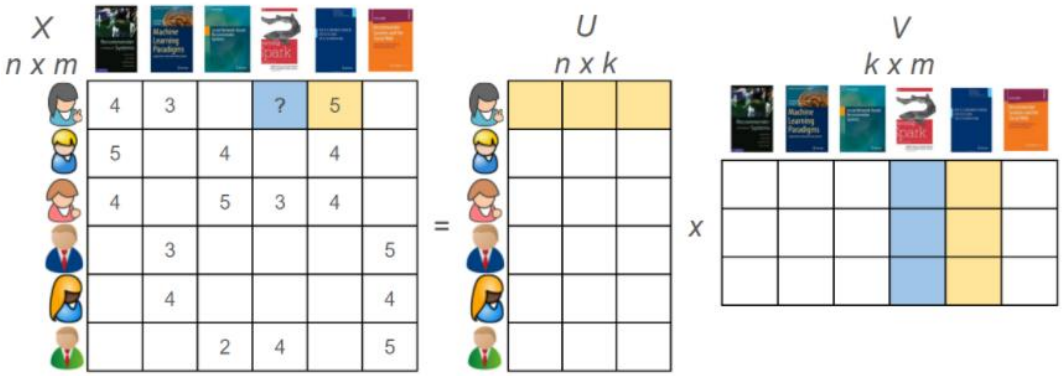


图 4.1 矩阵分解图

为使 $U$ 和 $V$ 之积与 $R$ 中已知评分数据尽可能的接近，定义损失函数如下：

$$C = \sum_{(i,j)\in R} [(r_{ij} - u_i^T v_j)^2 + \lambda(u_i^2 + v_j^2)] \tag{4.2}$$

使用梯度下降法等最优化方法最小化损失函数即可训练出最优的 $U$ 和 $V$，而预测评分 $\hat{r}_{ij}$ 可采用 4.1 计算。

### 4.1.2 协同过滤算法的使用

4.1.1 节建立起的协同过滤算法得到了每一个用户的偏好向量 $u$ 以及电影的特征向量 $v$，本部分介绍如何利用这两个已知信息来为用户推荐合适的电影。

首先需要确定一个能刻画推荐指数的指标以评估是否推荐某个电影，计算向量之间的相似性是刻画推荐指数的常用方法。实际应用中，有多种相似性算子可供使用，考虑到余弦相似性中具有向量内积的部分，和本文的应有场景很贴合，所以本文采用余弦相似性来刻画推荐指数。余弦相似性的计算如式 4.3，系统需要选择与用户喜好向量相似性大的电影进行推荐。

$$sim(u,v)=\frac{u^T \bullet v}{|u|\times|v|} \tag{4.3}$$

对每一个用户，本文都将系统中所有电影与之求余弦相似度，然后挑出相似度最高的 $n_u$ 个电影作为该用户的推荐列表，由于余弦相似度的计算量较大，本文将计算出的列表保存到数据库中以便后期直接使用，即用于离线推荐。由于当用户浏览某一电影时需要给出与之相似的推荐，即完成项目到项目的推荐，所以同理，本文也将通过计算电影之间的相似性保存根据电影的推荐列表。上述过程如图 4.2 所示。

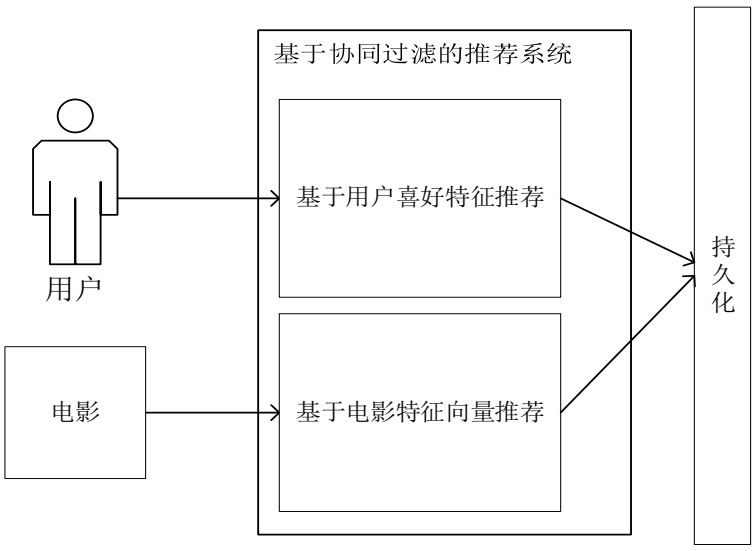


图 4.2 协同过滤推荐过程

## 4.2 基于内容的推荐算法

基于内容的推荐算法是在协同过滤的基础上发展起来的[14]，该算法思想简单，推荐结果的可解释性强。对于本文所涉及的电影推荐系统而言，其算法步骤主要有三步，即首先为电影画像，提取每部电影的特征，得到电影的特征向量，然后为用户画像，在电影画像的基础上再结合用户评分记录为每个用户进行画像，得到用户特征向量，最后再生成推荐列表，利用所得到的用户特征向量和电影特征向量求其喜好程度（或相似度）排序得到推荐序列。

1）为电影画像

为电影画像是要根据电影本身的内容特点构建电影的特征向量以表征每部电影。为电影画像的方法有很多，可以直接从原始数据中拿到类型数据，然后直接得到电影画像。为了使画像尽可能地精确，从原始数据所得到的维度应该尽可能多，可以使用的不仅仅是类型数据，还可以是导演，流派等。也可以应用自然语言处理等机器学习的方法，如 doc2vec，TF-IDF 方法等从电影名或者描述信息中获得相应的向量描述，甚至可以从图片，视频等数据中提取特征。

本文采用最简单的 one-hot 编码形式从原始电影数据中提取电影类型数据为电影进行画像。首先确定电影数据中所有电影类型标签所组成的集合 $L$，再确定电影画像向量的每一分量的值，其计算公式如下式。

$$l_{ij}=\begin{cases}0 & l_j \notin L_i \\ 1 & l_j \in L_i\end{cases} \qquad i \in (1,m),\ j \in (1,k)$$

（4.4）

其中 $l_{ij}$ 表示电影 $i$ 画像的第 $j$ 个分量，$k$ 为标签集合 $L$ 的元素个数，$L_i$ 表示电影 $i$ 的类型组成的标签集合。

2）为用户画像

对于用户画像，需要根据用户行为结合电影画像信息为用户进行画像，最基本的用户行为即评分行为，本文基于评分矩阵为用户画像。这里得到的用户偏好向量的分量表示用户对电影画像向量的对应分量的喜好程度。考虑到不同用户有不同的打分习惯，对电影的评价基准不一样，即有的用户打分的期望值高，有的低，为了消除用户本身打分习惯的影响，需要对每个用户求其平均分。用户画像的计算可以遵循公式 4.5，4.6。

$$avg_u = \sum_u r_{ui} \Big/ cout(u)$$

（4.5）

其中 $avg_u$ 表示 $u$ 的平均评分，$r_{ui}$ 代表用户 $u$ 对电影 $i$ 的评分，$cout(u)$ 表示 $u$ 所评价电影的总数。

$$like_{ul} = \sum_{l \in L_i} (r_{ui} - avg_u) \Big/ cout(l \in L_i) \tag{4.6}$$

$like_{ul}$ 表示用户 u 对标签 $l$ 的喜爱程度，$r_{ui}$ 表示用户 $u$ 对包含标签 $l$ 的电影 $i$ 的评分，$cout(l \in i)$ 代表了这样的电影的总数。

3）根据相似度进行推荐

在获得电影画像和用户画像的基础上便可通过用户或者电影进行相应的推荐了。与协同过滤类似，本文根据余弦相似度，分别从用户和电影角度获得相应推荐列表并保存。

# 第5章　在线推荐算法

## 5.1 在线推荐算法的目标

自此，本文引入了的基于内容的推荐算法和基于协同过滤的推荐算法作为个性化推荐系统的一部分，两者的传统用法的共同特点是基于用户的历史数据而训练模型评估用户的偏好，进而根据用户偏好和数据库中所有电影对比，即进行相似度计算，选取和用户偏好最匹配的电影作为该用户的推荐项目。分析可知，一方面，在此过程中，需要进行大量的计算，这使系统往往达不到实时性的要求。为此，本文将每个用户应该得到的电影推荐列表都先存入了数据库，以达到相应的性能要求和避免重复计算而导致算力浪费。另一方面，由于模型训练精度的要求，模型训练需要用户的大量历史数据，这些历史数据可能是用户相当长的时间内的评分纪录，也就是说，模型对用户当前偏好的刻画并不精确，且对用户短时间内的喜好偏好的改变并不易快速感知。由此，由于难以捕捉用户的当前偏好及其变化，就很难使系统做到满足实时性要求的推荐。在电影推荐系统中，如果用户刚刚对某电影给予了较高的评分，往往可能更希望观看与之相近的电影，即假设用户的偏好是有一定的连贯性的，如果此时系统不能快速捕捉到用户的近期偏好，而基于用户的长期历史数据进行推荐，则效果往往不好。由此需要提出在线推荐模型，其基于的是用户的近期数据，在线训练的模式进行推荐。

综合以上分析，在线推荐算法需要能够捕捉用户的近期偏好，并在此基础上快速为用户给出推荐。其推荐结果应尽量保证实时性，多样性等特点。

## 5.2 基于偏好队列的在线推荐算法

基于以上目标，本文提出了一种基于偏好队列的在线推荐算法（Online recommendation algorithm based on preference queue，ORBPQ）。基于偏好队列（ORBPQ）的在线推荐算法建立起了一个队列，该队列反映了用户近期的偏好的综合信息。队列中的每一个元素反映的用户的某一方面的偏好及其强度，随着时间的流逝用户偏好强度会逐渐降低直至消失。

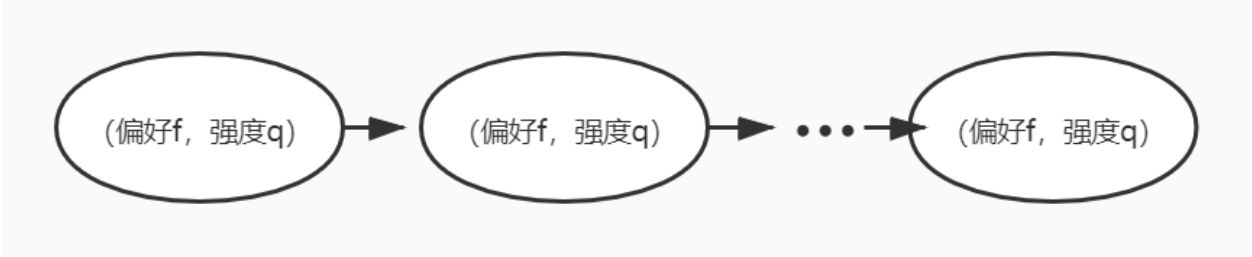


图 5.1：偏好队列

由于基于内容的推荐和基于协同过滤的推荐算法的建立及其实现，本文已经可以使用电影的特征向量或者隐特征向量代表每个电影，使用用户的偏好向量或隐偏好向量代表每个用户。由此，偏好队列的每个元素中的偏好信息 $f$ 就可以利用这些特征向量以一定规则所求得。下文将介绍队列元素的构造规则以及队列的出入队规则。

### 5.2.1 队列的出队入队

1）出队操作

为了保证队列元素中用户的偏好信息具有一定的时效性，需要构建出队规则。本文拟采用随着时间的推移，用户的偏好强度不断减小，并使强度小于阈值的偏好出队的方法保证偏好的时效性。

系统每经过时间 $t$，强度 $q$ 随公式 5.1 变化一次， $w_{out}$ 控制者强度减弱的速率。

$$q = q - w_{out} \tag{5.1}$$

如果队列中某一元素的强度 $q$ 满足 $q < q_{small}$，其中 $q_{small}$ 表示出队的强度阈值，则需要将该元素出队。

2）入队操作

初始时偏好队列是为空的，由此需要规定入队规则。为了保证队列能够及时反映用户的偏好信息，需要在用户表现出对某一电影感兴趣后，偏好队列能够及时反映，即偏好队列应实时刻画用户偏好。另一方面，由于队列长度有限，因此需要保证队列元素刻画的偏好 $f$ 之间具有一定的差异性以保证推荐的多样性。

当用户对某电影评分后，即进行入队操作。首先读取该电影的画像向量 $f_{movie}$，利用 5.2 式，从队首至队尾依次与队列元素所代表的偏好向量 $f$ 比对。

$$sim(f_{movie,}, f) < s \tag{5.2}$$

其中 $sim(f_{movie,}, f)$ 代表 $f_{movie,}$ 和 $f$ 之间的相似度， $s$ 为设置的相似度阈值，如果满足上式，则需对相应的偏好向量 $f$ 按下式进行修正：

$$f = (1-w) f + w \times f_{movie} \qquad w \in (0,1) \tag{5.3}$$

其中 $w$ 越大，则表示对最新特征越看重。同时按下式更新强度，其中 $w_{in}$ 为入队时强度变化率， $r_{ui}$ 为用户 u 对电影 $i$ 的评分。

$$q = q + w_{in} \times r_{ui}$$

（5.4）

如果队列中所有元素均不满足式 5.2，则需要插入新的元素到队列中，插入时分两种情况，一是队列未满的情况，即 $l < l_{\max}$，其中 $l$ 为队列当前长度，$l_{\max}$ 为队列的最大长度，二是队列已满的情况，此时需要先替换，再插入。替换的规则即替换掉队列中强度最小的元素即可。

### 5.2.2 队列元素推荐比例的确定

5.2.1 节建立起的队列代表着用户当前的综合偏好，而中的每一元素都代表了用户当前喜好的某一方面的偏好，如何综合这些偏好而给出一个推荐列表就是本节要解决的问题，即本节需要通过 5.2.1 节建立起的偏好队列按要求给出长度为 $n$ 的电影推荐列表。

考虑到队列中每一元素的强度信息代表着该元素所能反映用户当前最新偏好在该元素方面的强度，故本文以此作为划分各元素推荐比例的依据。元素 $i$ 所能推荐电影数量满足下式

$$n_i = \left\lceil \frac{q_i}{\sum_{i=1}^{m} q_i} \times n \right\rceil$$

（5.5）

其中 $n_i$ 表示第 $i$ 个队列元素所应产生的推荐电影数量，$m$ 等于队列长度，$n$ 为外部所期望 ORBPQ 所给出的推荐项目的个数。

### 5.2.3 相似电影快速搜索

5.2.1 节建立起的队列可以实时的，多角度的反映用户当前的偏好信息，但是如何根据这些偏好信息从电影空间中快速搜索出用户可能感兴趣的电影以满足在线推荐的性能要求就尤为关键。由于解空间的不连续性以及偏好向量具有高维的特性，传统智能搜索算法往往很难应用。文献[15]提出了一种利用聚类方法压缩搜索空间以达到快速搜索到推荐电影的方法。本文在此基础上提出一种基于多层聚类的快速搜索方法以满足在线推荐的性能要求。

聚类或分割方法有很多，本文拟采用最简单的 k-means 聚类算法对元素数据进行多层聚类。

k-means 聚类算法需首先指定聚类个数 k，每次根据点离聚类中心的余弦相似度划分点所属的类，再在类中求各个维度的均值以迭代产生新的聚类中心[16,17]。

本文将原本的电影特征向量数据利用 k-means 算法将电影数据聚类为$k_1$类，并得到$k_1$个聚类中心，再将这$k_1$个聚类中心作为下一次聚类的原始数据，进一步聚类$k_2$类，得到$k_2$个聚类中心又可以进一步聚类，由此重复$h$次，多层聚类的示意图如图 5.2 所示,首先将原始数据聚类为 0 到 9 共 10 类，然后再将这 10 类的聚类中心进一步聚类为 A，B，C 三类。

多层聚类自底向上建立起了一颗多路搜索树，如图 5.3 所示。当已知用户偏好向量而要求推荐电影时，则自顶向下搜索多路查找树即可。即首先在根结点内部搜索距离该向量最近的聚类中心，再在下一层的结点内搜索，以此直到搜索到叶子结点，再搜索以这个向量为聚类中心的电影，根据余弦相似度挑选出满足条件的电影推荐即可。

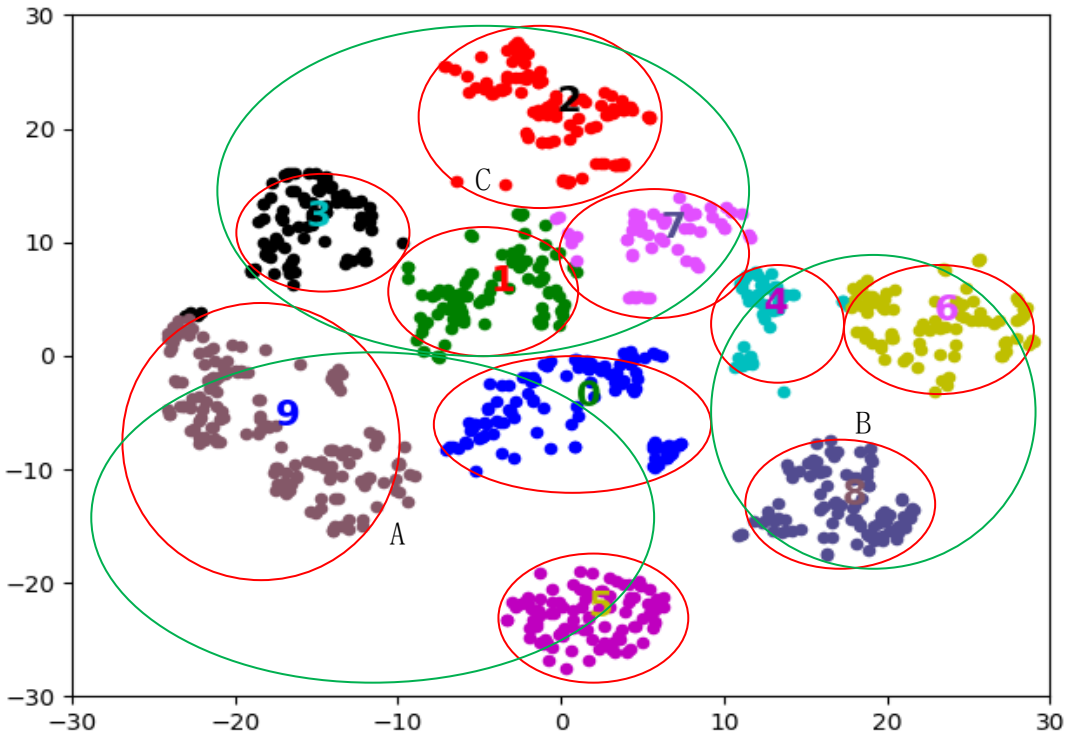


图 5.2 多层聚类

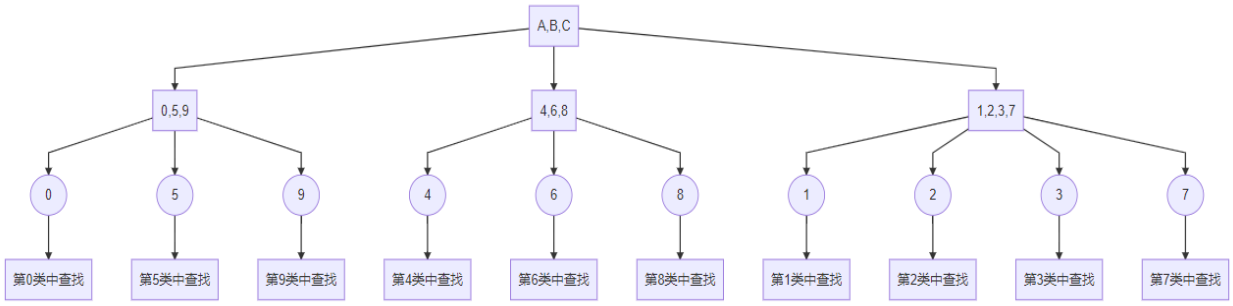


图 5.3 多路查找树

分析可知，如果电影数量为 $n$，则如果根据传统遍历整改电影空间而给出推荐电影，则时间复杂度为 $O(n)$。而根据本文的基于多层聚类的搜索方法，假设每次聚类个数平均为 $k$，形成的多路搜索树高为 $h$，最低层聚类内部电影数量平均为 $n^*$，则时间复杂度约为 $O(k\times h+n^*)$，显然电影数量 $n$ 满足下式

$$n=k^{h-1}\times n^* \tag{5.6}$$

所以易得采用本文方法可是搜索的时间复杂度呈指数级下降。考虑到建立多路搜索树的聚类过程，所以基于多层聚类的搜索方法具有离线建立慢，在线搜索快的特点，使得在线快速搜索成为可能。

## 5.3 ORBPQ 的近似实现

考虑到 ORBPQ 中利用的 k-means 算法需要预先指定簇数，且其随机生成原始聚类中心的特点使其可能陷入局部最优的固有缺陷[18],加之需要考虑推荐效果等重点因素，所以利用多层聚类建立起合适的多路搜索树并不容易。所以本文进一步提出一种 ORBPQ 近似算法。

之所以要利用多路搜索树是由于需要快速匹配出与队列元素中对应的偏好向量相近的电影。通过本文的离线推荐算法，已经知道并保存了每个电影相近的电影，而队列中的偏好向量就是根据电影的特征向量而求得的，之所以需要重新计算是因为式 5.3 的修正产生了新的，系统中不存在的特征向量作为用户的偏好向量，为了降低复杂度，本文对入队时的操作进行修改。在每一次用户对电影评分后不再与已有元素进行相似度比较，而是直接加入队列，并利用公式 5.7 设置偏好强度 $q$。

$$q=w_{in}\times r_{ui} \tag{5.7}$$

由此，队列中的偏好向量均是系统中所存在的电影特征向量，以此便可以利用离线推荐中基于项目而推荐项目的方式进行快速推荐。

# 第6章　混合推荐

自此，已经建立起了基于协同过滤的离线推荐模块，基于内容的离线推荐模块，基于协同过滤的在线推荐模块和基于内容的在线推荐模块四大模块。本章介绍如何将四大模块通过合理的方式组织起来形成一个综合推荐模块。

综合推荐模块要解决的问题主要有个方面：

1）综合四大模块，对外屏蔽四大模块的实现及组织细节，以统一的，易使用的接口为外层提供服务。

2）合理组织四大模块，充分利用各大算法的优点同时规避各个算法的固有缺陷，提高推荐的质量。

## 6.1 冷启动问题

冷启动问题[19,20]是推荐算法常见的问题，可分为用户冷启动问题和项目冷启动问题。由于推荐算法是基于用户的历史数据为用户进行推荐的，所以当一个新用户刚进入系统时，或者系统可以获得用户的历史数据过少时，对用户喜好的判断常常就不够准确，这就是用户冷启动问题。同理，当一部电影刚刚发布或者为其评分的人过少时，协同过滤对电影特征的刻画也不准确，从而导致推荐效果不满足要求，这就是项目冷启动问题。

## 6.2　混合推荐算法

### 6.2.1 问题分析

分析可知，基于协同过滤的推荐算法存在着项目冷启动问题，而基于内容的推荐算法不存在这一点，而基于协同过滤算法进行项目推荐往往比基于内容推荐更为准确。所以对基于项目的冷启动问题的解决可以先通过基于内容的算法进行推荐，使电影快速在系统内扩散开来，待电影获得一定量的评分数据积累时再逐步扩大基于协同过滤的推荐比例。

基于内容的推荐算法和基于协同过滤的推荐算法均存在用户冷启动问题，原因是他们均需要对用户喜好特征进行刻画，在用户数据较少时这往往不够准确。考虑到在线推荐算法是基于用户短时间内的电影浏览数据而给出相应电影推荐列表的模式，所需的数据非常少，所以当用户历史数据较少时应该更多的使用在线推荐算法，随着用户数据的搜集逐渐增多而逐步增加离线推荐的比例。

可见，混合推荐算法需要准确判断用户或电影状态并确定各个推荐算法的推荐比例。文献[19]提出的混合推荐算法将推荐分为冷启动用户，冷启动项目和历史数据丰富的老用户三种场景，每一种场景都有唯一确定的推荐策略进行推荐。受其启发，本文用用户的评分总数来刻画用户的状态，用电影的被评分总数来刻画电影的状态。如图 6.1 所示，在利

用用户数据进行推荐时，用户评价的电影数直接影响在线推荐和离线推荐的结果比例，即 $y_1/y_2$ 的值。同理，电影被评价数量直接影响协同过滤推荐和内容推荐的结果比例 $y_3/y_4$。

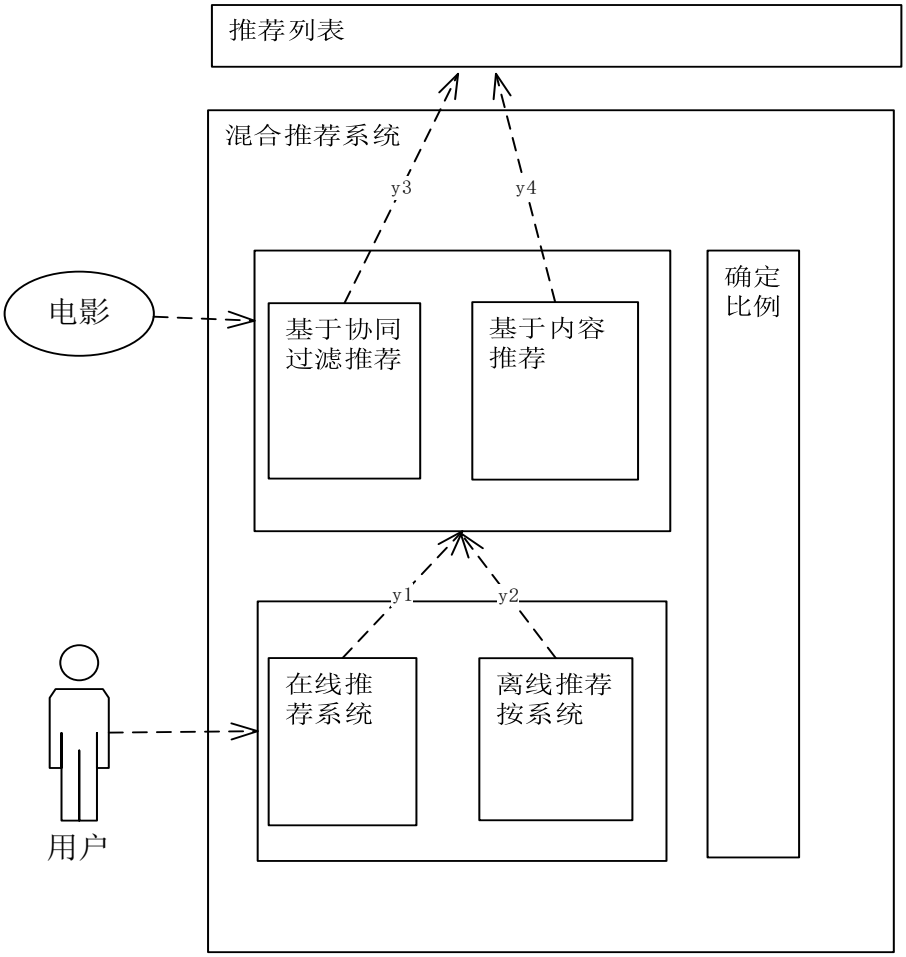


图 6.1 混合推荐系统

### 6.2.2 用户状态转移和比例确定

假定用户初始状态（冷启动用户）确定在线推荐与离线推荐的比例 $q=[1,0]$，表示推荐结果全来自与在线推荐算法，在用户使用系统一段时间后，共为 $n$ 部电影打分，此时他的推荐列表中来自在线推荐和离线推荐的比例逐渐稳定至 $q=[y_1,y_2]$，稳定比例为系统所指定。

定义函数 $h(x)$ 如式 6.1，易知 $\lim_{x\to\infty} h(x)=1$，$h(x)$ 的图像如图 6.2 所示，可知只要适当调整 $\sigma$ 的值便可调整 $h(x)$ 趋近于 1 的速度。

$$h(x)=\frac{x}{x+\sigma} \tag{6.1}$$

当用户 $u_i$ 评价电影数 $n_i$ 时，令其在线推荐与离线推荐的推荐比例 $q_i$ 满足下式

$$q_i = \begin{bmatrix} y_1 \times h(n_i) \\ 1 - y_1 \times h(n_i) \end{bmatrix}^T \tag{6.2}$$

同理，可以根据电影被评价数确定混合推荐系统中为电影推荐相似电影时协同过滤和基于内容的推荐的比例，本文不再赘述。

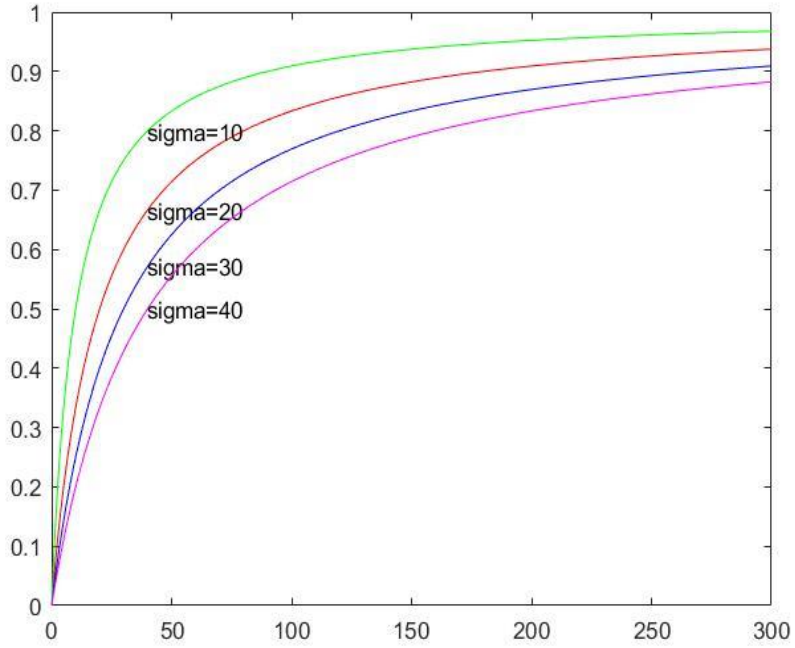


图 6.2 函数 $h(x)$ 示意图

# 第7章　排序系统

经过推荐系统的召回阶段，各类推荐算法便可以从海量电影数据中获取用户可能感兴趣的电影，排序系统就要从这些推荐序列中选出外部指定的电影个数，即排序系统需要在推荐算法召回阶段的得到的推荐序列的基础上，进一步进行选择，排序，去重和热门补足返回最终的推荐结果。

推荐召回阶段返回的推荐序列包括了电影 id 和用户对该电影的偏好程度，如何从 $n$ 个候选电影中选出 $n_{final}$ 个最终推荐电影是该阶段需要考虑的问题。考虑到候选队列中相邻电影的预测得分相差并不大，故直接根据预测得分排序结果选取前 $n_{final}$ 个并不合理。本文采用轮盘赌的方法随机选出 $n_{final}$ 个电影，一方面避免了 Top-N 方法直接截断候选列表的缺陷，另一方面使得每次刷新得到的推荐列表都有所差别，使推荐结果在满足准确性的同时又更具多样性，利于充分挖掘用户的潜在喜好。选择方法如式 7.1，7.2。

$$p_j = \frac{\hat{r}_j}{\sum_{i=1}^{n} \hat{r}_i} \tag{7.1}$$

其中 $\hat{r}_j$ 是电影 $j$ 的预测评分， $p_j$ 为电影 $j$ 被选中的概率。

$$s_j = \sum_{i=1}^{j} p_i \tag{7.2}$$

产生 0-1 的随机数 $r$，如果 $s_{j-1} < r \le s_j$，则电影 $j$ 被选中，由此选出 $n_{final}$ 部电影即可。

此外，排序阶段还包括了去掉用户已经观看后的电影，去掉重复推荐的电影，如果结果不足还有热门补足等操作。

# 第8章　系统实现

## 8.1 参数设置

影响协同过滤算法推荐的准确性的主要参数是矩阵分解时对应的低维矩阵的维度 rank 和正则化参数 $\lambda$，本文用 MovieLens 的 ml-latest-small 数据集进行模型训练和调参。数据集大小 1MB，其中包含有 600 个用户对 9000 部电影打分的 100，000 个评分数据，有评分数据文件 rating，电影数据文件 movies，可以寻找电影数据源 links 文件和电影标签数据文件 tags。

将数据集中的评分数据按照 6：2：2 的比例随机划分为训练集，验证集和测试集以用于调参。使用如式 8.1 所示的均方根误差（Root Mean Squared Error）作为调参标准，其中 $n$ 为样本个数，$r_i$ 为真实评分，$\hat{r}_i$ 为模型估计评分。

$$rmse=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i-\hat{r}_i)^2} \tag{8.1}$$

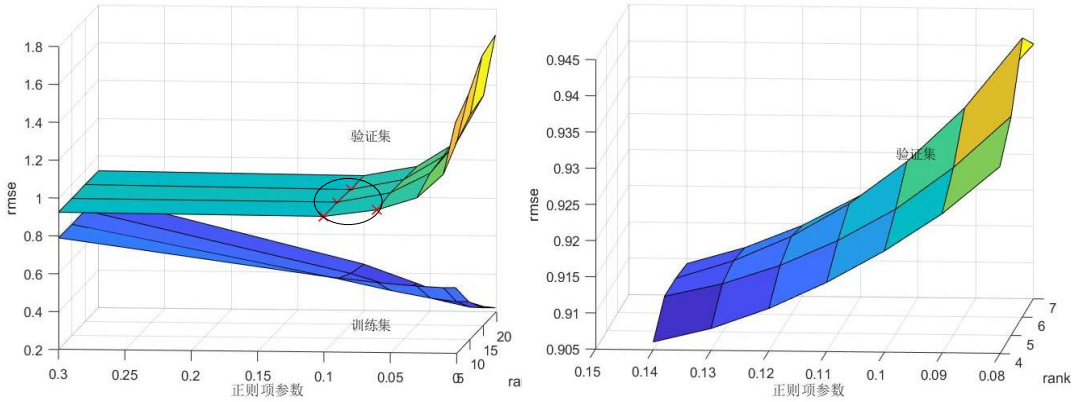


图 8.1 正则化系数与 rank 组合调参

分析图 8.1 的验证集和训练集结果可知，rank 的最佳值在 5 附近，$\lambda$ 的最佳值在 0.1 到 0.2 之间，固定 rank 为 5，寻找最佳 $\lambda$，得到结果如图 8.2 所示，观察验证集最佳 $\lambda$ 应该选择为 0.16，同时已知，$\lambda$=0.16 时测试集结果也满足要求，故选择 rank=5，$\lambda$=0.16。

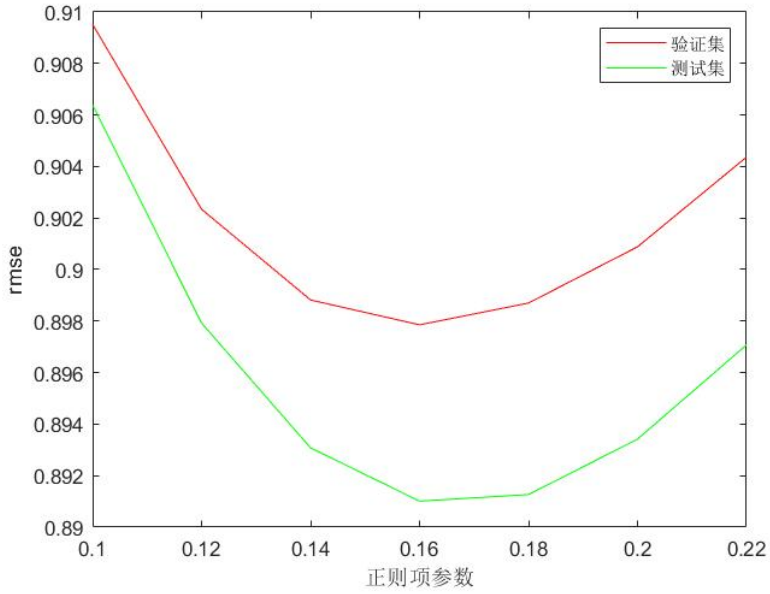


图 8.2 rank=5 时正则化系数调参图

此外，系统其余指定参数如下表所示

表格 8.1 系统参数默认值

| 符号 | 设定值 | 说明 |
|---|---|---|
| $l_{\max}$ | 20 | ORBPQ 算法最大队列长度 |
| t | 20s | ORBPQ 队列元素更新周期 |
| $w_{in}$ | 2 | ORBPQ 入队强度变化率 |
| $w_{out}$ | 1 | ORBPQ 队列元素强度减弱的速率 |
| $q_{small}$ | 0 | ORBPQ 出队强度门限 |
| $q_{on-off}$ | $[0.75, 0.25]$ | 稳定时在线和离线推荐的比例 |
| $q_{als-cont}$ | $[0.75, 0.25]$ | 稳定时协同过滤和内容推荐的比例 |

## 8.2 数据表设计

为了按照系统架构实现系统功能，就必须要利用数据库所提供的数据持久化功能，本节将介绍系统中数据库表的设计。

表格 8.2 用户表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| UserId | 是 | 否 | 用户 id |
| Password | | 否 | 账户密码 |
| Usercount | | 是 | 用户评价数 |

其中 Usercount 字段记录着用户共评价了多少部电影，设置该字段的目的是方便混合推荐算法确定推荐算法比例时使用。

表格 8.3 电影表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| MovieId | 是 | 否 | 电影 id |
| Title | | 否 | 电影名称 |
| Genres | | 是 | 电影类型 |

电影表的初始数据从原始的训练数据直接导入，将数据从文件导入数据库，方便后续的训练以及使用。

表格 8.4 电影链接表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| MovieId | 是 | 否 | 电影 id |
| ImdbId | | 否 | Imdb 对应的电影 id |
| TmdbId | | 否 | Tmdb 对应的电影 id |

电影链接表的初始数据从原始的训练数据文件中直接导入，其作用是方便系统直接从数据库中提取对应电影在其他电影网站的 id，以便在电影展示时爬取电影海报，电影描述等信息。

表格 8.5 用户评分表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| Userid | | 否 | 用户 id |
| movieId | | 否 | 电影 |
| rating | | 否 | 用户的评分 |
| timestamp | | 否 | 评分时间 |

用户评分表的初始数据从原始的训练数据文件中直接导入，用户评分表是推荐算法训练的基础性数据。

表格 8.6 电影统计信息表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| MovieId | 是 | 否 | 电影 id |
| avgScore | | 否 | 电影平均分 |
| Times | | 否 | 电影被评价的次数 |

电影统计信息表中的数据随着热门电影推荐模块的训练而更新。

表格 8.7 ALS 离线推荐表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| Userid | | 否 | 用户 id |
| recommendId | | 否 | 被推荐的电影 id |
| predictScore | | 否 | 预测的电影评分 |

表格 8.7 记录着 ALS 离线推荐的结果，基于 ALS 算法每次重新训练将会更新数据表中的内容。

表格 8.8 ALS 相似电影表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| MovieId | | 否 | 电影 id |
| similarUserId | | 否 | 相似用户 id |
| similarDegree | | 否 | 相似度评估 |

表格 8.8 记录着 ALS 离线计算相似电影的结果，由于 spark 并未直接提供相似电影的计算，为了充分利用 spark 所提供的高速计算的功能，所以本文实现系统时首先记录与电影相似的用户，再选择与用户相似的电影，即通过 I2U2I 的方式间接找到与某一电影相似的电影列表。

表格 8.9 记录着基于内容的离线推荐的结果，基于内容的推荐算法每次重新训练将会更新数据表中的内容。

表格 8.10 记录着基于内容的推荐算法评估出相似电影列表，每次基于内容的推荐算法重新训练会更新该表。

表格 8.9 基于内容的离线推荐

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| Userid | | 否 | 用户 id |
| recommendId | | 否 | 被推荐的电影 id |
| predictScore | | 否 | 预测的电影评分 |

表格 8.10 基于内容的相似电影表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| MovieId | | 否 | 电影 id |
| similarId | | 否 | 相似电影 id |
| similarDegree | | 否 | 相似度评估 |

表格 8.11 在线推荐的偏好队列表

| 字段名 | 主键 | 允许空 | 描述 |
|---|---|---|---|
| Id | 是 | 否 | 自动生成 |
| Userid | | 否 | 用户 id |
| movieId | | 否 | 电影 id |
| intention | | 否 | 强度 |

表 8.11 记录了基于偏好队列的在线推荐算法的偏好队列信息，在用户退出系统时，数据库需要记录用户当前的偏好队列信息，在用户下次进入该系统时，便从数据库中提取出来，这样在线推荐算法便可根据用户上次退出系统时的偏好队列信息进行推荐了。

## 8.3 系统核心算法实现

依照第 3 章的总体架构以及第 4 到 7 章建立的各类算法即可完成本系统。本节介绍系统中的实现各个核心算法的函数抽象。

### 8.3.1 协同过滤实现

按照系统架构，协同过滤算法属于离线推荐，即模型训练完成后随即为用户计算推荐列表并保存。MoveRecommend 类实现了协同过滤，其包括的主要方法如表 8.12 所示。方法 Train_and_save 完成基于协同过滤推荐的训练并将训练得到的模型保存，其需要评分数据作为训练数据并需要指定矩阵分解成的低秩矩阵的秩以及正则项系数。

Recommend_product_by_userid 函数利用 Train_and_save 方法训练出模型为指定的 id 的用户推荐电影，这里系统为每位用户推荐 20 部电影并将推荐结果保存到数据库表 ALS 离线推荐表。同理 Recommend_user_by_moveid 方法将每部电影推荐给 20 个用户并保存到 ALS 相似电影表。

表格 8.12 MoveRecommend 类主要函数

| 方法名 | 说明 |
| --- | --- |
| Train_and_save | 进行模型训练 |
| Recommend_product_by_userid | 为给定 id 的用户产生推荐列表并保存 |
| Recommend_user_by_moveid | 将给定 id 的电影退出推荐的用户列表并保存 |

### 8.3.2 基于内容的推荐算法实现

按照系统架构，基于内容的推荐属于离线推荐，即模型训练完成后随即为用户计算推荐列表并保存。cb 类实现了基于内容的推荐算法，其核心方法如表 8.13 所示。prepare_item_profile 方法通过分析训练文件中的电影类型字段为电影进行画像，user_profile 方法在电影画像数据的基础上结合用户评分数据为用户进行画像。在为电影，为用户画像的基础上，通过余弦相似度的计算，recommendbyuser 方法为每一位用户推荐 20 部电影并保存到基于内容的离线推荐表中，recommendbymoive 方法为每一部电影寻找最相似的 20 部电影并保存至基于内容的相似电影表中。

表格 8.13 cb 类主要方法

| 方法名 | 说明 |
| --- | --- |
| prepare_item_profile | 为电影画像 |
| user_profile | 为用户画像 |
| recommendbyuser | 为给定 id 的用户产生推荐列表并保存 |
| recommendbymoive | 为指定 id 的电影寻找相似电影并保存 |

### 8.3.3 基于偏好队列的推荐算法实现

按照系统架构，基于偏好队列的推荐算法属于在线推荐，即在线完成用户偏好的计算并在此基础上快速给出推荐。recomendOnline 类完成基于偏好队列的推荐算法中对用户偏好的捕捉，即完成对偏好队列中元素的构造，更改以及消亡并完成对应元素的出入队操作。

recomendOnline 类的主要方法如表 8.14 所示，当用户对某一电影评分后，便要进行入队操作，入队操作由 put 方法完成，put 方法的参数有用户 id，电影 id 以及用户为该电影的打分，由 5.3 节 ORBPQ 算法的近似实现的介绍，即可构造出相应的元素并按照相应的规则完成出入队操作，同样 pop 方法按照 5.2 节相应描述完成元素出队操作。同时，为了保证用户退出系统后偏好队列不消失而在下次进入系统时继续利用，所以需要 savequeue，loadqueue 方法利用数据库对偏好队列进行持久化操作。

表格 8.14 recomendOnline 类主要方法

| 方法名 | 说明 |
|---|---|
| put | 元素入队操作 |
| pop | 元素出队操作 |
| savequeue | 用户退出系统时将偏好队列保存至数据库 |
| loadqueue | 用户进入系统后将已有的偏好队列加载至系统中 |

### 8.3.4 混合推荐算法的实现

表格 8.15 recommendBasedOnfunc 类主要函数

| 方法名 | 说明 |
|---|---|
| getalsoffline | 从数据库中查询得到 als 离线推荐的结果 |
| getcontoffline | 从数据库中查询得到基于内容的离线推荐的结果 |
| recommend_online | 利用偏好队列得到在线推荐的结果 |
| getratio | 利用用户或电影‘状态’确定各类推荐算法的比例 |
| recmguasslike | 为给定的用户 id 推荐电影 |
| recmProduct | 为给点的电影 id 返回相似的电影 |

按照系统架构，混合推荐算法处在推荐算法层的最高层，其综合了本文提出的各类推荐算法以统一的接口向展示层提供推荐服务，即为主页面提供‘猜你喜欢’服务和为电影详情页提供‘相似电影’服务。recommendBasedOnfunc 类实现了混合推荐算法的功能，recommendBasedOnfunc 类的主要函数如表 8.15 所示，getalsoffline 方法和 getcontoffline 方法均属于离线推荐，只要给定用户 id 和要推荐出的电影数目即可从数据库中提取出相应的推荐列表。recommend_online 方法利用用户的偏好队列信息给出指定数目的推荐结果。混

合推荐算法需要对各类算法的应用做到扬长避短，即在不同情况时使用各类推荐算的的比例有所不同，getratio 方法可以确定比例，只要给定用户评分数量或者电影被评分数量，结合系统设定的推荐比例稳定时的最终比例，即可确定当前各类推荐算法的应用比例。recmguasslike 方法和 recmProduct 方法是对展示层的接口，利用上述几个函数，两者便可综合利用各类推荐算法给出合理推荐结果或找到合理的相似电影列表。

## 8.4 运行结果展示

本节展示并介绍个性化电影推荐系统的主要界面，即推荐主界面，电影详情页和浏览记录查询页。展示界面所应用的电影海报，简介，年代等信息均是通过网络爬虫的方式多线程地从电影评分网站 imdb 上爬取的。

如图 8.3 所示，个性化电影推荐系统的主页面分为“热门电影推荐”和“猜你喜欢”两部分组成，“猜你喜欢”栏目所展示的电影是根据用户的行为记录，综合应用本文所提出的各类推荐算法所给出的综合推荐结果。

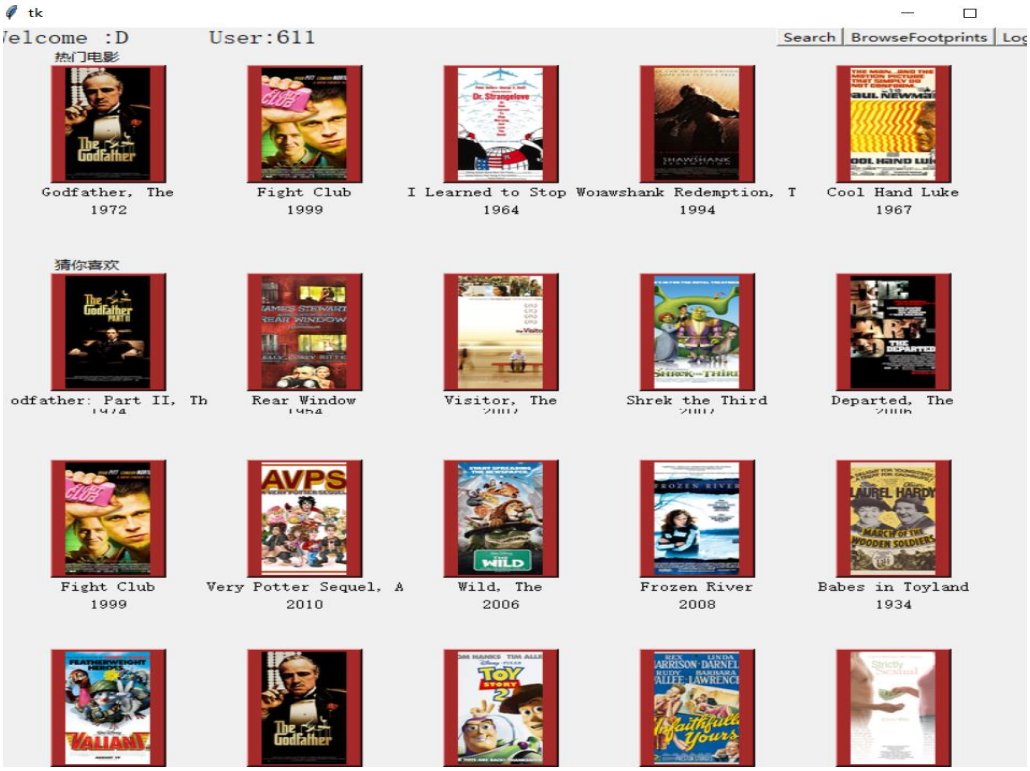


图 8.3 个性化推荐系统主页面

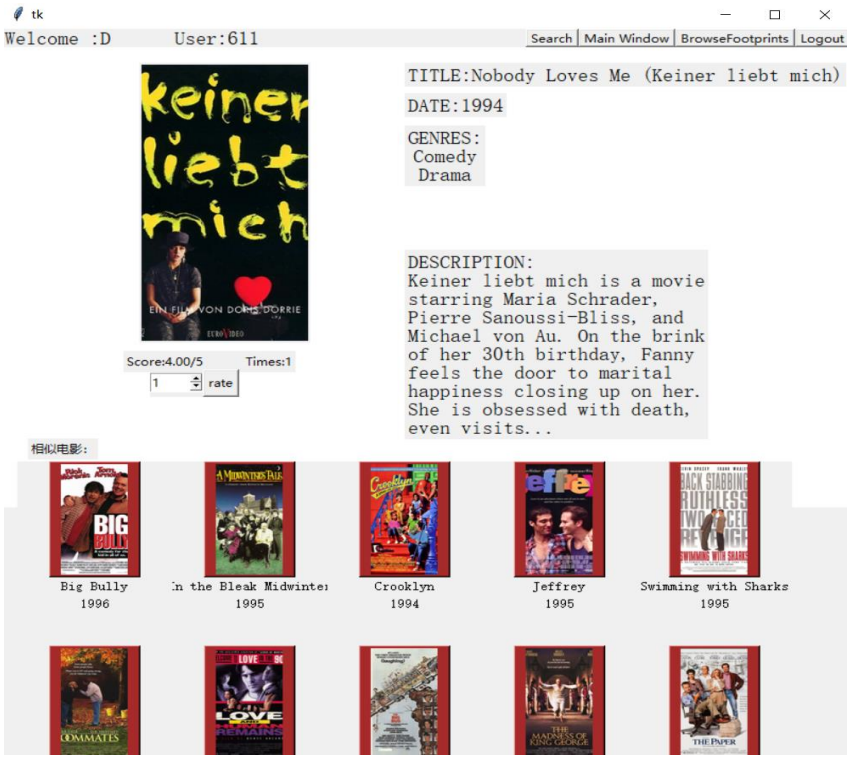


图 8.4 电影详情展示页

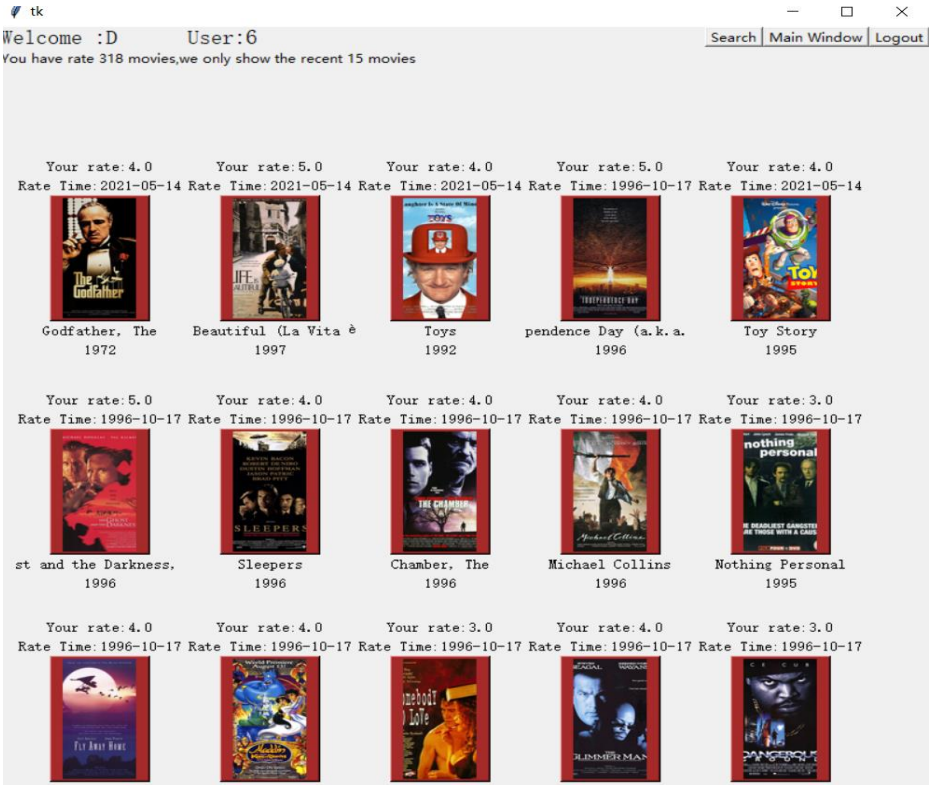


图 8.5 用户浏览记录

点击某一电影的封面即可进入如图 8.4 所示的电影详细信息展示页。电影详情页分两大部分组成，上部是电影的详情展示，并可完成评分操作，下部是综合了协同过滤和基于内容的推荐算法的离线推荐结果的相似电影展示。

如图 8.5 所示，用户还可以浏览自己的评分记录。

此外，系统还提供可对电影名进行模糊搜索，注册登陆等功能。

# 第9章　总结与展望

## 9.1 全文总结

影视领域是最早应用推荐系统的领域之一，推荐算法应用到视频领域有许多固有的难点，例如（1）一般而言，电影上映初期也是人们观看的高峰期，这就要求电影能够在上映和快速在人群中传播，而此时推荐算法将其推荐给对应人群时又有项目冷启动问题；（2）同样，新用户进入系统时又存在用户冷启动问题；（3）推荐算法中对比余弦相似度需要大量计算，在系统数据量较大时不能满足性能需求；（4）人的喜好经常性的改变，在不同的阶段可能偏好大有不同，传统推荐算法很难捕捉到当前用户的喜好并根据用户近期行为快速给出推荐；（5）推荐结果应该在满足准确性的同时充分挖掘用户的兴趣，即推荐结果应具多样性。

针对以上问题，本文使用基于内容的推荐和基于协同过滤推荐相结合的混合推荐方法，首先将冷启动项目使用基于内容的推荐算法快速推荐给对应人群，以使电影在上映初期就得到一定积累评分数据供协同过滤推荐算法进一步做更准确的推荐，由此缓解问题一。同时，本文使用综合在线推荐和离线推荐的混合推荐，将需要进行大量计算的基于内容的推荐和基于协同过滤的推荐作为离线推荐部分，从而提前计算出其推荐结果并保存由此解决了问题三，同时在用户数据较少时更多的使用在线推荐算法而解决问题二。本文提出的 ORBPQ 方法使用偏好队列的方法标识用户近期偏好信息从而解决问题四。对于问题五，本文使用扩大候选集而在排序阶段使用轮盘赌的方法随机选取的方法加以缓解。

## 9.2 未来展望

本文所设计的个性化电影推荐系统还有很多不足之处有待解决：

协同过滤算法重新训练与之前训练好的协同过滤模型无关，而是完全的推倒重新训练，可以改进协同过滤算法的训练过程，使其在之前训练好的模型的基础上方便的应对评分数据，电影数据，用户数据等训练数据的增删情况而重新训练以寻求降低训练计算量。

基于内容的推荐算法中使用 one-hot 编码方式只使用电影的类型数据来为电影画像。可以使用常用机器学习方法如 doc2vec，TF-IDF 方法等从电影名或者描述信息中获得相应的向量描述，甚至可以从图片，视频等数据中提取特征，从而使电影画像更准确。

排序系统中从候选集中的选择可以引入机器学习方法使推荐效果更佳。

在控制训练模块中需要人为的选取适当的时机进行模型重新训练，可以考虑使系统自动找到访问量小，重新训练后收益大的时机重新训练模型。例如可以考虑综合历史访问量，是否节假日，星期，时段，系统新产生的用户行为数据量等诸多因素使控制训练模块能自动选择一个恰当的时机重新训练各类模型。

# 参考文献

[1]Wang M , Li L , Lim E P , et al. Behavior analysis in social networks: Challenges, technologies, and trends[J]. Neurocomputing, 2016, 210(OCT.19):1-2.

[2]孙建凯. 面向排序的个性化推荐算法研究与实现[D].山东大学,2014.

[3]黄宇. 基于协同过滤的推荐系统设计与实现[D].北京交通大学,2015.

[4]尤方圆. 电影推荐系统的设计与实现[D].华中科技大学,2013.

[5]侯林坤.电影个性化推荐系统的构建[J].电脑知识与技术,2020,16(27):41-42.

[6] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Trans. Inf. Syst. 22 (2004) 143–177.

[7] H. Li, J. Cui, B. Shen, J. Ma, An intelligent movie recommendation system through group-level sentiment analysis in microblogs, Neurocomputing 210 (2016) 164–173

[8]唐瑞. 基于内容的推荐与协同过滤融合的新闻推荐研究[D].重庆理工大学,2016.

[9]赵凤跃. 协同过滤与基于内容的混合推荐算法研究[D].天津财经大学,2016.

[10]庞帆栋. 基于 Spark 的个性化电影推荐系统的设计与实现[D].东南大学,2017.

[11]Yutian Hu,Fei Xiong,Dongyuan Lu,Ximeng Wang,Xi Xiong,Hongshu Chen. Movie collaborative filtering with multiplex implicit feedbacks[J]. Neurocomputing,2020,398.

[12]单京晶. 基于内容的个性化推荐系统研究[D].东北师范大学,2015.

[13]王茄力. 基于 Spark 的混合推荐系统[D].中国科学技术大学,2017.

[14]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(01):1-15.

[15]魏慧娟,戴牡红,宁勇余.基于最近邻居聚类的协同过滤推荐算法[J].中国科学技术大学学报,2016,46(09):736-742.

[16]Shashi Sharma, Ram Lal Yadav. Comparative Study of K-means and Robust Clustering[J]. International Journal of Advanced Computer Research (IJACR),2013,3(12).

[17] Hornik K , Feinerer I , Kober M , et al. Spherical k-Means Clustering[J]. Journal of statistical software, 2012, 50(10):1-22.

[18]Mohamed Abubaker,Wesam Ashour. Efficient Data Clustering Algorithms: Improvements over Kmeans[J]. International Journal of Intelligent Systems and Applications(IJISA),2013,5(3).

[19]单晓红,王春稳,刘晓燕,张晓月.基于在线评论的混合推荐算法[J].系统工程,2019,37(06):130-138.

[20]于洪,李俊华.一种解决新项目冷启动问题的推荐算法[J].软件学报,2015,26(06):1395-1408.

# 致　谢

至此，本科阶段最后一门课程也快要结束，本科四年也将要画上句点，还依稀记得四年前我坐上校车穿过长江大桥来到“小余村”，如今将要离开这里还真有些不舍得。回想过往时光，有太多的人需要感谢。

首先感谢我的父母。他们平常很少过问我的学业，但每次告诉他们我的状态或决定时他们总是表示理解与支持，感谢他们无条件的信任与支持，这是我动力的源头。

其次要感谢这里的每一位老师。感谢他们对我学业上的帮助，他们在课堂上的风采令我神往，在学术上的严谨、勤奋是我终身学习的榜样。在这里特别感谢**老师，刘春老师和张睿麒老师。**老师是我的毕业设计指导老师，感谢她每隔两周都抽出时间听我的进度汇报，并耐心的给出指导意见，这篇论文也是在*老师的精心指导和大力支持下才完成的。刘春老师是我的班主任，感谢她对我们班学生的学习和生活的关心。张睿麒老师是我的辅导员，感谢他像朋友般的与我们相处，关心我们的学习和生活，耐心解答我们的各种疑惑，鼓励我们一点点变得更好。

最后，感谢一起生活，学习的同学，我们一起交流，共同成长，我从他们身上学到了很多。他们在我高兴时陪我插科打诨，在我失落失望时给我鼓励，在我犹豫不定时也会给我建议，再次谢谢他们，祝福他们。