**智能系统Lab2文档**

**17302010021 林晨**

## 代码基本结构

**基于用户和基于产品的协同过滤算法都采取首先计算相似矩阵的办法。**

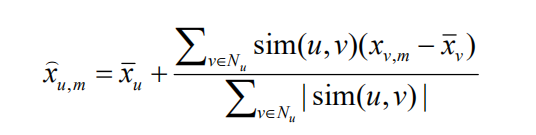
#### ①user\_based

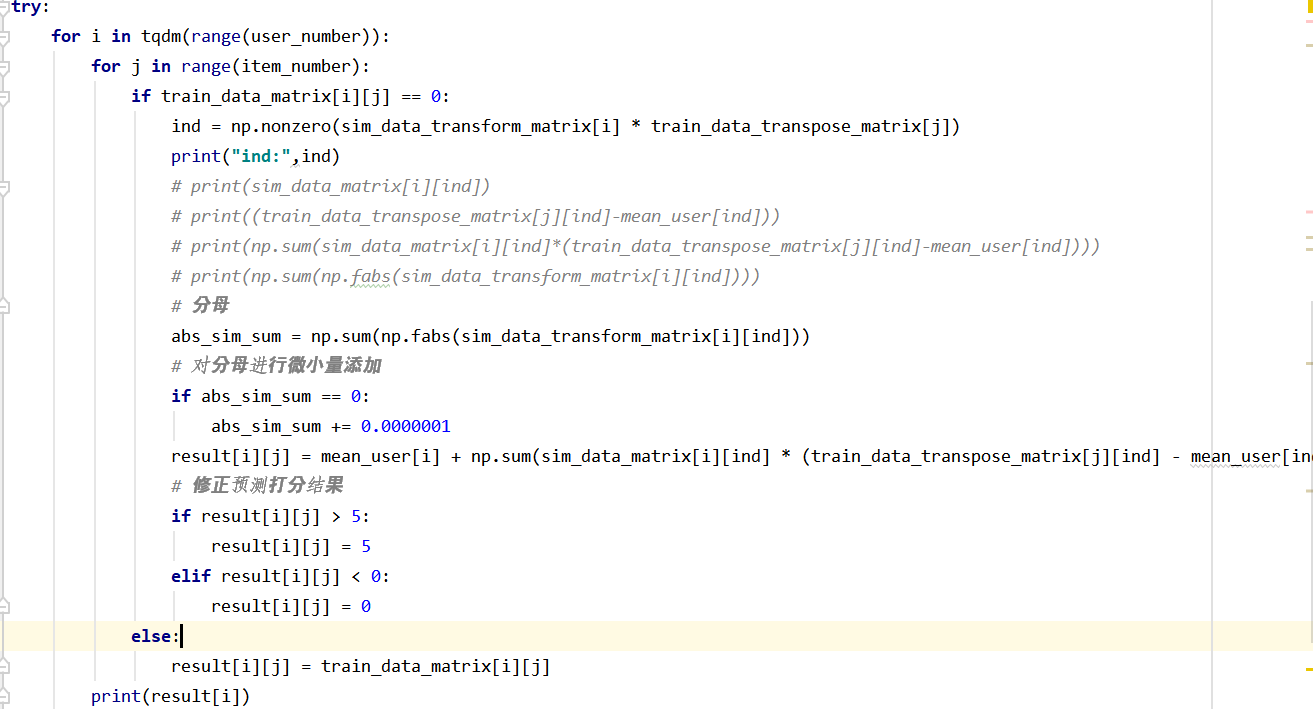
**cf\_user\_based\_sim.py：根据助教所给的计算相似矩阵的算法，进行改进优化形成相似矩阵并保存在文件user\_based\_similar.csv，方便下一步的操作。**

**cf\_based\_user：读取上述矩阵。由于助教表示对于user\_based可以不区分测试集和训练集，因而我取所有打分数据作为训练集。接着利用循环对于所有打分相似度在一定范围以上并且已经对相应产品（即电影）有评分的电影ID即下标索引。由于助教取消了rmse的计算因而未打印。如果要打印也简单，只要调用打印rmse函数即可。**

**然后循环使用如下公式：**

**最终写入predict\_user.csv。**





值得注意的是，我对于分母出现0的意外情况进行了讨论。分母出现零，一方面可是是由于冷启动问题，此时用0来表示后面的项。还有可能是u或者v的方差是0，也就是说，u，v这个用户对电影都分别是一个固定的打分（如u:3,3,3;v:4,4,4）--此时算出来相似度的分母是0。

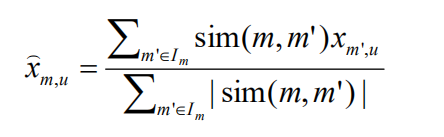
#### ②item\_based

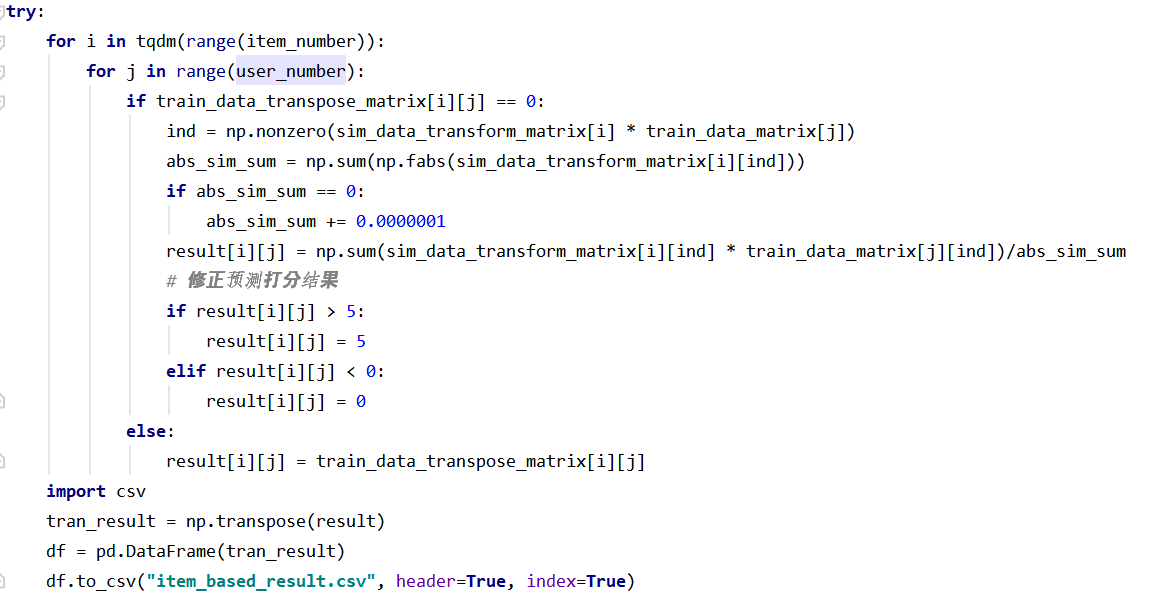
**cf\_item\_based\_sim.py：根据助教所给的计算相似矩阵的算法，去除其中的均值项，形成相似矩阵并保存于item\_based\_similar.csv，方便下一步操作。**

**cf\_based\_item：读取上述矩阵。由于助教表示对于item\_based可以不区分测试集和训练集，因而我取所有打分数据作为训练集。接着利用循环对于所有打分相似度在一定范围以上并且已经对相应有评分的用户ID，即下标索引。由于助教取消了rmse的计算因而未打印。如果要打印也简单，只要调用打印rmse函数即可。**

**然后循环以下过程：**

**结果保存在predict\_item.csv**





值得注意的是，我对于分母出现0的意外情况进行了讨论。分母出现零，一方面可是是由于冷启动问题，此时用0来表示后面的项。还有可能是m1或者m2的方差是0，也就是说，m1，m2这个电影相对每个用户都分别是一个固定的打分（如u:3,3,3;v:4,4,4）--此时算出来相似度的分母是0。

CF优点：

不需要任何的用户或者内容的属性特征信息

推荐效果的惊喜度、准确度、多样性较好

随着用户行为增多，推荐效果也会提升

CF缺点：

需要大量、准确的用户行为数据；

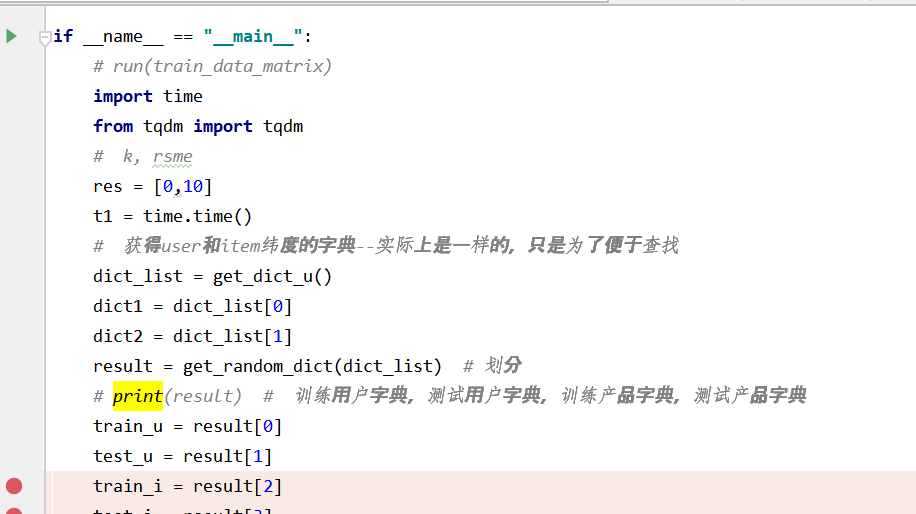
冷启动问题（新用户/物品没有行为，也找不到类似用户）

灰羊问题（个别口味特殊的用户找不到口味类似的用户）

因为采用历史行为数据，不易修改，不够灵活

覆盖率较低（数据稀疏问题，很多物品没有行为）

#### ③PMF-ALS



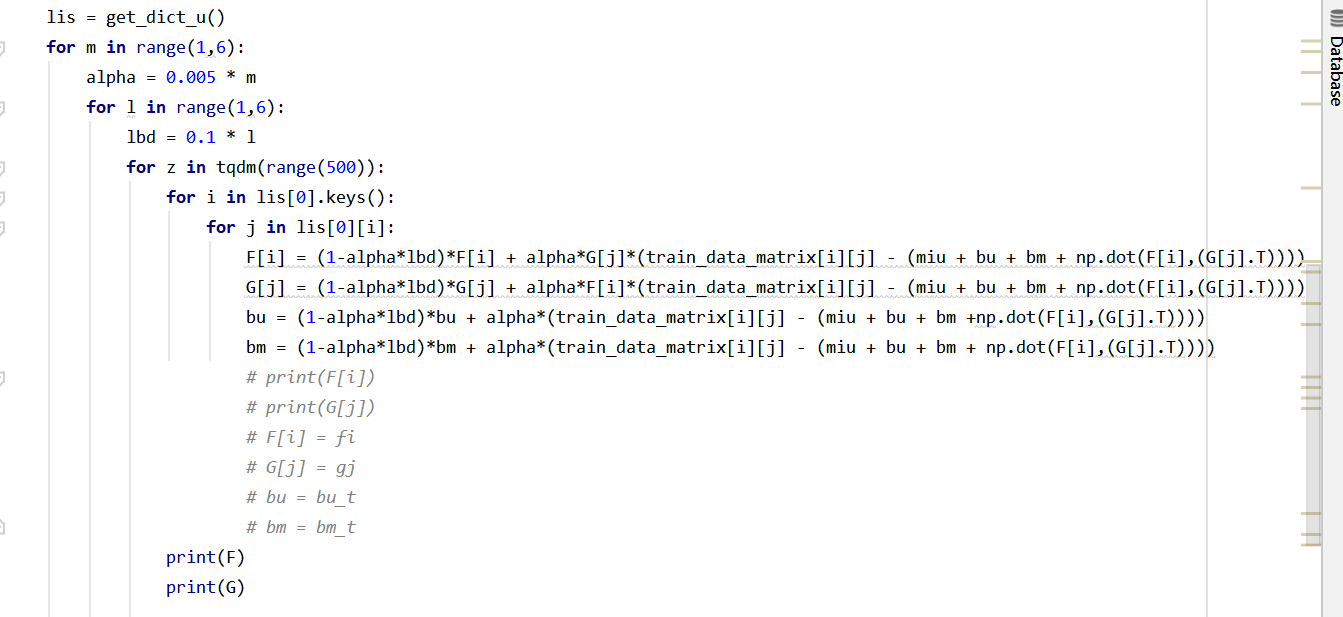
将数据划分为测试集和训练集按照2:8的比例

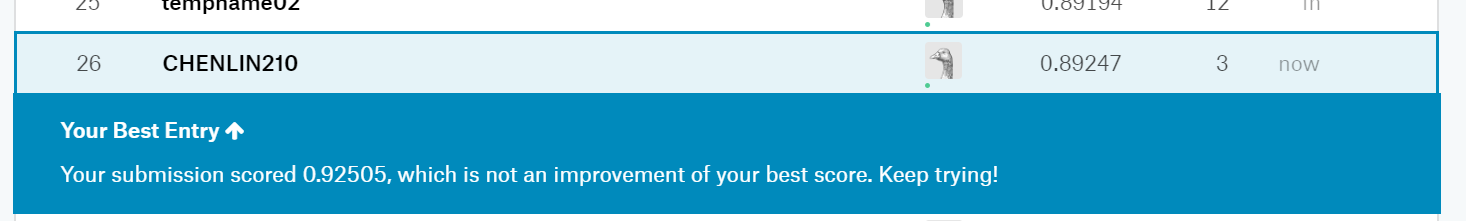
接下来为pmf具体实现：

**PMF.py：循环分解为1-50的的K值，利用训练集得到的F和G，分别求取其对于测试集合rmse的值，得到最佳K值。（同理，还有K值）**

④SVD++算法：

SVD++.py：根据SVD++算法我们初始化alpha和lbd，然后选取不同的值求借F和G的收敛值和打分矩阵值。最后lbd取0.3，alpha取1。rmse=0.88





# 参数、算法比较

①算法比较

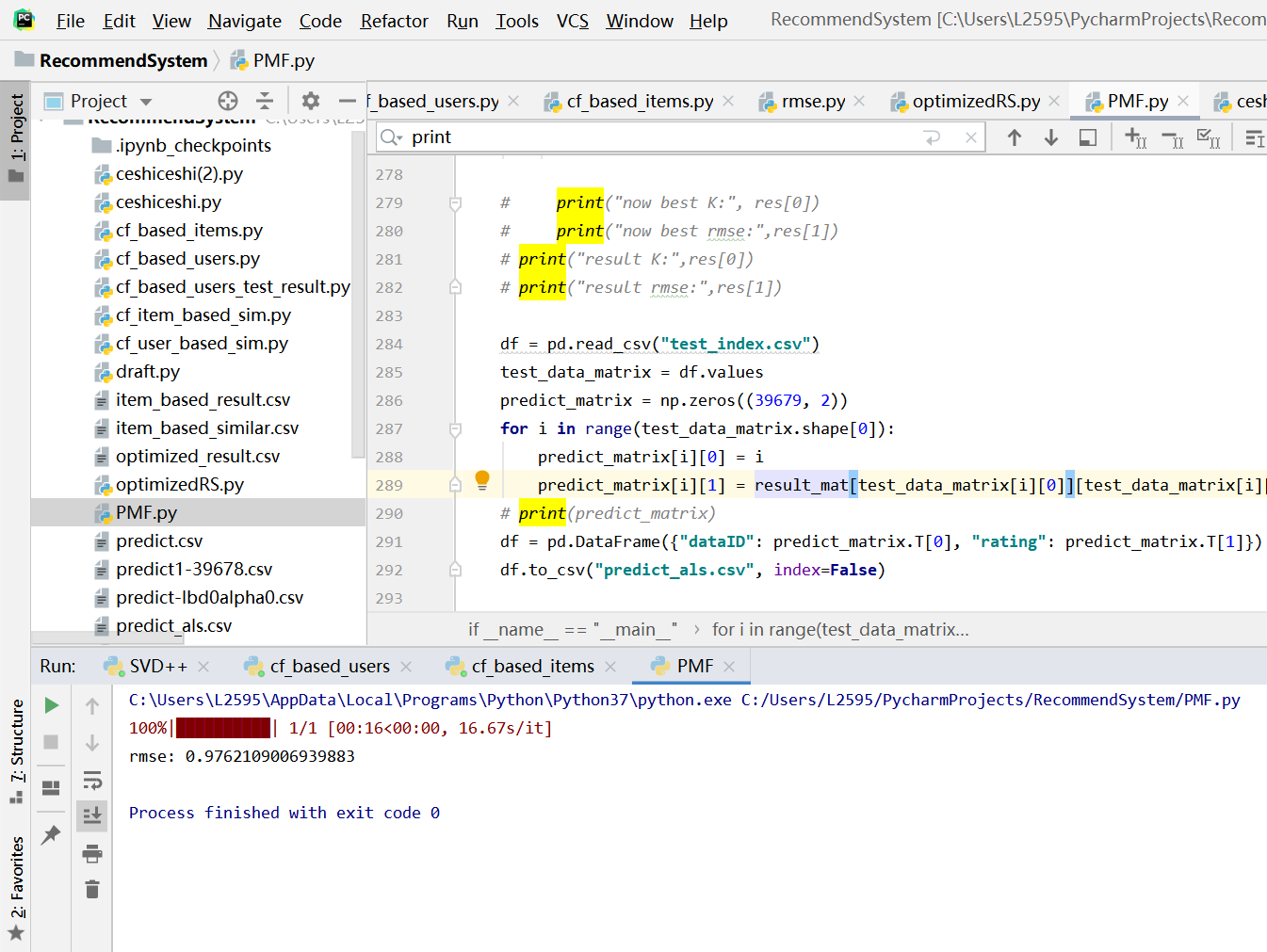
从目前的rmse提交测试结果来看，RMSE(item\_based) > RMSE(user\_based) > RMSE(ALS) > RMSE(SVD++)，分别约为1.2,1.1,0.97,0.88。虽然这也收到参数的影响，但是整体上还是呈现上述的rmse排名趋势。

#### ②参数比较

1、user:选取的最小的相似度为0.2左右（sim>= 0.2）得到的rmse相对来说是比较小的，但是这个波动并不大，根本上是算法本身的限制（还有数据集中部分用户评分过少等原因）。

2、item:选取所有的item进行计算，得到的rmse值本身较大，变化非常小，参数调整意义不大。

3、ALS:使用K的变量组合选取取得最佳的K值。最终确定K取11效果不错（lbd在0.2左右）。因为其rsme相对最小，约为0.976结果如下（为缩短文档长度，取了前14条K的数据，lbd同理使用循环求最佳）：



born rmse: 0.9771560318741384

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 0.984755183575681

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 1.0128120633227335

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 1.0491261732166681

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 1.0785924776799825

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 1.0959014568735463

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 1.1227138017414493

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 0.9776657037287566

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 0.9771682188927695

now best K: 1

now best rmse: 0.9771560318741384

born rmse: 0.976536498864334

now best K: 10

now best rmse: 0.976536498864334

born rmse: 0.9764728746242117

now best K: 11

now best rmse: 0.9764728746242117

born rmse: 0.9765590015422806

now best K: 11

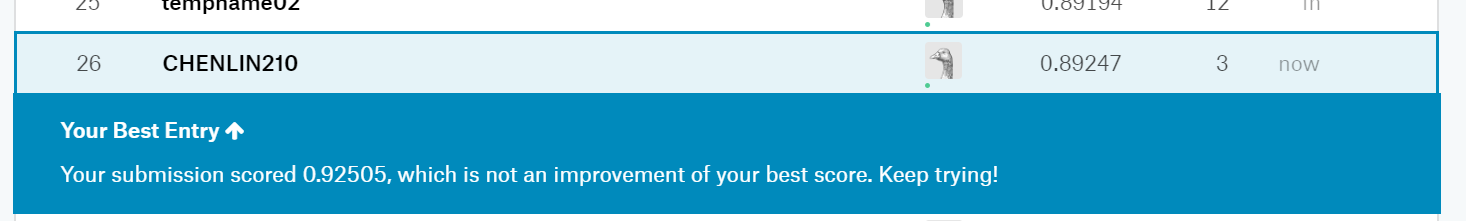
now best rmse: 0.9764728746242117

born rmse: 0.976760009400246

now best K: 11

now best rmse: 0.9764728746242117

4、SVD++：使用循环变化参数求得不同的rmse，由于使用助教给的网站测试rmse即可，最终在lbd为0.3，alpha为1左右。



# 改进思路、实验

①减小参数的步长。可以选取0.001的步长来求取最佳参数，这样得到的rmse会更小，但是要花很多时间运行。

②梯度下降算法：可以尝试一下看看是否会得到更小的rmse

③块处理：ppt中对于网络中大量数据使用块处理，对于大量数据可能更为有效，可以一试。

④基于内容的推荐（Content-based Recommendation）是信息过滤技术的延续与发展，它是建立在项目的内容信息上作出推荐的，而不需要依据用户对项目的评价意见，更多地需要用机 器学习的方法从关于内容的特征描述的事例中得到用户的兴趣资料。在基于内容的推荐系统中，项目或对象是通过相关的特征的属性来定义，系统基于用户评价对象 的特征，学习用户的兴趣，考察用户资料与待预测项目的相匹配程度。用户的资料模型取决于所用学习方法，常用的有决策树、神经网络和基于向量的表示方法等。 基于内容的用户资料是需要有用户的历史数据，用户资料模型可能随着用户的偏好改变而发生变化。--不适合这个lab

⑤基于效用的推荐（Utility-based Recommendation）是建立在对用户使用项目的效用情况上计算的，其核心问题是怎么样为每一个用户去创建一个效用函数，因此，用户资料模型很大 程度上是由系统所采用的效用函数决定的。基于效用推荐的好处是它能把非产品的属性，如提供商的可靠性（Vendor Reliability）和产品的可得性（Product Availability）等考虑到效用计算中。--需要更多的数据

改进实验：调节ALS和SVD的lbd--结论：效果一般

梯度下降算法--结论：效果差不多

结论--要想rmse小雨0.8很难，**我们需要更多的信息。**