



SHANGHAI MARITIME UNIVERSITY

电影推荐系统详细设计报告

课程名称： 软件项目管理

指导教师： 刘 晋

成 员： 肖 品 202030310304

成 员： 王子箐 202030310190

成 员： 邓盼盼 202030310178

日 期： 2021年4月 29 日

# 一、引言

## 1.1研究背景和意义

目前正处于信息爆炸的时代，成千上万的网络信息充斥在我们的生活方方面面。使用互联网的人数的增长也带来了用户相关数据的指数级的增长。面对海量的数据，用户往往感到束手无策、无处下手。如何帮助用户从海量的信息中获取用户最感兴趣的信息逐渐成为当今的热门研究之一。

信息过载问题的传统解决方案是基于被动响应的服务模式。即根据用户提出的有指向性的需求，过滤筛除用户不感兴趣的内容。但这种方式只适用于共同兴趣点较高的信息如：新闻、音乐等，无法满足用户获取个性化信息的需求。

通过个性化推荐系统，系统分析用户的历史数据对用户的兴趣爱好进行建模。在用户使用系统的过程中，记录并学习每个用户的兴趣，及时更新。用户无需特意填写大量的兴趣调查信息，可以减轻用户负担，提高用户认可度。

## 1.2国内外研究概况

### 1.2.1发展历程

1995年3月，在美国人工智能协会上，卡耐基.梅隆大学的Robert Armstrong等人提出了个性化导航系统Web Watcher；斯坦福大学的Marko Balabanovic 等人推出了个性化推荐系统LIRA。

1995年8月，在国际人工智能联合大会(IJCAI)上，麻省理工学院的Henry Lieberman提出了个性化导航智能体Letizia，第二年互联网巨头Yahoo推出了个性化入口My Yahoo,早期的基于协同过滤的个性化推荐系统PHOAKS和ReferralWeb是由美国AT&T公司设计。

现在热门的个性化电子商务如：京东，当当，淘宝的前身也可以追溯到1999年德国Dresden技术大学的Tanja Joerding TELLIM原型系统。

2003年，Google通过用户搜索的关键词来提供相关的广告,开创了AdWords的新盈利模式，成为Google广告收入的主要来源。而后Google又为AdWords添加了个性化服务元素。摒弃了以往只关注单次搜索的关键词的方法，通过对用户近期的搜索历史记录进行分析，建立用户的喜好和需求模型，使得推荐的广告内容更加精准。

随着Google AdWords营利模式的成功，雅虎也推出了SmartAds广告方案。雅虎拥有海量的用户信息，如用户的性别、年龄、地理位置、收入水平以及生活方式等能反映用户粗略个性的属性信息，再加上对用户搜索、浏览行为记录的分析，雅虎可以为用户呈现个性化的广告。

### 1.2.2推荐系统实例

近年来在机器学习与数据挖掘的领域中出现过大量的成型推荐系统实例:

1) ACF：Active Collaborative Filtering系统为卡耐基梅隆大学开发的电子文档主动协同过滤推荐系统，用户可以通过主动的方式将创建的电子文档或文档的超链接推荐给可能对此感兴趣的用户，适用于用户群体比较小的场合。

2) GroupLens： GroupLens系统是麻省理工大学开发的新闻信息自动协同过滤推荐系统，利用基于邻域的的方法，找到与用户有着相似兴趣的邻居用户，然后根据这些用户对特定物品的评分来预测该用户对物品的评分产生最终的推荐结果。

3) Ringo：Ringo系统由麻省理工大学媒体实验室开发的音乐协同过滤推荐系统,用于提供个性化的音乐推荐服务。预测用户喜欢的音乐和对特定音乐的评分。

4) MovieLens：MovieLens电影推荐系统是明尼苏达大学开发的协同过滤推荐系统，是一个推荐系统和虚拟社区网站，其主要功能为应用协同过滤技术和用户对电影的喜好，向用户推荐电影。

5)Netflix：Netflix是一家在线影片租赁提供商。公司能够提供超大数量的DVD，而且能够让顾客快速方便的挑选影片，同时免费递送。Netflix已经连续五次被评为顾客最满意的网站。可以通过PC、TV及iPad、iPhone收看电影、电视节目，可通过Wii，Xbox360，PS3等设备连接TV。Netflix大奖赛公开了大约1亿个1-5的匿名影片评级，数据集仅包含了影片名称，评价星级和评级日期，没有任何文本评价的内容。比赛要求参赛者预测Netflix的客户分别喜欢什么影片,要把预测的效率提高10%以上。

## 1.3报告主要内容

此报告是电影推荐系统的详细设计报告。电影推荐系统分为电影模块、推荐系统模块、用户模块 三个功能模块。

第二章分析了推荐系统中常用的相似度计算方法，基于邻域的算法，协同过滤的算法，基于内容的算法等评分预测算法的算法与优缺点问题。对热门电影的排行算法也进行了介绍。在设计过程中，我们通过分析各种算法找出最适合本系统的相似度计算公式以及评分预测算法。

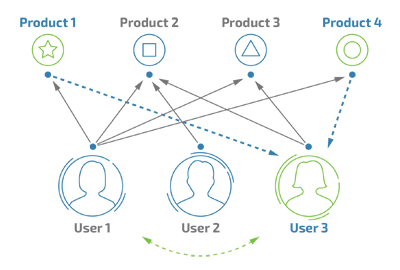
第三章对推荐系统中各个功能模块的设计进行了详细的说明，借助UML图表来辅助说明系统中数据的流向，各功能模块运行的流程，系统中实体以及实体的属性实体之间的联系等。

# 二、推荐算法

为在计划实现内按时交付，本系统采用易于实现且具有成效的推荐算法——协同过滤算法。

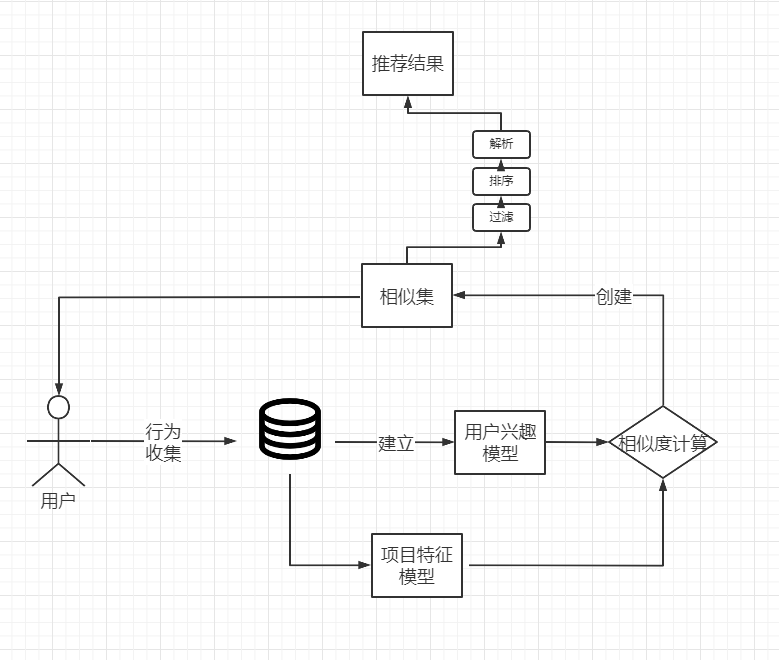
## 2.1基本思想

在本电影推荐系统中，基于用户的协同过滤算法是指通过用户的历史行为数据发现用户对item或content的喜欢，如收藏，评分等行为。根据不同用户对相同电影或内容的偏好程度计算用户之间的关系。在有相同喜好的用户间进行相同电影推荐。简单的说就是如果User1, User2两个用户都收藏了Product1、Product2、Product3相同的三部电影，并且给出了5星的评价。那么A和B就属于同一类用户。可以将User1看过的Product1也推荐给用户User2。同样的，可以对于电影也进行一个协同过滤的操作，之后进行推荐。



图一 基于用户协同过滤算法的示意图

协同过滤算法主要分为两个步骤：①寻找相似的用户集合；②寻找集合中用户喜欢的且目标用户没有的进行推荐。



图二 推荐算法具体设计图

## 2.2具体设计

### 2.2.1寻找相似的用户集合（数学问题）

1）Jaccard公式

Jaccard系数主要用于计算符号度量或布尔值度量的个体间的相似度，因为个体的特征属性都是由符号度量或者布尔值标识，因此无法衡量差异具体值的大小，只能获得“是否相同”这个结果，所以Jaccard系数只关心个体间共同具有的特征是否一致这个问题。如果比较X与Y的Jaccard相似系数，只比较xn和yn中相同的个数。

2）皮尔逊相关系数

皮尔逊相关系统是比欧几里德距离更加复杂的可以判断人们兴趣相似度的一种方法。它在数据不是很规范时，会倾向于给出更好的结果。

公式有4，且4个公式等价，其中E是数学期望，cov表示协方差，N表示变量取值的个数：

1 

2 

3 

4 

3）欧几里德距离

假定两个用户X、Y，均为n维向量，表示用户对n个商品的评分，那么X与Y的欧几里德距离就是：



多维欧几里德距离公式

数值越小则代表相似度越高，但是对于不同的n，计算出来的距离不便于控制，所以需要进行如下转换：



相似度公式，使得结果分布在(0,1]上，数值越大，相似度越高。

4）余弦距离

余弦距离，也称为余弦相似度，是用向量空间中两个向量余弦值作为衡量两个个体间差异大小的度量值。与前面的欧几里德距离相似，用户X、Y为两个n维向量，套用余弦公式，其余弦距离表示为：



余弦距离公式

即两个向量夹角的余弦值。但是相比欧式距离，余弦距离更加注意两个向量在方向上的相对差异，而不是在空间上的绝对距离。

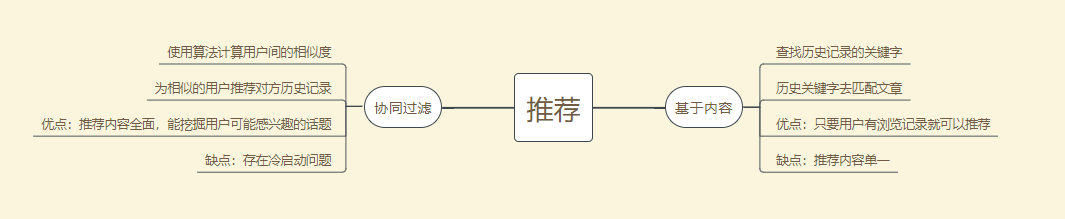
## 2.2.2寻找集合中用户喜欢的且目标用户没有的电影进行推荐

在选取上述方法中的一种得到各个用户之间相似度后，针对目标用户u，我们选出最相似的k个用户，用集合S(u,k)表示，将S中所有用户喜欢的物品提取出来并去除目标用户u已经喜欢的电影。然后对余下的电影进行评分与相似度加权，得到的结果进行排序。最后由排序结果对目标用户u进行推荐。其中，对于每个可能推荐的电影i，用户u对其的感兴趣的程度可以用如下公式计算：



用户u对物品i感兴趣的程度p

rvi表示用户v对i的喜欢程度，即对i的评分，wuv表示用户u和v之间的相似度。



图三 两种推荐算法介绍

## 2.3用户行为

要从用户的行为和偏好中发现规律，并基于此给予推荐，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。用户有很多方式向系统提供自己的偏好信息，而且不同的应用也可能大不相同，此电影推荐系统以以下（根据可靠程度降序）：

|  |  |  |  |
| --- | --- | --- | --- |
| 用户行为 | 类型 | 特征 | 作用 |
| 提交评分 | 显式 | 整数量化的偏好，取值是 [0, 5]； | 通过用户对电影的评分，可以尤其精确的得到用户的偏好 |
| 收藏 | 显式 | 布尔量化的偏好，取值是 0 或 1 | 通过用户对物品的投票，可以较精确的得到用户的偏好 |
| 前往观看 | 隐式 | 布尔量化的偏好，取值是 0 或 1 | 用户的购买是很明确的说明这个项目它感兴趣。 |
| 点击量( 查看 ) | 隐式 | 一组用户的点击，用户对物品感兴趣，需要进行分析，得到偏好 | 用户的点击一定程度上反映了用户的注意力，所以它也可以从一定程度上反映用户的喜好。 |
| 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 | 用户的页面停留时间一定程度上反映了用户的注意力和喜好，但噪音偏大，不好利用。 |
| 评论 | 显示 | 一段文字，需要进行文本分析，得到偏好 | 通过分析用户的评论，可以得到用户的情感：喜欢还是讨厌 |

表一 用户行为数据

收集了用户行为数据，我们还需要对数据进行一定的预处理，

其中最核心的工作就是：减噪和归一化。

1）减噪：用户行为数据是用户在使用应用过程中产生的，它可能存在大量的噪音和用户的误操作，我们可以通过经典的数据挖掘算法过滤掉行为数据中的噪音，这样可以是我们的分析更加精确。

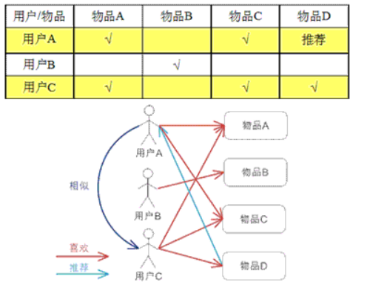
2）归一化：如前面讲到的，在计算用户对电影的喜好程度时，可能需要对不同的行为数据进行加权。但可以想象，不同行为的数据取值可能相差很大，比如，用户的查看数据必然比购买数据大的多，如何将各个行为的数据统一在一个相同的取值范围中，从而使得加权求和得到的总体喜好更加精确，就需要我们进行归一化处理。我们采用最简单的归一化处理，就是将各类数据除以此类中的最大值，以保证归一化后的数据取值在 [0,1] 范围中。

进行的预处理后，根据不同应用的行为分析方法，可以选择分组或者加权处理，之后我们可以得到一个用户偏好的二维矩阵，一维是用户列表，另一维是物品列表，值是用户对物品的偏好，一般是 [0,1] 或者 [-1, 1] 的浮点数值。

## 2.4算法具体应用

### 2.4.1基于用户的协同过滤（CF：Collaborate Filter）

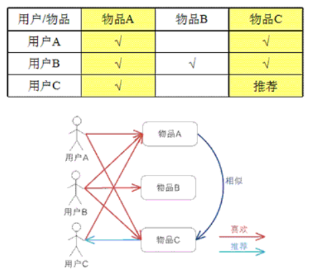
基于用户的 CF 的基本思想相当简单，基于用户对电影的偏好找到相邻邻居用户，然后将邻居用户喜欢的推荐给当前用户。计算上，就是将一个用户对所有电影的偏好作为一个向量来计算用户之间的相似度，找到 K 邻居后，根据邻居的相似度权重以及他们对物品的偏好，预测当前用户没有偏好的未涉及电影，计算得到一个排序的电影列表作为推荐。图 2 给出了一个例子，对于用户 A，根据用户的历史偏好，这里只计算得到一个邻居 – 用户 C，然后将用户 C 喜欢的电影 D 推荐给用户 A。



图四 基于用户的协调过滤示意图

### 2.4.2基于Item的协同过滤

基于Item的 CF 的原理和基于用户的 CF 类似，只是在计算邻居时采用电影本身，而不是从用户的角度，即基于用户对电影的偏好找到相似的电影，然后根据用户的历史偏好，推荐相似的电影给他。从计算的角度看，就是将所有用户对某个电影的偏好作为一个向量来计算物品之间的相似度，得到电影的相似电影后，根据用户历史的偏好预测当前用户还没有表示偏好的电影，计算得到一个排序的电影列表作为推荐。图 3 给出了一个例子，对于电影 A，根据所有用户的历史偏好，喜欢电影 A 的用户都喜欢电影 C，得出电影 A 和电影 C 比较相似，而用户 C 喜欢电影 A，那么可以推断出用户 C 可能也喜欢电影 C。

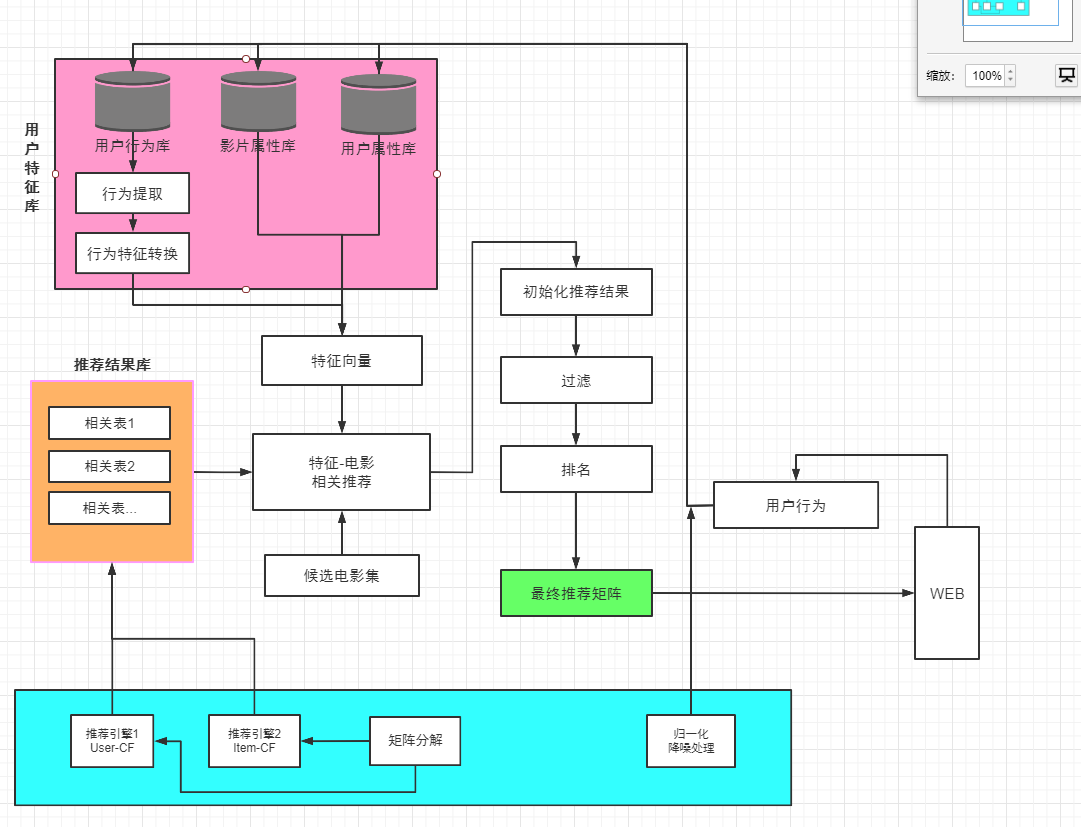


图五 基于Item的协调过滤示意图

### 2.4.3总结User CF 与 Item CF

研究推荐引擎的学者们在相同的数据集合上分别用 User CF 和 Item CF 计算推荐结果，发现推荐列表中，只有 50% 是一样的，还有 50% 完全不同。但是这两个算法确有相似的精度，所以可以说，这两个算法是很互补的。因此在我们的电影推荐系统中，两种方式均使用，以此来相互补充。

算法示意图如下图所示：



图六 算法示意图

# 三、系统设计

## 3.1系统功能介绍

在推荐系统的首页会对热门电影排序进行展示，对各类型电影进行部分的展示。

新用户在初次使用系统时，会先提交个人信息进行注册，勾选兴趣标签。

用户登录后可以按类别查看电影，按名称搜索电影，在电影详情页面进行浏览、评论、收藏操作，完善个人信息。

系统在收集了用户的浏览历史、评分、收藏、标签后会进行计算，及时反馈，推送给用户个性化的电影推荐信息。

管理员可以将新电影上新到系统中并且完善电影的具体信息，以便推荐使用。

## 3.2系统设计

### 3.2.1系统功能说明

电影推荐系统分为电影模块、推荐系统模块、用户模块，三个功能模块。

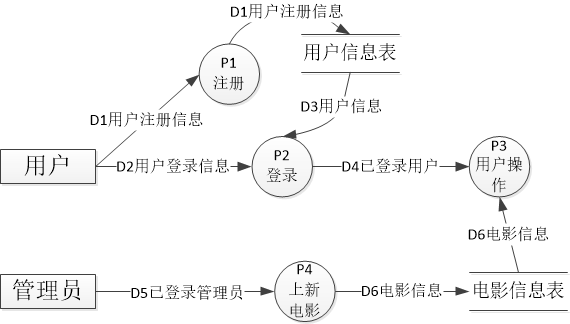
1）电影模块功能包括：上新电影，按主演、导演查看相关电影，按类别查看电影，按名称搜索电影。

2）推荐系统模块功能包括：推荐算法的运行，后台信息更新，热门及高分排行榜单，相似电影推荐。

3）用户模块功能包括：用户注册、登录，用户评论，用户收藏，用户浏览电影，添加兴趣标签。

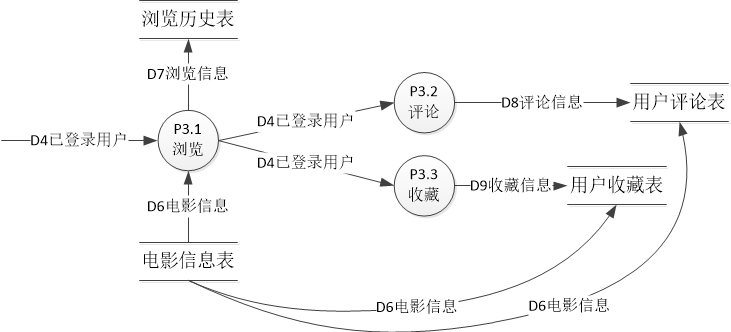
### 3.2.2系统数据流图

电影推荐系统的顶层数据流图如下图所示：



图七 电影推荐系统顶层数据流图

操作P3的用户操作层数据流图如下图所示：



图八 用户操作数据流图

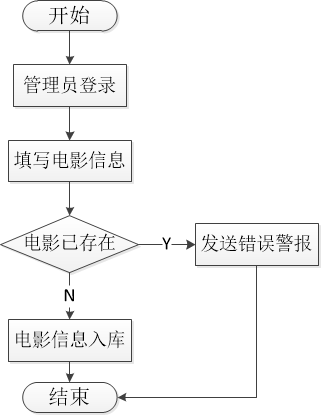
## 3.3功能模块设计

### 3.3.1电影模块：

电影模块功能包括：上新电影，按主演、导演查看相关电影，按类别查看电影，按名称搜索电影。

1）上新电影：

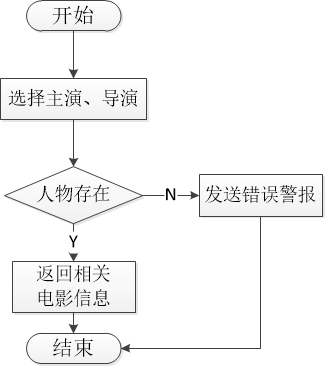
管理员可以通过系统上架新的电影，填写电影的描述和分类信息,使新上架的电影可以迅速被推荐给感兴趣的用户。运行流程图如下图所示：



图九 上新电影流程图

2）按主演、导演查看相关电影：

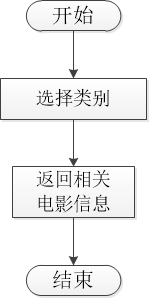
用户可以在电影详情页针对具体主演或导演进行进一步查找，浏览其导演或主演的电影。运行流程图如下图所示：



图十 按主演、导演查看相关电影流程图

3）按类别查看电影：

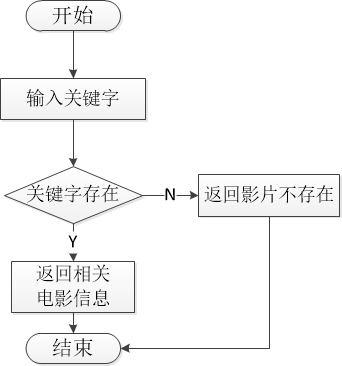
电影信息包括名称、年份、演员、导演、地区、类型、年份等。用户可以根据随意分类进行电影的查看。运行流程图如下图所示：



图十一 按类别查看电影流程图

4）按名称搜索电影：

系统提供给用户对电影按名称搜索的功能，用户可以直接搜素电影名称查找电影并对其进行评分、评论与收藏操作。运行流程图如下图所示：



图十二 按名称搜索电影流程图

### 3.3.2推荐系统模块：

推荐系统模块功能包括：推荐算法的运行，后台信息更新，热门及高分排行榜单，相似电影推荐。

1）推荐算法的运行：

系统在收集了用户的浏览历史、评分、收藏、标签后会进行计算，及时反馈，推送给用户个性化的电影推荐信息。运行流程图如下图所示：



图十三 推荐算法流程图

2）后台信息更新：

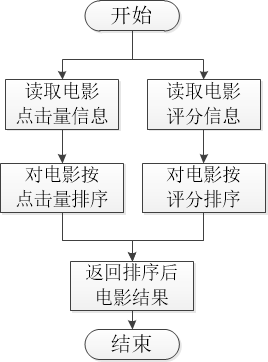
系统会分析用户的历史数据对用户的兴趣爱好进行建模。在用户使用系统的过程中，记录并学习每个用户的兴趣，及时更新。为用户提供基于个性化推荐的电影信息。运行流程图如下图所示：



图十四 后台信息更新流程图

3）热门及高分排行榜单：

系统在排除用户个人信息后，仅基于大数据的分析，对所有的电影进行热度排行。运行流程图如下图所示：



图十五 热门及高分排行榜单流程图

4）相似电影推荐：

推荐系统后台的分析算法通过协同过滤的算法计算出了每个电影的相似度，用户在电影的详情页面可以看到与该电影相似的其它电影。运行流程图如下图所示：



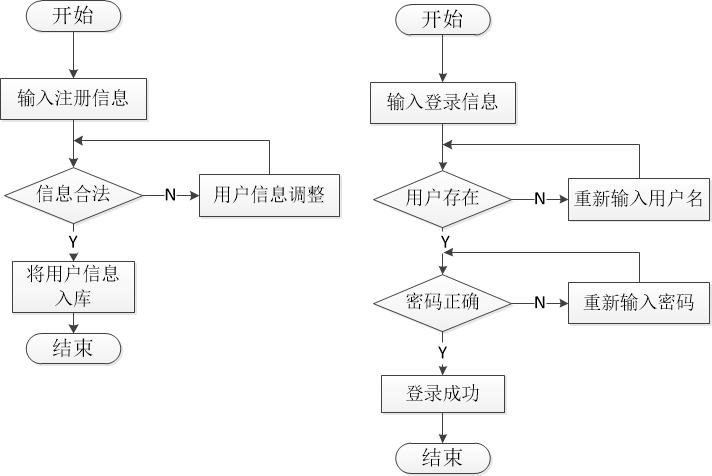
图十六 相似电影推荐流程图

### 3.3.3用户模块：

用户模块功能包括：用户注册、登录，用户评论，用户收藏，用户浏览电影，添加兴趣标签。

1）用户注册、登录：

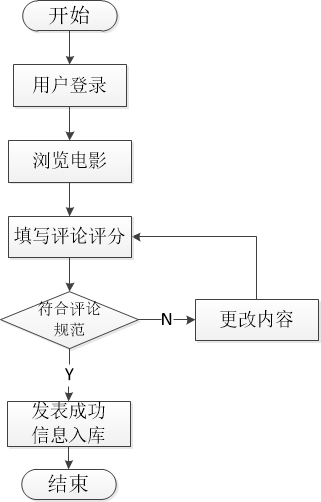
用户提交注册信息，完善个人信息页面。登录后可以进行各种用户操作以及在个人信息页面查看自己的浏览、评论和收藏。运行流程图如下图所示：



图十七 用户注册、登录流程图

2）用户评论：

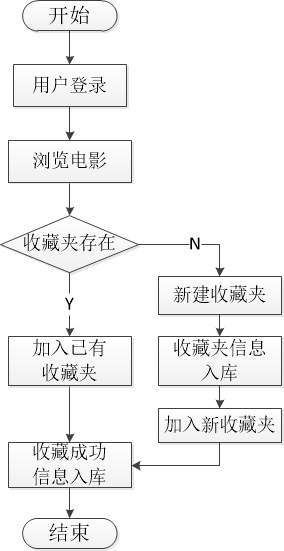
用户在电影详情页面浏览后，可以基于电影内容、其他用户评论、个人想法等对此电影进行评论与评分。系统在收集评论信息后，会对用户数据进行更新，为用户提供更新后的个性化推荐。运行流程图如下图所示：



图十八 用户评论流程图

3）用户收藏：

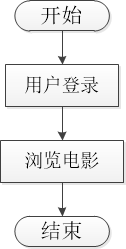
用户在电影详情页面浏览后，可以将感兴趣的电影收藏至个人收藏夹，用户个人可以创建自己的个性收藏夹。系统在收集收藏信息后，会对用户数据进行更新，为用户提供更新后的个性化推荐。运行流程图如下图所示：



图十九 用户收藏流程图

4）用户浏览电影：

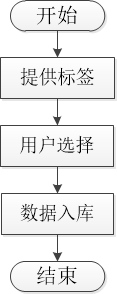
用户根据排行榜或搜索或首页推荐进入电影详情页进行浏览。系统会记录用户浏览时间以及浏览停留时长，对用户数据进行更新，为用户提供更新后的个性化推荐。运行流程图如下图所示：



图二十 用户浏览电影流程图

5）添加兴趣标签：

用户在注册时可以选择关于个人的兴趣爱好的标签以便开始使用推荐系统。在使用中也可以根据标签查找电影。系统会收集用户感兴趣标签的信息，用于个性化的推荐。运行流程图如下图所示：



图二十一 添加兴趣标签流程图

## 3.4数据库设计（E-R图、数据库的表）

### 3.4.1数据库表的设计

用户登录表（用户ID，密码）

管理员表（管理员ID，密码）

用户信息表（用户ID，用户名，密码，生日，性别，电话号码，邮箱，个人介绍，头像，地区）

电影信息表（电影ID，电影名称，发行日期，url观看地址，详情介绍，地区，*类别ID，标签ID，导演ID，演员ID*）

电影分类表（类别ID，类别名称）

用户浏览历史表（用户ID，*电影ID*，浏览时间，停留时间）

用户相似度信息表（用户ID，*相似用户ID*，相似度）

电影相似度（电影ID，*相似电影ID*，相似度）

用户评论表（评论ID，*用户ID，电影ID*，评分，评论，时间，热度）

用户收藏表（收藏夹ID，*电影ID*，收藏时间）

收藏夹表（收藏夹ID，*用户ID*，收藏夹名称）

演员表（演员ID，姓名，*出演电影ID*）

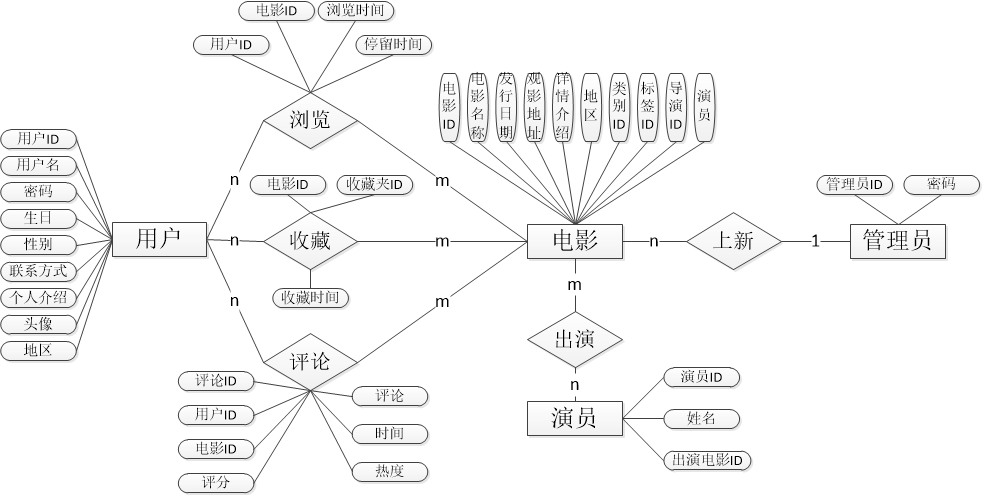
导演表（导演ID，姓名，*导演电影ID*）

用户标签表（用户ID，*标签ID*）

标签表（标签ID，标签名称）

### 3.4.2 E-R图

电影推荐系统的实体-联系图如下图所示：

****

图二十二 电影推荐系统E-R图

### 3.4.3 数据库详细设计

1）管理员表（管理员ID，密码）

admin（AID，password）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 管理员ID | AID | Int(15) | PK | 管理员登录系统使用的ID，是唯一的 |
| 密码 | password | varchar(20) |  | 登录系统所用的密码 |

2）用户登录表（用户ID，密码）

login（UID，password）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 用户ID | UID | Int(15) | PK | 用户登录系统使用的ID，是唯一的 |
| 密码 | password | varchar(20) |  | 登录系统所用的密码 |

3）用户信息表（用户ID，用户名，密码，生日，性别，电话号码，邮箱，个人介绍，头像，地区）

uinfo（UID，name，password，birth，sex，phone，email，introduction，avatar，region)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 用户ID | UID | Int(15) | PK | 用户登录系统使用的ID，是唯一的 |
| 用户名 | name | varchar(20) |  | 用户自主设置的名称 |
| 密码 | password | varchar(20) |  | 登录系统所用的密码 |
| 生日 | birth | Date |  | 用户生日，用于计算年龄与推荐 |
| 性别 | sex | varchar(20) |  | 男或女或保密 |
| 电话号码 | phone | Int(20) |  | 用户的联系方式 |
| 邮箱 | email | varchar(50) |  | 用户的有效邮箱 |
| 个人介绍 | introduction | varchar(100) |  | 用户自己填写的个人介绍信息 |
| 头像 | avatar | varchar(100) |  | 保存图像的链接，作为用户头像 |
| 地区 | region | varchar(50) |  | 用户所在地区用于推荐 |

4）标签表（标签ID，标签名称）

tag（TID，name）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 标签ID | TID | Int(15) | PK | 系统中各个性化标签的ID，是唯一的 |
| 标签名称 | name | varchar(20) |  | 标签的具体名称 |

5）用户标签表（用户ID，*标签ID*）

utag（UID，*tag ID*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 用户ID | UID | Int(15) | PK | 用户的ID，是唯一的 |
| 标签ID | TID | Int(15) | FK | 保存此用户所选择的标签 |

6）电影信息表（电影ID，电影名称，发行日期，url观看地址，详情介绍，地区，*类别ID，标签ID，导演ID，演员ID*）

movie（MID，name，date，URL，introduction，region，*CID，TID，DID，AID*）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 电影ID | MID | Int(15) | PK | 电影的ID，是唯一的 |
| 电影名称 | name | varchar(50) |  | 电影的名称 |
| 发行日期 | date | Date |  | 电影发行的日期，用于按年份查找电影 |
| url观看地址 | URL | varchar(100) |  | 提供官方观影地址链接 |
| 详情介绍 | introduction | varchar(200) |  | 电影的详情介绍 |
| 地区 | region | varchar(50) |  | 电影发行地，用于按地区查找电影 |
| 类别ID | CID | Int(15) | FK | 电影所在类别 |
| 标签ID | TID | Int(15) | FK | 电影所涉及标签 |
| 演员ID | DID | Int(15) | FK | 电影的演职员名单 |
| 导演ID | AID | Int(15) | FK | 此电影的导演 |

7）电影分类表（类别ID，类别名称）

category（CID，name）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 类别ID | CID | Int(15) | PK | 系统中电影各个类别的ID，是唯一的 |
| 类别名称 | name | varchar(20) |  | 类别的具体名称 |

8）收藏夹表（收藏夹ID，*用户ID*，收藏夹名称）

favorite（FID，*UID*，name）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 收藏夹ID | FID | Int(15) | PK | 此收藏夹的ID，是唯一的 |
| 用户ID | UID | Int(15) | FK | 创建此收藏夹的用户的ID |
| 收藏夹名称 | name | varchar(20) |  | 收藏夹的具体名称 |

9）用户浏览历史表（用户ID，*电影ID*，浏览时间，停留时间）

history（UID，*MID*，time，residence）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 用户ID | UID | Int(15) | PK | 用户的ID，是唯一的 |
| 电影ID | MID | Int(15) | PK,FK | 此用户浏览的电影ID |
| 浏览时间 | time | DateTime |  | 用户开始浏览的时间 |
| 停留时间 | residence | DateTime |  | 用户停留的时间 |

10）用户评论表（评论ID，*用户ID，电影ID*，评分，评论，时间，热度）

Comment（CID，*UID，MID，*rating，comment，time，popularity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 评论ID | CID | Int(15) | PK | 此评论的ID，是唯一的 |
| 用户ID | UID | Int(15) | FK | 用户的ID，是唯一的 |
| 电影ID | MID | Int(15) | FK | 此用户评论的电影ID |
| 评分 | rating | varchar(20) |  | 用户对此电影的评分 |
| 评论 | comment | varchar(200) |  | 用户对此电影的评论 |
| 时间 | time | DateTime |  | 用户评论的时间 |
| 热度 | popularity | varchar(20) |  | 此条评论的热度即点赞量 |

11）用户收藏表（收藏夹ID，*电影ID*，收藏时间）

ufavorite（FID，MID，time）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 收藏夹ID | FID | Int(15) | PK | 收藏夹的ID，是唯一的 |
| 电影ID | MID | Int(15) | PK,FK | 此用户浏览的电影ID |
| 收藏时间 | time | DateTime |  | 将电影加入收藏夹的时间 |

12）用户相似度信息表（用户ID，*相似用户ID*，相似度）

user\_similarity（UID，*sUID*，similarity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 用户ID | UID | Int(15) | PK | 用户的ID，是唯一的 |
| 相似用户ID | sUID | Int(15) | PK,FK | 与此用户相似的用户的ID |
| 相似度 | similarity | Int(10) |  | 计算出来的相似度 |

13）电影相似度（电影ID，*相似电影ID*，相似度）

movie\_similarity（MID，*sMID*，similarity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 电影ID | MID | Int(15) | PK | 电影的ID，是唯一的 |
| 相似电影ID | sMID | Int(15) | PK,FK | 与此电影相似的电影的ID |
| 相似度 | similarity | Int(10) |  | 计算出来的相似度 |

14）演员表（演员ID，姓名，*出演电影ID*）

actor（AID，name，*MID*）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 演员ID | AID | Int(15) | PK | 演员的ID，是唯一的 |
| 姓名 | name | Varchar(20) |  | 演员的姓名 |
| 出演电影ID | MID | Int(15) | FK | 此演员出演的所有电影的ID |

15）导演表（导演ID，姓名，*导演电影ID*）

director（DID，name，*MID*）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文 | 字段名 | 类型 | 主外键 | 描述 |
| 导演ID | DID | Int(15) | PK | 导演的ID，是唯一的 |
| 姓名 | name | Varchar(20) |  | 导演的姓名 |
| 导演电影ID | MID | Int(15) | FK | 此导演导的所有电影的ID |