## 基于MovieLens 的电影推荐

### 201250104 苏致成

### 算法选择

1. 推荐系统的主流算法是协同过滤，分为基于用户的最近邻协同过滤和基于物品的协同过滤。本系统采取基于用户的最近邻协同过滤。
2. 因为电影的标签原因，系统还实现了 content\_based 的算法。但因为仅根据标签判断内容相似度并不十分准确，因此在实现中仅将其作为辅助手段。

### 算法流程

以下括号内为对应函数名称。

1. 数据预处理，如去除重复值等等。
2. 基于 content\_based 的推荐。( contentBased() )
   * 获取用户最喜欢的前10部电影，根据下述算法获取与该电影最接近的电影。对每部电影获取一部最相似的电影。
   * 上述获取一部最相似的电影算法如下：根据电影类型进行 tfidf 分析得出，对电影之间两两计算内容相似度。相似度最高进行记录。( tfidf() ）
   * 如果得到的结果中含有用户已经看过的电影，则剔除。
   * 根据相似度返回5部电影，返回的结果才能进行 “基于用户的最临近协同过滤”。
3. 基于用户的最临近协同过滤。（ userCF() ）
   * 获取用户对电影的打分矩阵。( get*user*sim\_matrix() )
     + 考虑用户主观打分的情况，因此对打分进行标准化（去均值）。
     + 计算两用户打分的余弦值作为评判相似度的标准。
     + 获取前 个相似用户。( userSim() )
   * 根据以下公式计算该用户对某部电影的预测分数：( get\_predict() )
   * 参数含义如下：
     + 为用户 评分的向量， 为用户 对电影 评价的分数。
     + 为与 最相似的，对 评分的 个用户的集合。
     + 表示用户 和用户 之间的相似度。
   * 针对某一目标用户，为其预测所有电影的评分，并且若其看过某电影，则最后处理将其预测分数置为0 。
   * 排序并获取评分最高的一部电影。
4. 上述结果处理：
   * 只推荐 “基于用户的最临近协同过滤” 评分最高的一部电影给用户。( save() )

概括：

首先需要经过基于内容的推荐，选择和用户喜爱的电影中最为相似的5部电影。电影只有通过该步骤过滤方可进入下一步骤。

对上述的过滤结果采取基于用户协同过滤对每个用户推荐的一部电影。

### 算法结果

结果见 csv 文件。