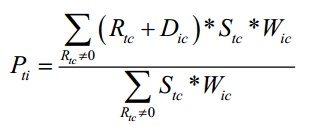
**UserBasedSlopOne 算法实现**

陈秋良 李正 卢晓燕

根据 A Creative Personalized Recommendation Algorithm——User-based Slope One Algorithm 这篇论文，使用Matlab编程完成。

主要算法分为4个步骤：

* 计算Item之间的平均评分差值矩阵D，并得到同时给Item i和j打分的用户个数矩阵W
* 计算用户相似度矩阵Sim，根据余弦相似度
* 计算S矩阵，Sti表示给item i打分的其余用户和用户t的平均相似度
* 计算用户对item的打分估计值。



当然在实现的时候还需要考虑读取数据集等问题。

**最重要的一点是**，在Matlab中如何优化代码，提高运行效率。

在优化以前，由于整个算法的复杂度是O（m\*n\*n）-- m 为用户数，n为物品数，在matlab中运行非常慢，因为直接写代码会有三重循环，基本上要**20个小时**左右才能出结果。

优化的方法是：

1. 利用matlab和c混合编程，将大量循环操作放到c中完成。这里主要利用mex函数接口。具体见代码。
2. 尽量用矩阵运算，转置、点乘等代替不必要的循环操作。
3. 计算Sim矩阵时候，利用repmat函数优化时间。

提交代码说明：

1. MovieLens.mat是训练数据集，主要是user-item的评分矩阵。
2. MatrixData2.mat是训练的结果，保存了上面的R, D, W, Sim, S矩阵。
3. UserBasedSlopeOne.m 是训练的代码，主要实现上面算法的四个步骤。

使用方法： p = UserBasedSlopeOne(userID, itemID,file); 给user和item的ID，以及mat文件如’MovieLens.mat’，自动训练并输出评分预测

1. UserBasedSlopeOne2.m是利用训练好的matrixData2.mat的数据进行预测。

使用方法：p = UserBasedSlopeOne2(userID, itemID); 给user和item的ID，输出对应的评分预测

5. MAE和RMSE文件，给定预测值和目标值，输出计算误差。

6. test.m 文件，调用MAE和RMSE计算误差。

7. dataset.mat 测试数据文件

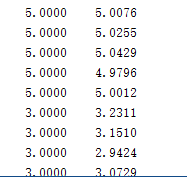
8. GetTheAveSimMat.c、GetTheDiff&WeightMat.c和GetTheDiffMat.c是优化的c代码。用来和matlab协作。

9. GetTheAveSimMat.mexw64、GetTheDiff&WeightMat. mexw64和GetTheDiffMat. mexw64是c代码生成的mex函数，供matlab调用。

**实验和测试：**

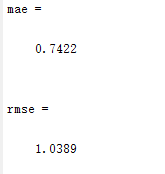
利用训练好的数据进行预测，结果如下：

实际值 预测值



**误差分析：**

进行100个预测，与实际结果对比，计算误差：



结果表明该算法的有效性。

最后，感谢TA哥指导！

2015.6.1