**技术文档**

1、本系统技术介绍

* 前端: bootstrap3 + vue2 + jquery
* 后端: django 2.2.1 + django--rest-framework (MVC框架)
* 数据库: sqlite3/mysql8.0
* 算法: 基于用户的协同过滤推荐算法 + 基于物品的协同过滤推荐算法

# # 运行说明

每次运行需打开pycharm软件

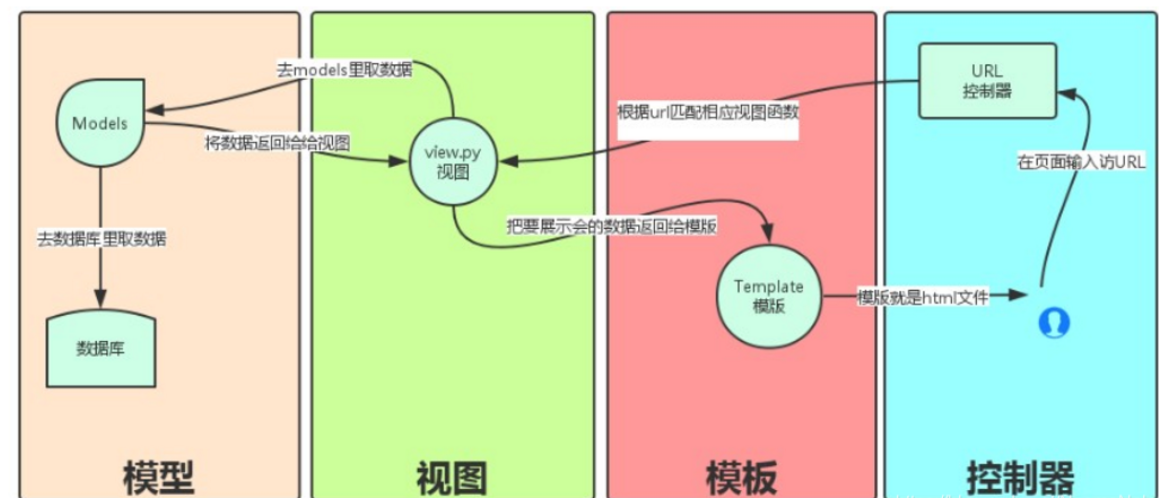
-pycharm 的terminal命令行下执行：python manage.py runserver

- 浏览器输入 http://127.0.0.1:8000 访问

- 后台访问地址：http://127.0.0.1:8000/admin/

账号密码 (admins，admins)

1、Django框架



MVC是众所周知的模式，即：将应用程序分解成三个组成部分:model(模型),view(视图),和 controller(控制 器)。其中：

M——管理应用程序的状态（通常存储到数据库中），并约束改变状态的行为（或者叫做“业务规则”）。

C——接受外部用户的操作，根据操作访问模型获取数据，并调用“视图”显示这些数据。控制器是将“模型”和“视图”隔离，并成为二者之间的联系纽带。

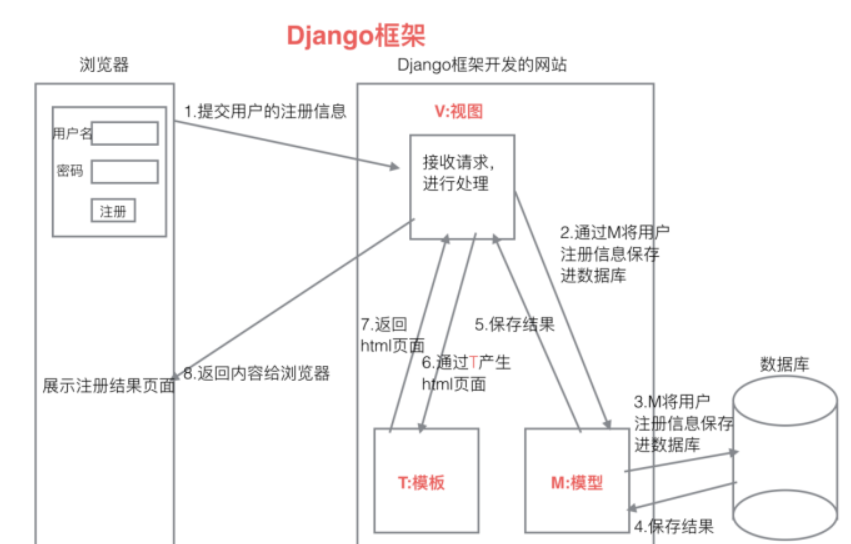
V——负责把数据格式化后呈现给用户。

Django也是一个MVC框架。但是在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式：

M 代表模型（Model），即数据存取层。 该层处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。

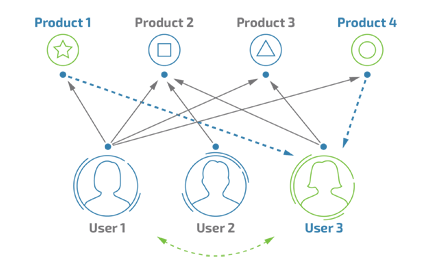
T 代表模板(Template)，即表现层。 该层处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。

V 代表视图（View），即业务逻辑层。 该层包含存取模型及调取恰当模板的相关逻辑。 你可以把它看作模型与模板之间的桥梁。



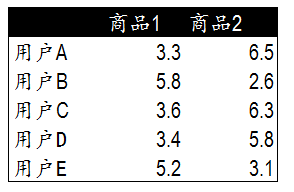
1、基于用户的协同过滤算法(user-based collaboratIve filtering)

　　基于用户的协同过滤算法是通过用户的历史行为数据发现用户对商品或内容的喜欢(如商品购买，收藏，内容评论或分享)，并对这些喜好进行度量和打分。根据不同用户对相同商品或内容的态度和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商品推荐。简单的说就是如果A,B两个用户都购买了x,y,z三本图书，并且给出了5星的好评。那么A和B就属于同一类用户。可以将A看过的图书w也推荐给用户B。

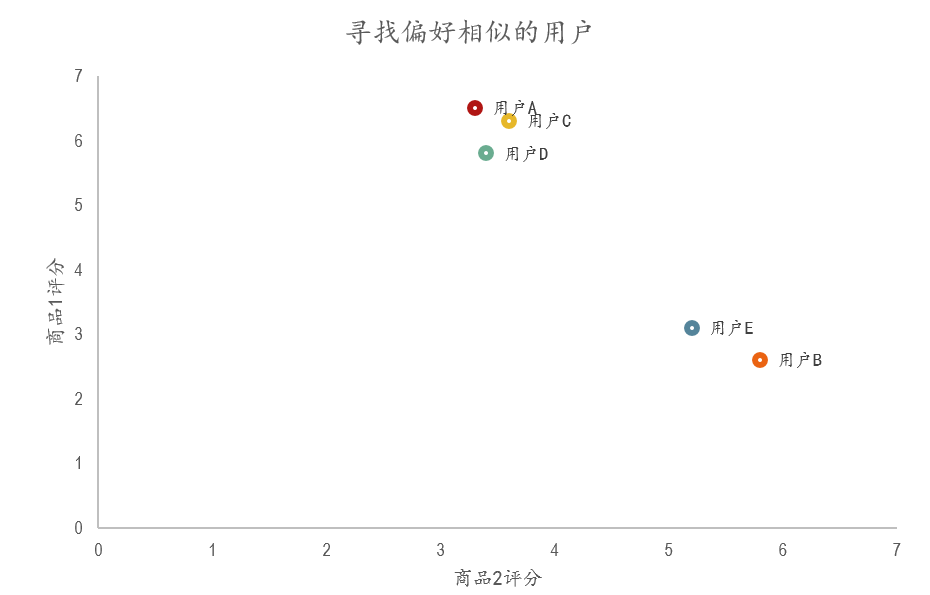


1.1寻找偏好相似的用户

　　我们模拟了5个用户对两件商品的评分，来说明如何通过用户对不同商品的态度和偏好寻找相似的用户。在示例中，5个用户分别对两件商品进行了评分。这里的分值可能表示真实的购买，也可以是用户对商品不同行为的量化指标。例如，浏览商品的次数，向朋友推荐商品，收藏，分享，或评论等等。这些行为都可以表示用户对商品的态度和偏好程度。



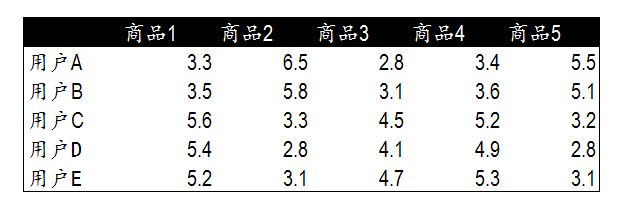
　　从表格中很难直观发现5个用户间的联系，我们将5个用户对两件商品的评分用散点图表示出来后，用户间的关系就很容易发现了。在散点图中，Y轴是商品1的评分，X轴是商品2的评分，通过用户的分布情况可以发现，A,C,D三个用户距离较近。用户A(3.3 6.5)和用户C(3.6 6.3)，用户D(3.4 5.8)对两件商品的评分较为接近。而用户E和用户B则形成了另一个群体。



　　散点图虽然直观，但无法投入实际的应用，也不能准确的度量用户间的关系。因此我们需要通过数字对用户的关系进行准确的度量，并依据这些关系完成商品的推荐。

　　1.3皮尔逊相关度评价

　　皮尔逊相关度评价是另一种计算用户间关系的方法。他比欧几里德距离评价的计算要复杂一些，但对于评分数据不规范时皮尔逊相关度评价能够给出更好的结果。以下是一个多用户对多个商品进行评分的示例。这个示例比之前的两个商品的情况要复杂一些，但也更接近真实的情况。我们通过皮尔逊相关度评价对用户进行分组，并推荐商品。



　　1.4皮尔逊相关系数

　　皮尔逊相关系数的计算公式如下，结果是一个在-1与1之间的系数。该系数用来说明两个用户间联系的强弱程度。

　　相关系数的分类

0.8-1.0 极强相关

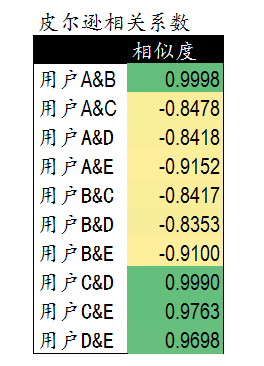
0.6-0.8 强相关

0.4-0.6 中等程度相关

0.2-0.4 弱相关

0.0-0.2 极弱相关或无相关

　　通过计算5个用户对5件商品的评分我们获得了用户间的相似度数据。这里可以看到用户A&B，C&D，C&E和D&E之间相似度较高。下一步，我们可以依照相似度对用户进行商品推荐。



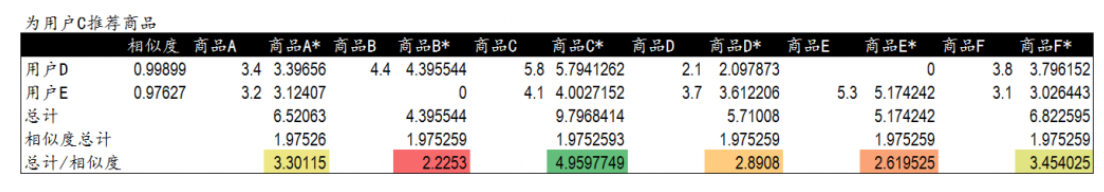
　　2,为相似的用户提供推荐物品

　　为用户C推荐商品

　　当我们需要对用户C推荐商品时，首先我们检查之前的相似度列表，发现用户C和用户D和E的相似度较高。换句话说这三个用户是一个群体，拥有相同的偏好。因此，我们可以对用户C推荐D和E的商品。但这里有一个问题。我们不能直接推荐前面商品1-商品5的商品。因为这这些商品用户C以及浏览或者购买过了。不能重复推荐。因此我们要推荐用户C还没有浏览或购买过的商品。

　　加权排序推荐

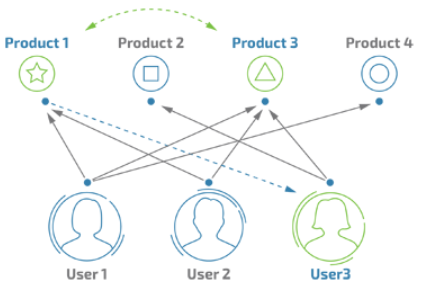
　　我们提取了用户D和用户E评价过的另外5件商品A—商品F的商品。并对不同商品的评分进行相似度加权。按加权后的结果对5件商品进行排序，然后推荐给用户C。这样，用户C就获得了与他偏好相似的用户D和E评价的商品。而在具体的推荐顺序和展示上我们依照用户D和用户E与用户C的相似度进行排序。



　　以上是基于用户的协同过滤算法。这个算法依靠用户的历史行为数据来计算相关度。也就是说必须要有一定的数据积累(冷启动问题)。对于新网站或数据量较少的网站，还有一种方法是基于物品的协同过滤算法。

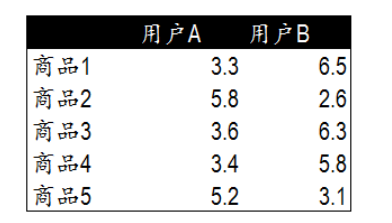
二、基于物品的协同过滤算法(item-based collaborative filtering)

　　基于物品的协同过滤算法与基于用户的协同过滤算法很像，将商品和用户互换。通过计算不同用户对不同物品的评分获得物品间的关系。基于物品间的关系对用户进行相似物品的推荐。这里的评分代表用户对商品的态度和偏好。简单来说就是如果用户A同时购买了商品1和商品2，那么说明商品1和商品2的相关度较高。当用户B也购买了商品1时，可以推断他也有购买商品2的需求。

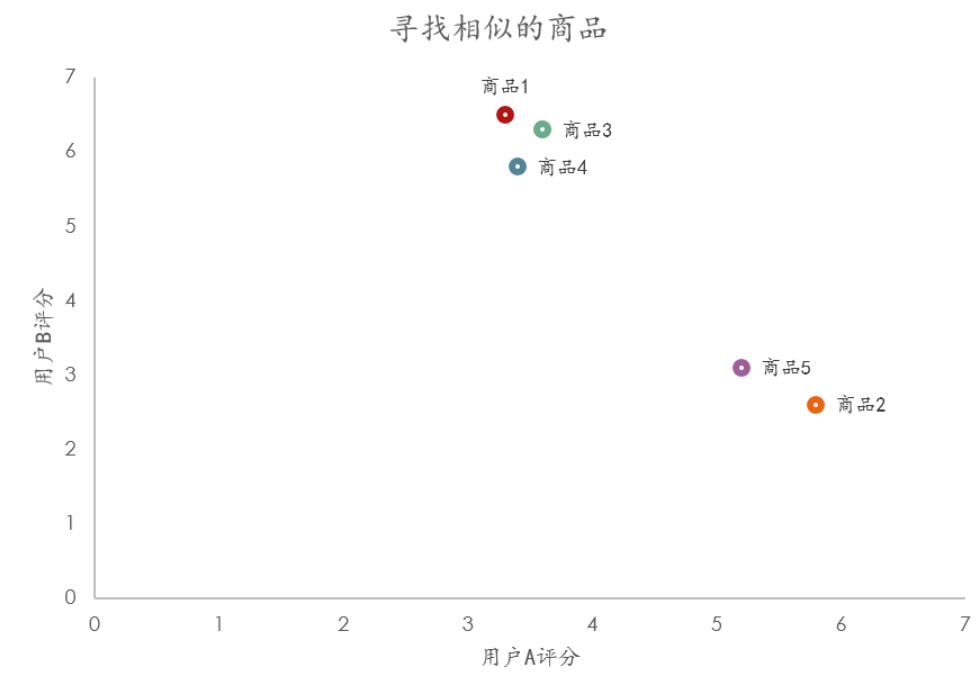


　　1.寻找相似的物品

　　表格中是两个用户对5件商品的评分。在这个表格中我们用户和商品的位置进行了互换，通过两个用户的评分来获得5件商品之间的相似度情况。单从表格中我们依然很难发现其中的联系，因此我们选择通过散点图进行展示。

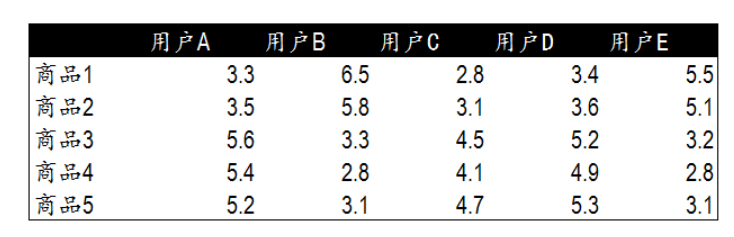


　　在散点图中，X轴和Y轴分别是两个用户的评分。5件商品按照所获的评分值分布在散点图中。我们可以发现，商品1,3,4在用户A和B中有着近似的评分，说明这三件商品的相关度较高。而商品5和2则在另一个群体中。



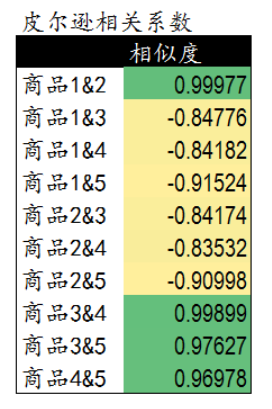
1、皮尔逊相关度评价

　　我们选择使用皮尔逊相关度评价来计算多用户与多商品的关系计算。下面是5个用户对5件商品的评分表。我们通过这些评分计算出商品间的相关度。



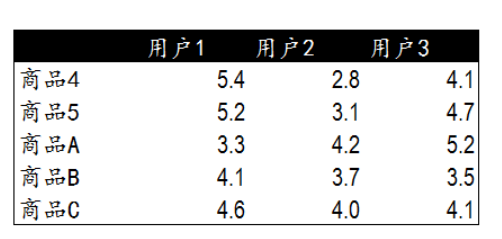
　　皮尔逊相关度计算公式

　　通过计算可以发现，商品1&2，商品3&4，商品3&5和商品4&5相似度较高。下一步我们可以依据这些商品间的相关度对用户进行商品推荐。

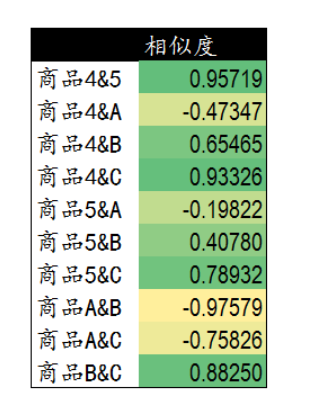


　　2,为用户提供基于相似物品的推荐

　　这里我们遇到了和基于用户进行商品推荐相同的问题，当需要对用户C基于商品3推荐商品时，需要一张新的商品与已有商品间的相似度列表。在前面的相似度计算中，商品3与商品4和商品5相似度较高，因此我们计算并获得了商品4,5与其他商品的相似度列表。

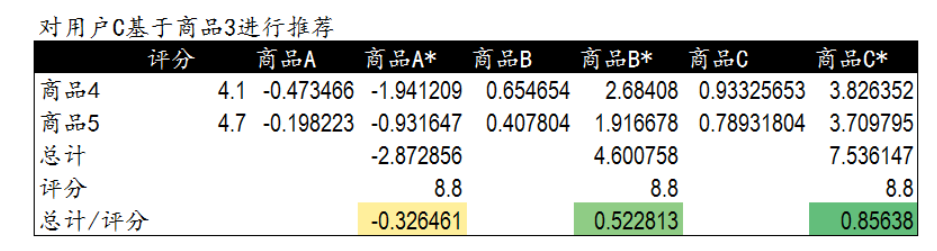


　　以下是通过计算获得的新商品与已有商品间的相似度数据。



　　加权排序推荐

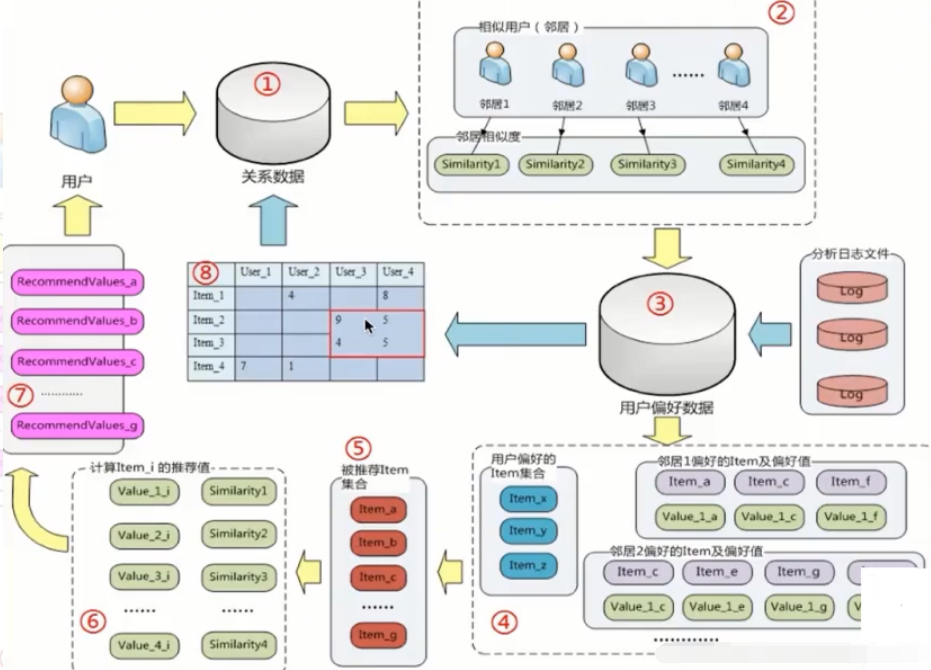
　　这里是用户C已经购买过的商品4,5与新商品A,B,C直接的相似程度。我们将用户C对商品4,5的评分作为权重。对商品A,B,C进行加权排序。用户C评分较高并且与之相似度较高的商品被优先推荐。



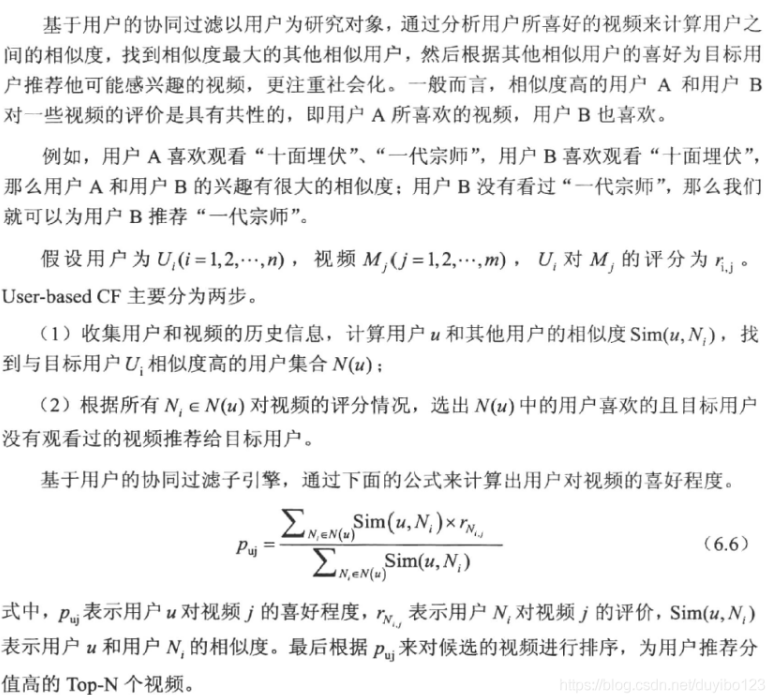
附录

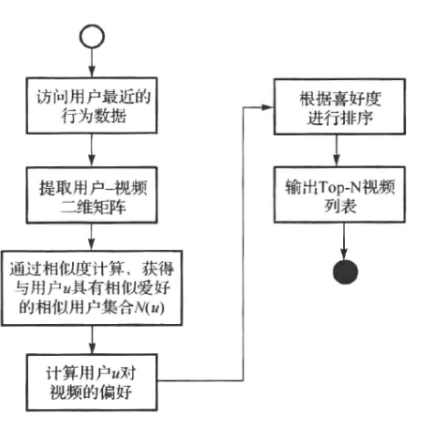
一、基于用户的协同过滤算法实现过程

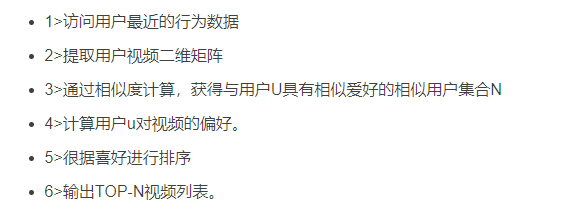
（1）模型图

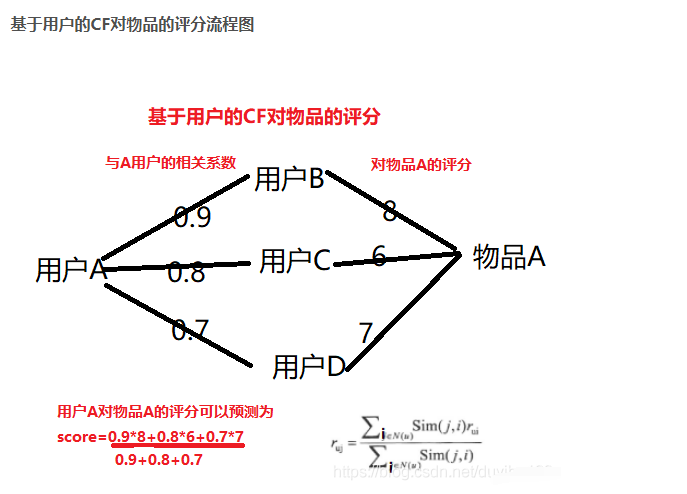


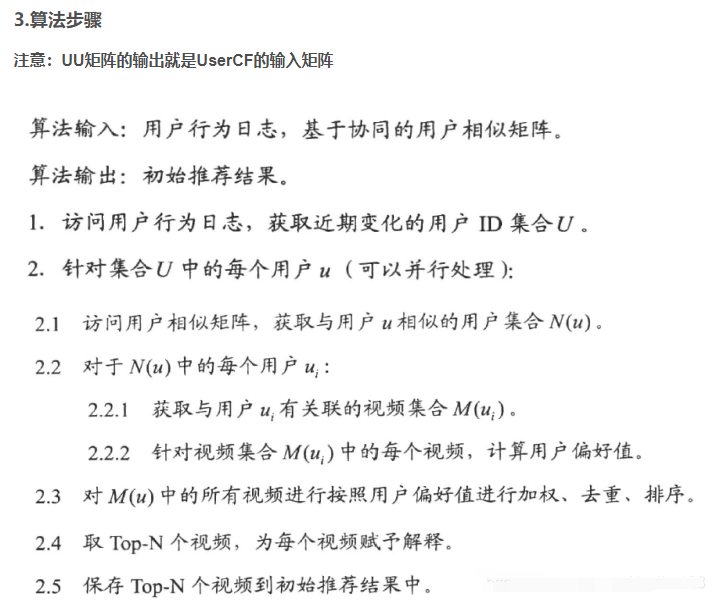
（2）算法原理及实现过程



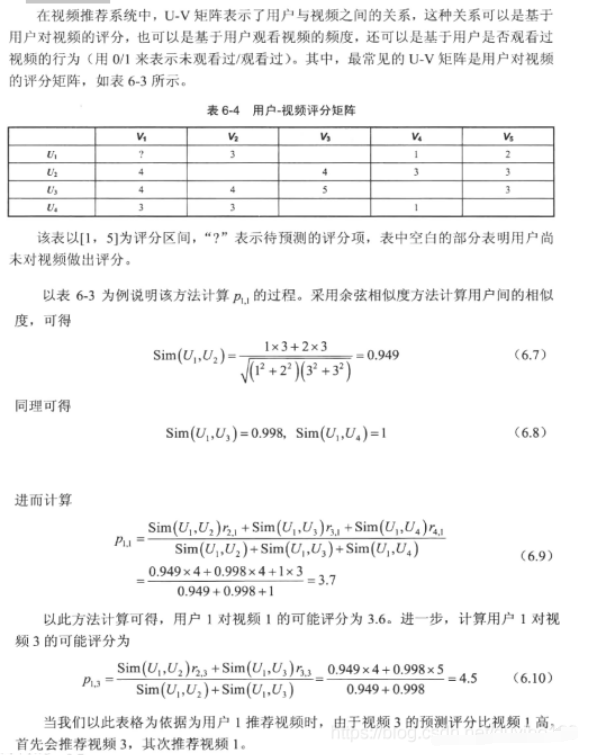




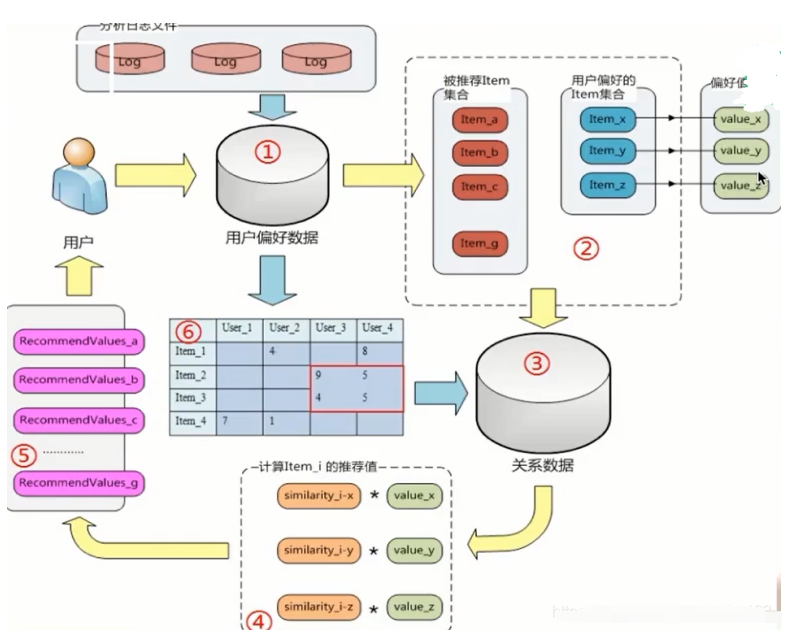


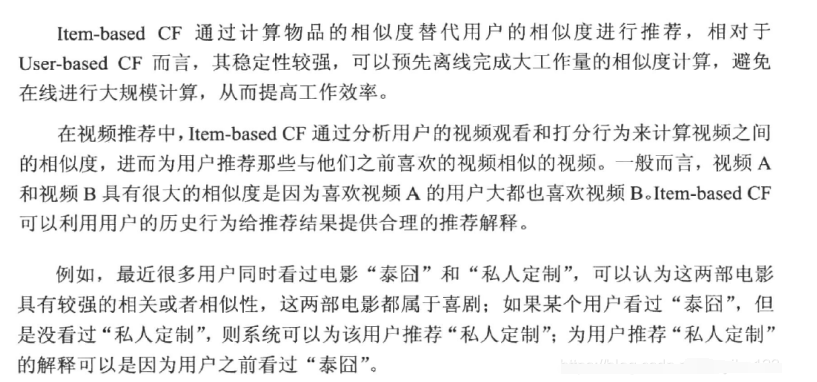


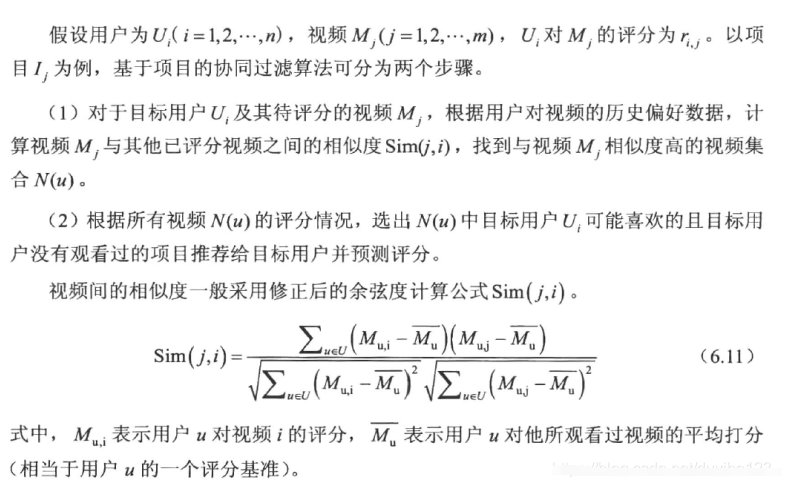
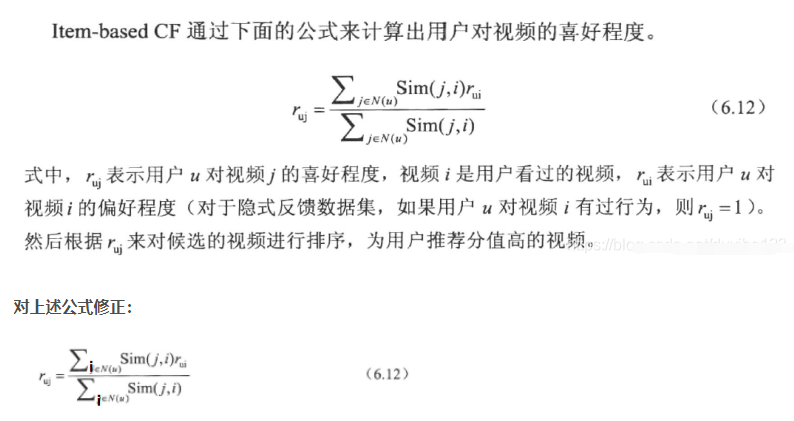
4、算法举例



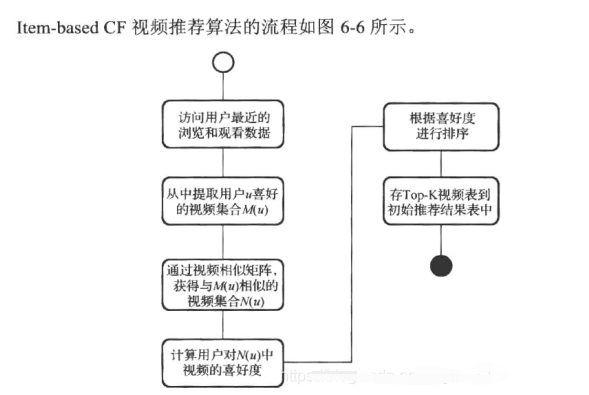
二、基于物品的协同过滤算法实现过程

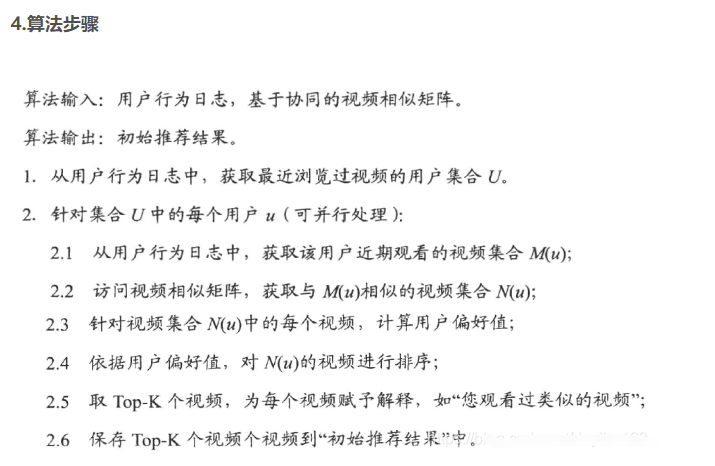


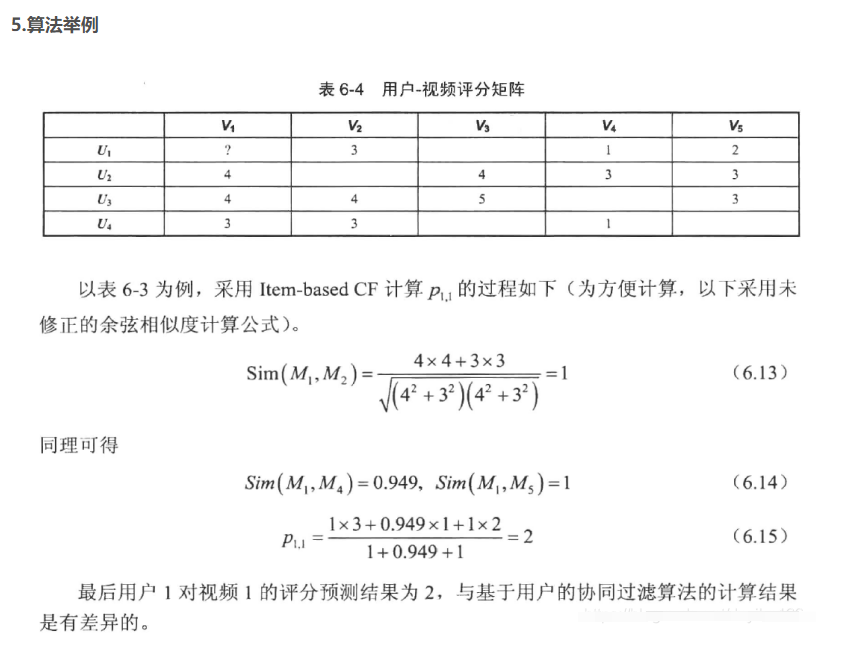


2、算法流程







注意：

本系统属于内部交流项目，未经允许不得对外，谢谢！