**亚马逊电影推荐**

张快 2120151061

胡建 2120150993

季鹏 2120150998

# 简介

推荐系统（RS）是一种软件工具和技术方法，它可以向用户建议有用的物品，这种建议适用于多种决策过程，如购买什么物品、听什么音乐、在网上浏览什么新闻等。

“物品”是用来表示系统向用户推荐内容的总称。一个推荐系统通常专注于一个特定类型的物品（如CD或新闻），因此它的设计、图形用户界面以及用于生成建议的核心的推荐技术都是为特定类型的物品提供有用和有效的建议而定制的。

推荐系统主要针对的是那些缺乏足够的个人经验和能力的人，他们无法评估潜在的大量可供选择的物品，比如，某个网站提供的商品。一个典型的例子就是图书推荐系统，它帮助用户挑选一本书来读。在亚马逊，网站采用个性化推荐系统为每个用户进行推荐。由于推荐系统通产是个性化的，不同的用户或用户组接受的建议是不用的。当然也存在非个性化推荐。他们都是非常简单的，通常出现在报纸或者杂志上。典型的例子包括书籍和CD等的top10推荐（最热销的前10名）。虽然在某些情况下它们可能是有用和有效的，但这些类型的非个性化推荐通常不是推荐系统研究要解决的问题。

个性化推荐最简单的形式是提供一个排好序的物品列表。用过这个排序列表，推荐系统试图根据用户的偏好和其他约束条件来预测最合适的产品或服务，为了完成这样的计算任务，推荐系统收集用户的喜好，这种喜好是显式的，如为产品打分，或通过解释用户的行为做出推断例如，推荐系统可能会把访问某个特定商品详情页的行为作为该用户喜爱这个主页上的商品的隐式信号。

本次项目运用现有的几种推荐方法对亚马逊网站上的用户进行电影推荐，并比较几种推荐方法的好坏。

# 问题陈述

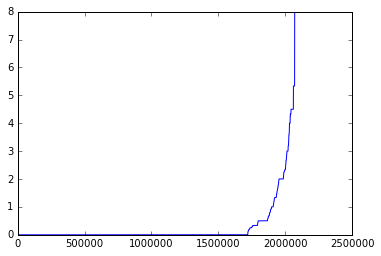
本次项目的数据集来自<https://snap.stanford.edu/data/web-Amazon.html>，总共有用户电影数据4607047条，其中包含用户200多万个，电影17万左右。

对于每个评价条目包括用户id，电影id以及用户对电影的评分（1-5）分，我们要做的工作就是用过已有的数据来预测用户对新的物品的评分。

# 解决方案

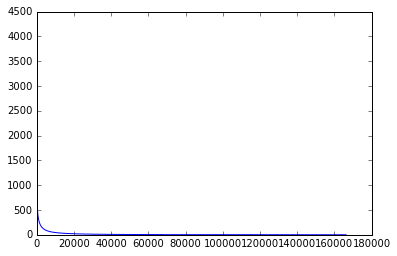
## 数据预处理

数据取自<https://snap.stanford.edu/data/web-Amazon.html>，总共有用户电影数据4607047条，对最初的数据进行了一下初步的分析:



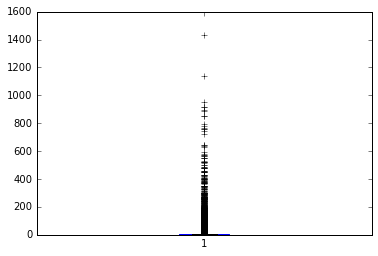
**用户评分方差图**

通过统计，在所有的200多万的用户中，有大约175万的用户要么只有一条评分数据，要么对所有电影的评分都一样，我们认为这样的数据不利于做预测，于是舍弃掉了这一部分数据。



**电影评分数量曲线**

如图，整个电影数据集当中总共有大约17万部电影，但是大部分的评论都集中在前2万部电影当中，在实验当中我们取了前5000部电影最为item集合。



**用户评价数目盒图**

通过统计每个用户所看的电影的数目，发现有一些用户的评论个数太多，我们怀疑存在有多个用户共享一个账号的情况，于是舍弃掉了这一部分用户，最终我们取了评论数目在10到100的用户。

最后我们取80%的数据做训练数据，20%的数据做测试数据，最终的数据集为：总共334784条评分数据，用户15090个，电影5000部，训练集一共273762条数据，测试集一共61021条数据。

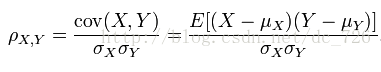
## 实现方法

### 平均分预测（基准方法）

直接用训练机上用户的平均分作预测，这里不再赘述。

### 基于用户的协同过滤

基于用户的协同过滤的主要思想就是找评分最相近的用户，常用的相似度测量方法有余弦相似度，调整余弦相似度，皮尔森相似度。这里我们使用皮尔森相似度进行用户相似度的测量，其公式如下：



具体实现方法是对于测试集里面的每个用户，通过计算他和训练集用户的皮尔森相似度来找与其最相似的用户（并且用户对该电影有评分记录），然后用训练集用户的评分当作测试集用户对电影的评分

### SVD

矩阵因子分解模型吧用户和物品两方面的信息映射到一个维度为f的联合隐语义空间中，因此用户-物品之间的交互作用被建模为该空间中的内积。这个隐语义空间试图通过描述物品和用户在各个因子上的特征来解释评分值，而这些因子是从用户反馈自动推出的。例如，如果物品是电影，因子将会用来度量如喜剧或悲剧、情节的数量或者面向儿童的等级这些明显的维度，以及如性格发展的深度或者“突变”等隐式维度，甚至是完全无法解释的维度。

相应地，每一个物品i都与一个f维向量相关，每一个用户都与一个f维向量相关。给定一个物品i，q，向量的维度值代表了该物品拥有这些因子的程度（例如，某部电影的搞笑因子程度为5，而恐怖因子程度为1），其取值大小反映了物品拥有这些因子的积极或者消极程度。给定一个用户u，向量的维度值代表了用户对这些因子的偏好程度（例如，某用户对搞笑因子的偏好程度为1，而对恐怖因子的偏好程度为0.1），同样地，这些值的大小反映了用户对这些因子的积极或者消极的评价。点积记录了用户和物品之间的交互，也就是用户对物品的总体兴趣度。加上之前提到的只依赖于用户或者物品的基准预测可以得到最终额评分。因此，评分通过下面的规则预测得到：



为了学习模型中的参数，也就是，我们可以最小化以下正则化的平方误差：



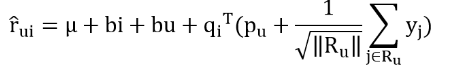
最小化过程一般通过随机梯度下降算法或者交替最小二乘法来实现：

其中预测误差。

### SVD++

考虑隐式反馈信息可以增加预测准确度，这些隐式反馈信息提供了用户爱好的额外指示。这对于那些提供了大量隐式反馈但是仅仅提供少量显式反馈的用户尤为重要。正如之前解释的那样，即使在独立的隐式反馈缺失的时候，我们也可以通过考虑用户评分的物品来得到用户兴趣的信息，而无需考虑这些物品的评分值。这样就出现了几种根据用户评分的物品来对用户的某个因子（用户对某个因子的喜好程度，比如，电影评分记录中用户对喜剧电影的喜爱程度因子）建模的方法。

为了达到这个目的，SVD++增加了第二个物品因子集合，即为每一个物品i关联一个因子向量。这些新的物品因子向量根据用户评分的物品集合来描述用户的特征。确切的模型如下:



其中R(u)包含用户u评分的所有物品，y为物品的隐式特征。更新过程如下:

## 实验结果

所有的方法采用RMSE来判断好坏。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | Standard | User-based | SVD | SVD++ |
| RMSE | 1.384 | 1.126 | 1.026 | 1.033 |

从实验结果可以看出，基于矩阵分解的方法要明显好于其他方法，但是SVD和SVD++的效果差不多。