**2019年第九届国际教育和社会科学会议(ICESS，2019年)**

**个性化推荐系统中协同过滤推荐算法的研究**

彝人1,a徐精科1,b黄杰1,c和CuionChi1,d

1.中国辽宁沈阳建筑大学

a.1181134429@qq.com；b.1401308379@qq.com；c.1269028746 @qq.com；d.726733255 @qq.com

关键词：推荐系统、协同过滤、个性化推荐

**摘要。**在信息超载的时代，推荐系统出现在历史时刻。协同过滤推荐系统算法是最成熟、最广泛的推荐技术。本文模拟了电影推荐的场景，在协同过滤推荐系统下应用了不同的推荐算法，并对推荐质量进行了比较，得到了一个相对客观的实验结论。

# 介绍

随着信息技术和互联网的发展，信息过载给用户和生产者带来了巨大的挑战。创建推荐系统是为了处理用户需求不明确和长尾信息等问题。推荐系统通过分析用户的历史行为来模拟用户的兴趣，从而推荐符合用户兴趣和需求的[1,2]信息。目前，协同过滤推荐系统已被应用于亚马逊、京东商城、淘宝、toutiao.com等各种电子商务平台上。网飞公司是电影推荐领域最成功的公司之一。它将基于项目的推荐算法应用于协同过滤推荐算法中。由于其成功的推荐系统，网飞公司创造了巨大的收入增长。本文针对三种流行的基于模型的协同滤波推荐算法，将其应用于模拟电影评分环境并进行分析。这些包括：斜坡一，SVD，SVD++。协作过滤背后的基本思想是，如果用户在过去就有偏好，那么他们在未来就会有类似的偏好。基于此，如果两个用户的偏好重叠很多，他们可以向[4]推荐项目。这种技术被称为协作过滤，因为对感兴趣的项目的选择来自于过滤来自大量集合的结果，并且用户隐式地与他人协作。目前，协同滤波推荐算法大致可分为基于内存的协同滤波算法和基于模型的协同滤波算法[5]两类。其中，基于内存的协同过滤推荐算法被划分为最近邻推荐

基于用户和基于项目的最近邻推荐。

# 协同过滤推荐算法

**基于用户的最近邻推荐。**

通过使用目标用户的历史行为数据来了解用户对商品的偏好。然后，找到过去与目标用户有相似偏好的其他用户，即找到目标用户最近的邻居。对于目标用户没有看到的项目i，预测值由其邻居i的得分计算。如果结果为阳性，则将推荐目标用户使用[6]。

# 基于项目的最近邻建议。

项目之间的关系可以通过不同用户对不同项目或其他行为（如添加愿望清单、与他人分享、评价等）的评价来量化，以便找到项目最近的邻居。如果目标用户对项目i和项目有积极的态度

**版权所有：©（2019），弗朗西斯学术出版社，英国**

**307**

**DOI： 10.25236/icess.2019.059**

j是项目i的邻居，然后向目标用户推荐项目j。在具体的实践中，用户对项目的态度将被量化，从而作出更准确的建议，[7]。

# 基于模型的推荐算法。

基于模型的建议开始离线处理原始数据，例如使用一些降维技术[8]。在运行时，只需要预测或“学习”模型就能够预测。

# 电影推荐系统的推荐算法

在2009年的网飞竞赛中，更多的团队使用了基于模型的推荐算法，有效地提高了推荐系统[9]的预测精度。本文介绍了三种常见的基于模型的推荐算法，并将其应用于实验。实验数据是奈飞公司竞争对手的3万多名用户的电影收视率。本文将训练集和8：2的比例对训练集和测试集进行划分。以RMSE（均方根误差）为评价指标。RMSE值越小，预测质量[10]越好。

# 斜坡一算法。

算法思想：采用一个简单的线性回归模型进行预测。算法示例：下表1是四个用户对三部电影的评分，从而预测用户1对电影3的评分。

表1四个用户对三部电影的评分

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***电影1*** | ***电影2*** | ***电影3*** |
| 用户1 | 3 | 5 | ? |
| 用户2 | 2 | 4 | 5 |
| 用户3 | 4 |  | 3 |
| 用户4 | 3 | 4 | 4 |

计算电影3和电影1之间的平均距离：（3+（-1）+1）/3=1，以及电影3和电影2之间的平均距离：（1+0）/2=0.5。根据用户1用户1对电影1的评分，电影3的预测评分为4。根据用户1对电影2的评分，电影3的预测评分是4.5。考虑到电影3和电影1的同时评分为3，电影3和电影2的同时评分为2，预测用户1对电影3的评分为：



在推荐的过程中，为了防止用户习惯性地给高分来影响推荐的质量，有必要具体分析用户的评分习惯

1. 如果评分4.2是用户最喜欢的类别，则建议用户1看电影3。用户u对电影j的评级预测公式如下：



用户i用于预测影片的评级；s(u)表示u中的评级元素集，两个影片的平均偏差值为：



ui表示用户u对电影i的评级，R是整个评级数据j,i(R)标签包含一个对电影i和电影j的评级设置。

# SVD算法。

算法思想：采用基于奇异值分解模型的预测。算法原理：利用SVD算法将大稀疏用户评分矩阵映射到低维空间，计算未分级项目与低维空间其他项目的相似度，计算预测评级，将预测评级从高到低排序，返回前N项推荐用户。换句话说，重要的特征是由

压缩矩阵，然后进行计算和推荐。例如，电影故事片可以直观地分为惊悚片和喜剧片。

将用户评分矩阵M分解为三个乘积，U和V分别为左右奇异向量，Σ对角线的值为奇异值。



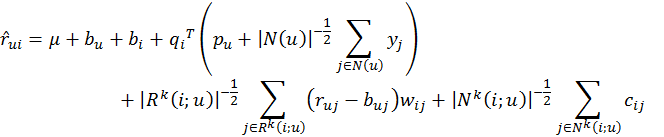
矩阵V对应于该用户，而矩阵U对应于该项。一般来说，只有观察奇异值最重要的特征，就可以近似地表示整个矩阵。常用的相似度计算方法包括欧氏距离、皮尔逊相关系数和余弦相似度。

# SVD++算法。

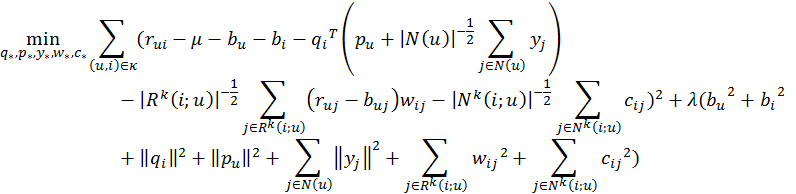
算法思想：SVD++算法是对SVD算法的一种改进，它也是一种基于奇异矩阵分解模型的算法。

算法原理：SVD++算法在SVD算法的基础上引入隐式反馈，并将用户的历史浏览数据、用户的历史评分数据等作为新的参数。隐式反馈与领域模型相结合。主要计算公式如下：

* 1. 评级预测：



* 1. 均方根误差：



在计算了用户对未知项目的评分后，用均方根误差来量化评分质量。添加正则化参数的意义是为了防止过拟合。这些符号如表2所示。

表2：这些符号的含义

|  |  |
| --- | --- |
| ***象征*** | ***含义*** |
| u | 用户 |
| i | 项目一（未知等级） |
| j | 与项目I相似的项目（已知等级） |
| k | 评级rui已知集 |
| rui | 用户u对项目i的真实评级 |
|  | 用户u预测了项目i的评级 |
| bu | 用户u评级偏差 |
| bi | 第一项额定值偏差 |
| qiT | 第一项中的因子向量 |
| pu | 用户u的因子向量 |
| N(u) | 用户u的itmes设置为隐式偏好 |
| k | 最接近项目i的项目数 |
| Sk(i) | 最接近项i的k个项的集合 |
| R(u) | 用户u的所有评级集 |
| wij | 从j到i的重量 |
| cij | 抵消了用户u对第j项的隐含偏好 |
| λ | 正则化参数 |

* 1. 算法应用结果：

三种算法的测试结果如表3所示。

表3：算法应用程序的结果

|  |  |
| --- | --- |
| ***预测结果***  ***算法*** | ***RMSE值*** |
| 一坡 | 0.8356574113243.2 |
| 斯维达 | 0.8134704664413.3 |
| ++路由器 | 0.816237015507 |

# 结论

由于在不同环境下运行的结果不同，本文在多台机器上进行了测试。初步结论：SVD++算法的推荐效果相对较准确。斜坡一种算法运行快速，但理论上相对简单，未能实现个性化推荐。SVD算法和SVD++是相对基本的算法，人们一直在不断改进。例如，timeSVD++算法是一种考虑了时间感知的算法。在推荐系统方面，人们收集智慧，整合各方面的知识，使推荐系统更加人性化和智能化。然而，我们应该继续探索。

# 参考文献

1. 德国，科尼格斯坦，韩国和韦默。机器学习研究杂志。卷。18（2012），p.3.
2. RMBell和YKoren。奈飞公司奖挑战的经验教训。AcmSigkdd探索时事通讯。卷。9（2007）No. 2，p.75.
3. 戈德堡D，尼科尔斯D，和OkiBM，等人。使用协作过滤来编织一个信息选项表。计算机协会的通讯，第一卷。第35号（1992年）第12号，第61页。
4. X.W.孟，S.D.。刘和y.j.。张志勇，等人。社会推荐系统的研究。软件杂志，卷。第26页（2015年）第6号，第1356-1372页。（中文版）。
5. 李德，宋。通过非负矩阵分解来学习对象的各个部分。《自然》杂志》，第一卷。401（1999）6755，p.788号。
6. 阿加瓦尔CC，WolfJL，WuKL，等。KDD-99提案：第五届ACMSIGKDD知识发现和数据挖掘国际会议。（1999年8月15年-18日，美国加州圣地亚哥）。p.201.
7. 野野、波莱加拉和石冢。国际AAAI会议项目(ICWSM2011)第五。（2011）。p.241.
8. 女主角和小权。利用基于排名的技术改进综合推荐的多样性。IEEE知识与数据工程学报。卷。24（2012）No. 5，p.896.
9. R.福尔萨蒂，马哈达维，沙姆斯法德和萨尔瓦特。《ACM的信息系统学报》，第一卷。第32期（2014年）第4号，第17页。
10. R.福尔萨蒂，巴贾斯特，埃斯法哈尼亚和第九届推荐系统会议记录（奥地利维也纳，2015年9月16-20日）。p.51.