大数据分析第八次作业 推荐算法

201250203 陈张熠

此程序运行时间较久，在我电脑上运行需要大约25分钟。
（所以如果不愿意跑的话，我可以保证跑出来的与我提交的文件一定一致）

一、处理数据

通过pandas读取rating和movies数据，将他们以“movieId”为线索整合成一个新的数据（类似于mysql中的连接）。此后，将其整合成一个字典形式格式，一个userId对应多个movieId，而每个movieId对应一个此用户所给的评分。对于每一个movies数据中的genres，将其中的tag进行切分，生成一个新的属性gene（将所有切分后的tags放入其中）。以上为所有数据处理过程。

二、算法描述

我所使用的推荐算法是以欧式距离作为用户相似度，以及将余弦距离作为内容相似度的混合过滤协同推荐算法。

1. 用户相似度

欧式距离的优点就是比较简单，此问题中因为我只对于用户对电影的评分进行相似度计算，随意采用欧式距离来计算用户间（对电影评分的）的相似度。流程如下：

对于任意两个用户，首先找到他们都看过的电影，对于这些电影两人都有一个评分，然后计算他们之间的距离，如下。

$$dis(X,Y)=\sqrt{\sum(x_i-y_i)^2}$$

对于距离越小的两个用户，他们之间的相似度应该越大（因为他们对于相同电影的评分是非常相近的，说明品味类似）。在代码中，体现为一个名为cal的函数，其中该函数的返回值越大，代表其相似度越高。如此对于每个用户，我们都可以找到与此用户相似程度最高的一批人。（代码中体现为similar函数）对于这批人所看过的电影，我们选择其中没有被目标用户所看过的电影，我们就得到了一个电影list。

2. 内容相似度

首先对于每个用户，我们可以计算一个19维的用户向量，其中每一维度代表一个类型的电影（来自于movies中的genres属性切割过后），而每一维度的数据大小（正为喜爱，负为讨厌）代表此用户对这个类型的喜爱程度。

为了完成这个内容，我们首先需要计算用户评分所给出的总平均值，然后对于每一个维度（即每一类型）去寻找此用户对于这个类型电影的评分，并计算平均值。将每一个类型电影的平均值减去总平均值就是用户向量每一维度的大小。

此后，我们根据在用户相似度中得到的电影list，将其中的每一个电影也变成一个19维电影向量，每一维度的大小只为 0 或 1 。1代表此电影属于某个特定电影类型，而0代表此电影不是某个特定类型。

最后，我们通过计算 用户向量 和 电影向量 的余弦距离，作为相似度，相似度越高，则认为用户更喜爱此类型的电影。最后进行排序，得到推荐电影。

$$dis(X,Y)=\frac{\sum X_iY_i}{\sqrt{(\sum X_i^2)}\sqrt{(\sum Y_i^2)}}$$

对于predict函数，参数默认值为2，意思为对于每个用户推荐2个电影，但是可以通过改变参数来改变推荐电影数量。