# 推荐系统第一次编程作业

**——实现基于用户的协同过滤算法**

## 1. 算法简介

### 1.1 简介

协同过滤算法是推荐系统的经典算法，在用户数多于物品数的情况下，基于用户的协同过滤算法相对基于物品的协同过滤算法更加有效。本次作业中我运用基于用户的协同过滤算法，使用uid、iid、score三个特征，测试了Pearson、Cosine、Jaccard三种不同的相似函数，通过网格测试找出了最合适的超参数与相似函数，对pred.csv完成了预测。

### 1.2 伪代码实现

基于用户协同过滤算法的伪代码如下所示：

function ucf\_predict(dataset,user,item,k,simularity\_function,predict\_function):

users = []

for user in dataset:

if user.scored(item):

users += [user]

//找到所有相似的用户

users.sortby(simularity\_function(users.element,user))

//根据相似度排序

users = users.maxk(k)

//根据超参数k保留相似度最大的k个用户

return predict\_function(simularity\_function,users,item)

//根据计算预测值的函数，用相似度提供权值，综合相似用户的评分得到结果

### 1.3 实现过程描述

首先运用pandas库中的read\_csv函数读入train.csv、test.csv、pred.csv三个文件，用DataFrame的筛选功能查找是否有脏数据，对数据进行简单筛选后，删去协同过滤算法不需要的评论和时间两列，只保留用户、物品、得分三列，输出用户id和物品id的最大值，为后续建立评分矩阵做准备。

根据数据集，建立元素为用户对物品评分的矩阵m，根据m可以得到用户-物品倒排表u2i和物品-用户倒排表i2u，这两个倒排表可以加速算法计算的过程。随后计算所有得分的平均值ms，这个值用于预测完全未知的物品和用户。再根据倒排表u2i建立用户-用户评分均值字典u2m，用于在算法中消除用户的评分个人偏见。这样实现基于用户协同过滤算法的一些基础的数据结构就准备完成了，可以实现具体的算法了。

首先实现三种不同的相似函数，相似函数以评分矩阵m、用户-物品倒排表u2i以及两个用户的uid为参数，计算用户相似度。我实现了Pearson、Cosine、Jaccard三种不同的相似函数，对于实现过程中的一些分母为零的异常情况，直接返回相似度最小值。余弦和Jaccard相似度的公式中只考虑正例，我把负例也加入了考虑，对于正例和负例分别计算相似度，然后取平均值得到最终的相似度。对于皮尔逊相似度，由于其中存在负值，和我选择的预测函数不相符，故我对皮尔逊相似度进行了正则化，映射到了(0,1)这个区间中。

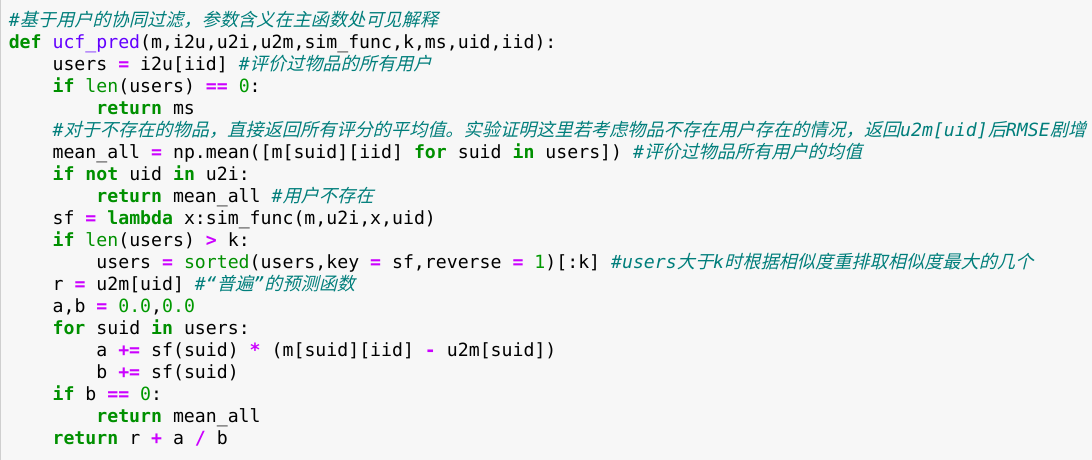
相似函数和数据结构准备完成后，就可以着手实现基于用户的协同过滤算法ucf\_pred了。ucf\_pred用到了上述的所有数据，对于不存在的物品和用户，直接返回所有评分的平均值，预测函数采用课件中所提到的“A common prediction function”，一个考虑了用户评分标准不同的根据相似度进行加权平均的预测函数。根据伪代码可以简单的完成这一算法。

算法的测试需要一个基准，故实现mean\_pred函数，对于任何询问直接返回购买了物品的所有用户的评分平均值，对于不存在的物品返回所有评分的平均值ms。理论上讲，推荐算法的均方根误差（RMSE）不得低于平均值预测。

最后我实现了evaluate函数，由于得分只有1～5的整数对于mean\_pred或ucf\_pred返回的值进行四舍五入后，根据测试集用RMSE评估误差。最后根据evaluate函数得到平均值的结果，并进行网格测试得到最优算法为相似用户数K=1情况下的Jaccard相似度。最后根据网格测试的结果填写pred.csv，得到my\_answer.csv。

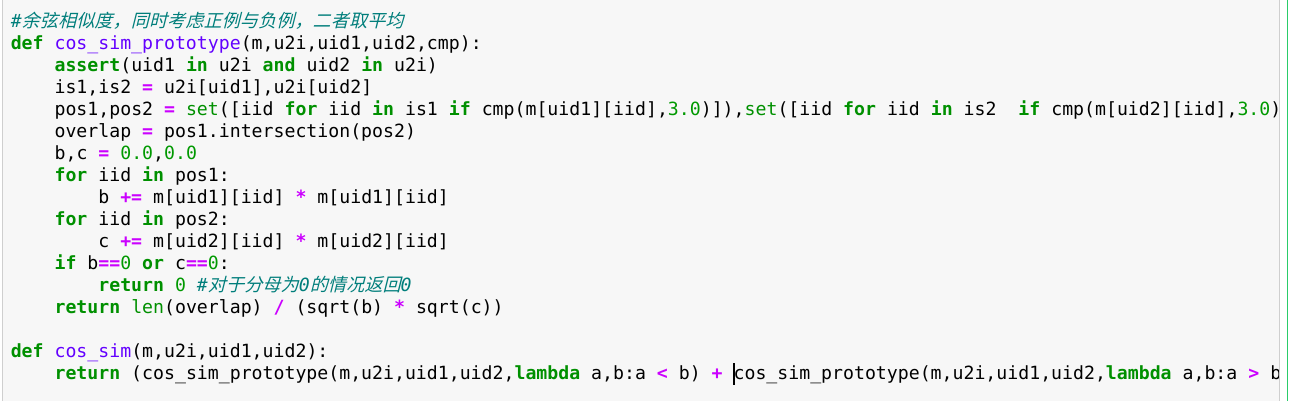
## 2. 核心代码注解

### 2.1 协同过滤算法



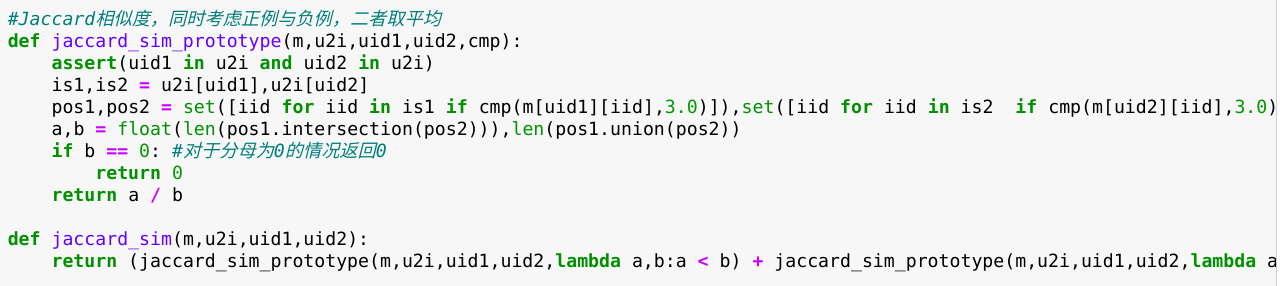
users表示评价过物品的所有用户，若users长度为0，那么显然iid是一个不存在于训练集中的iid，故直接返回平均值。lambda表达式sf意为对一个相似用户计算与需要预测的用户之间的相似度，用于重排users表与计算预测值中。

### 2.2 余弦相似度算法



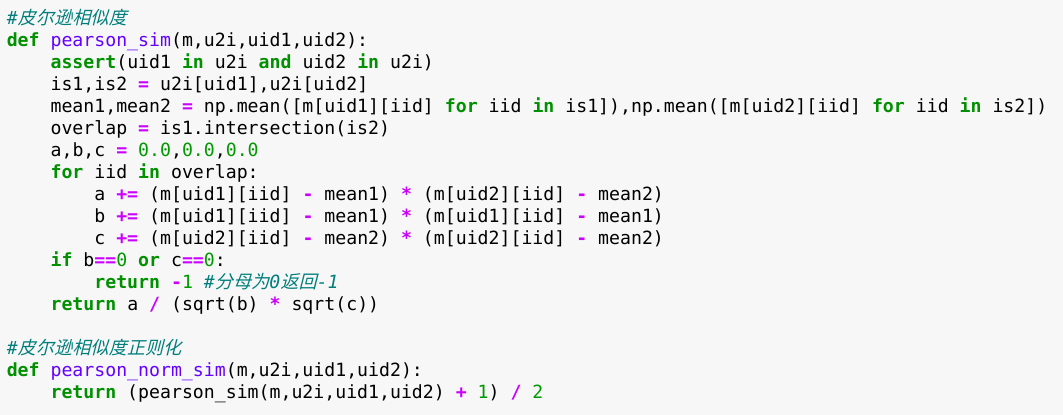
首先断言保证两个uid都是存在的uid。以3.0为正例和负例的分界点，cmp表示大于/小于，分出正例和负例，求出交集，根据公式计算余弦相似度，最后取平均。

### 2.3 Jaccard相似度算法



与余弦相似度算法思路类似，分出正例负例后根据公式计算，最后取平均。

### 2.4 Pearson相似度算法



mean1，mean2表示每个用户的得分平均值，overlap表示交集，根据公式计算，对于分母为0返回-1，计算完成后对Pearson相似度进行正则化。

## 3. 结果分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | 最佳RMSE | 超参数K | 优化比例 |
| 平均数（参照） | 1.2779839588573316 | 无 | 0% |
| 余弦相似度 | 1.126185012742946 | 12 | 11.9% |
| Jaccard相似度 | 1.121764644162269 | 1 | 12.2% |
| Pearson相似度 | 1.2319152471463943 | 3 | 3.6% |

因此 K=1的Jaccard相似度效果最好，使用此函数计算pred.csv。

## 4. 代码运行

### 4.1 环境

操作系统为Ubuntu 18.04 LTS，具体运行环境为Anaconda 4.8.2 附带的Jupyter Notebook。

### 4.2 依赖

Python 3.7

Pandas 0.25.1

Numpy 1.17.2

### 4.3 命令行命令

在Anaconda正确安装和配置后的的base环境中运行:

ipython notebook

运行后转到ipynb文件和数据文件所在的目录即可

## 5. 提交文件列表

### 5.1 RS\_HW1.ipynb

ipynb文件，其中包含三个有注释的python代码块，实现了基于用户的协同过滤算法，并且对pred.csv进行了预测，输出到my\_answer.csv中。

### 5.2 my\_answer.csv

pred.csv预测的结果，可用于教师作业评分

### 5.3 数据文件

train.csv、pred.csv、test.csv，供RS\_HW1.ipynb使用