**南 京 财 经 大 学**

研究生毕业论文开题报告

**题 目** 基于Spark的Slope One协同推荐算法

的改进与实现

**学生姓名：** 郝志远 **学 号：** 1120170709

**院 （系）：** 信息工程学院 **专 业：** 计算机应用技术

**指导教师：** 李树青

2019年4月23日

**研究生毕业论文开题报告**

|  |
| --- |
| **1. 选题依据（选题的背景与意义、国内外研究现状）**  **1.1选题背景与意义**  随着互联网和信息技术的迅速发展和广泛应用，越来越多的信息呈现在我们面前。Facebook注册用户达10亿，存储着约100亿张照片；淘宝的注册用户和在线商品数量都达到了几亿，每天活跃的数据量超过50TB。有统计机构表示，全球数据的使用量将在到2020年达到约44ZB（10万亿亿字节=1ZB)，将涉及经济社会发展的各个领域。大数据时代带来资讯极度丰富的同时也造成了信息超载，使得用户需要花费大量的时间去筛选自己真正需要的信息。传统的搜索引擎（百度，Google)只能通过用户的输入搜索相关信息，它为不同用户反馈的结果都是一致的，并不能根据用户的偏好提供个性化的服务。  为了解决上述问题，推荐系统应运而生，是当前解决信息超载问题的十分有潜力的方法。个性化推荐系统是一种过滤信息的重要方式，它可以在用户没有明确需求或者是信息量过于庞大、杂乱的情况下，根据用户的历史浏览行为信息，分析用户的兴趣爱好和需求，在海量的信息中找到用户感兴趣的内容，并推荐给用户。通俗地讲，推荐系统可以将合适的信息（人或者物品）推荐给合适的人，即它可以推荐更加符合人们口味和喜好的结果[1,2]。  近几年来，个性化推荐技术已经被成功的应用于电子商务、视频音乐网站、社交、个性化阅读等多个领域，成为人们日常生活中不可或缺的一部分，国内具有代表性的有淘宝、京东、豆瓣等。广泛的市场应用需求推动了推荐技术的快速发展，如今的推荐技术主要包括基于内容的推荐、基于关联规则的推荐和协同过滤推荐。其中协同过滤推荐是研究最广泛，应用最成功的一种推荐技术，协同过滤技术根据两个或者多个用户对于某些项目（如网页或者商品）给出的评价，找到用户之间或项目之间的相似性，再根据相似性大小找到最相似的即用户最为感兴趣的项目，推荐给用户。  在数据呈现出爆炸性增长的趋势背景下，面对海量的数据，传统的单一节点的推荐（所有的计算和数据存储都在单一的节点上完成）已经出现了存储能力不足、计算能力不足的弊端无法满足推荐算法的计算需求。为了应对这一挑战，新一代的推荐算法大多采用了并行化处理的方式，以此来缩短学习建模的时间，提高推荐算法的性能和运行效率。  Apache Spark是专为大规模数据处理而设计的快速通用的计算引擎，是由UC Berkeley AMP Lab所开源的类Hadoop MapReduce的通用并行框架[3]。Spark平台具有Hadoop MapReduce的所有优点[4]。但与Hadoop不同的是Spark作业任务的计算结果和中间输出可以保存在内存中，从而不必多次读写分布式文件系统，减少了大量的I0开销，因此其在计算过程中处理速度明显优于Hadoop[5,6]。  本文在各种类型的推荐系统被广泛提出和实践的背景下，对协同过滤推荐技术的优缺点进行了研究和分析，发现在实际过程中，单一的推荐算法不能很好的保证推荐质量，混合推荐算法可以进一步减少预测误差，提高推荐结果的精确性。同时，将改进算法在Spark平台下进行并行化设计与实现，提高推荐算法的运行效率，使算法能更有效的满足推荐作业的即时性要求，因此具有重要的应用价值。  **1.2国内外研究现状**  **1.2.1推荐算法的研究现状**  个性化推荐中推荐算法是最重要的部分，一定程度上决定了推荐的准确度和推荐系统性能的好坏。近年来学术界越来越关注推荐算法的研究，提出了预测用户兴趣的多种方法。  国外对推荐系统的研究起步很早，提出了预测用户兴趣的多种方法。其中基于用户的协同过滤算法是应用最早的算法，1992年就已经在Tapestry系统中得到了应用[7]，接着Group Lens 研究小组在1994年将其应用在了新闻推荐中。之后Sarwar B等人在2005年提出了基于项目的协同过滤算法[8]，并且在Netflix网站中得到了广泛的应用。这两种协同过滤算法思路简单且易于实现，但算法的数据稀疏性、冷启动导致算法对数据集的依赖程度很高。Daniel Lemire于2005年提出了一种简单实用的Slope One协同过滤算法，该算法具有较小的空间以及时间复杂度，但也存在数据稀疏性、冷启动的问题。面对协同过滤算法存在的冷启动问题，研究者们提出了一种基于内容的推荐算法，其主要思想是从可推荐项目集合中选择那些与用户曾经喜欢的项目在内容上相似的项目[9]。由于不依赖评分记录，所以该算法不存在数据稀疏性、冷启动的问题，但因多媒体对象特征提取比较困难，无法将该算法应用在视频、音频等对象上。随着推荐算法的发展，有许多成熟的机器学习方法应用到了推荐领域。基于关联规则的推荐算法通过发掘不同项目之间的关联性形成一定的关联规则[10]，一个著名的案例就是沃尔玛发现的啤酒与尿不湿的关联性。该算法简单直观，不需要建立用户的偏好模型，但由于难以制定合理的推荐规则，使用在大型推荐系统中会使系统变得笨重。文献[11]提出了一种可以对评分记录不完整的矩阵进行分解的矩阵分解方法，不需要对评分矩阵估值填充，通过矩阵分解进行降维，提取出隐式特征，一定程度上解决了数据稀疏造成的计算误差，有很好的推荐精度。在Netflix推荐系统竞赛中的应用表明，该矩阵分解能产生准确的推荐[12]。文献[13]将聚类技术引入到了推荐算法中，通过对相似用户的聚类，减少最近邻用户的搜索范围来提高推荐系统的响应速度。随着微信等社交网络的普及，社会化推荐逐渐成为了推荐领域的研究重点[14]。该方法主要通过分析用户之间的共同兴趣形成不同的社会化关系网络，当网络中用户对某一项目感兴趣时，系统会将其感兴趣的项目推荐给同一社会化网络中的其他用户[15]。  在国内，黄义纯提出一种改进的Slope One算法，该算法同时考虑了用户共同评分个数以及项目之间相似度，并以两者的乘积为权重。其中项目之间相似度分别采用余弦相似度、修正余弦相似度和皮尔逊相似度进行求解[15]。刘毓针对协同过滤算法的推荐精度不足问题，以基于用户协同过滤算法为前提，使用皮尔逊相似性计算用户间相似度，利用Top-N方法对相似用户进行筛选，把最相似用户作为邻居集，再结合加权Slope One算法，预测项目评分[16]。李桃迎等人提出了基于用户兴趣遗忘函数和用户最近邻居筛选策略的改进方案[17]。王冉等人提出了一种结合三部图综合扩散的Slope One算法，由于三部图综合扩散算法同时经过用户节点和项目类别节点进行扩散，并且计算得到的项目之间的推荐度是非对称的，更能体现用户兴趣[18]。文献[19] 利用了用户相似性，项目相似性和项目属性的优势，对原加权Slope On算法进行了改进。覃幸新提出一种基于改进的项目相似性度量、改进的项目属性相似性度量和用户评分概率函数的多权值的Slope One协同过滤算法, 在项目相似性度量方面将共同评价的两个项目的用户数量和Pearson 相关系数相融合, 在项目属性相似性度量方面将修正的拉普拉斯平滑与 Jaccard 系数相结合, 同时利用用户评分概率函数对用户已有评分进行有效区分[20]。张圣筛通过改进项目属性相似性计算方法，并将项目自身属性与用户项目评分相结合计算新的项目相似度[21]。 王万良等人引入巴氏系数作为项目之间的相似性度量方法，分别在用户和项目两个维度上进行了改进，并且在用户维度采用聚类方法消除用户行为习惯差异[22]。 冯勇等人从项目自身的所属类型和项目的评分数据两方面来综合计算项目间的相似度关系，并在此基础上增加了项目筛选策略，达到了稳定评分差、获取局部较为密集的项目评分矩阵的目的[23]。  **1.2.2分布式计算的研究现状**  Hadoop是目前应用广泛的开源分布式计算框架之一，当前Hadoop主要包括一套名为HDFS的文件系统以及MapReduce计算框架。它将计算作业任务分发到集群的各个节点，不同的节点实现并行的计算，提高了算发的执行效率；并且它能够通过扩展集群节点的方式增加计算平台的的存储能力和计算能力，因此Hadoop是可伸缩的。正是由于这种特性，Hadoop已经被广泛的应用于大数据处理领域，部分研究人员也在Hadoop平台上实现了推荐算法的并化。如Zhao Zhidan等人在Hadoop平台上实现了基于用户相似度的协同过滤[24]，李改等人将矩阵分解推荐算法在Hadoop平台上实现了并行化[25]，郑键在其论文中实现了基于内存模型的协同过滤的并行化[26]。  传统的基于Hadoop的MapReduce只提供了Map和Reduce两个操作，并且处理逻辑隐藏在代码细节中，没有体现出整体逻辑，造成表达力欠缺使用难以上手的问题。在需要多次迭代的计算中由于/O访问过多影响了计算任务的执行效率。而Spark平台依托内存计算和迭代计算刚好弥补了这些需求，得到了开源社区以及广大企业的青睐，成为了大数据处理方向的研究重点。国内对于Spark的研究目前主要集中在一些互联网行业，例如，腾讯公司TDW（Tencent Distributed Data Warehouuse)数据仓库已经在预测用户的广告点击率、计算两个好友之间的共同好友数等业务案例上使用Spark替代原来的Hadoop MapReduce，并取得了很好的性能提升[27]。淘宝最初使用Mahout结合自己开发的MapReduce来解决阿里搜索和广告业务中的复杂机器学习，导致学习效率低，代码维护困难，后来采用了Spark解决需要多次迭代、计算复杂度高的算法，取得了很好的效果。优酷土豆在使用Hadoop集群的过程中发现在进行模拟广告投放及图计算等大数据量计算时迭代运算过程占用了大部分资源而且处理的速度很慢。后来采用Spark提高了模拟广告投放的计算效率高，减少了网络传输延迟（同hadoop比延迟至少降低一个数量级），目前优酷土豆在视频推荐、广告业务中已经广泛使用了Spark[28]。梁化强在Slope 0ne算法中融入相似性计算、活跃用户筛选和用户聚类等技术，通过在MovieLens数据集上进行试验验证，并比较算法在Spark和Hadoop平台并行化的运行效率，证实了该算法可以有效降低MAE[29]，郭霖以用户为前提的协同过滤和以情境为前提的后过滤模型给出协同过滤算法处理步骤，并在Spark平台运行[30]。李星利用Spark平台设计了一个基于物品的协同过滤算法的商品推荐系统，并将其应用在 MovieLens数据集上运行测试[3l]。 |
| **2．研究目标与内容**  **2.1研究目标**  本文主要研究Slope One协同过滤算法的改进以及改进算法在Spark平台的并行化设计与实现。首先，针对MovieLens数据集的数据稀疏性问题，通过原始的Slope One算法预测部分用户对物品的评分，通过填充评分矩阵解决数据稀疏性造成的冷启动问题；分析原始Slope One推荐算法存在的问题，对于其仅考虑共同评分用户之间的差值，没有考虑用户与项目之间的相关性以及项目之间相似性的缺点，引入用户项目之间的相关性以及项目之间的相似度改进Slope One算法，通过控制用户、项目相似性优化算法效果；最后，研究Spark计算框架的工作原理和运行架构。在此基础上，实现改进的Slope One算法在Spark平台的并行化运行，提高算法的运行效率和可扩展性，并通过实验对文中改进的算法进行预测准确性和并行性能评测。  **2.2研究内容**   1. 应用Slope One算法解决冷启动问题   Slope One 算法是基于不同项目之间的评分差的线性算法，是一种经典的基于用户-项目评分矩阵的推荐算法，该算法是一个增量算法，对评分较少的用户也可以产生推荐，同时准确度比传统的基于用户和项目的协同过滤算法要好，故本文采用 Slope One 算法来计算预测评分值。与其他类似推荐算法相比，它的最大优点在于算法很简单，易于实现，执行效率高，同时推荐的准确性相对较高。  数据稀疏性一直以来都是协同过滤算法的一大难题，目前已有的稀疏矩阵填充方法虽然可以使推荐系统的质量得到提升，但仍然难以完美解决协同过滤算法中存在的所有问题，比如空值填补法中若采用缺省值（即用户评分中值、均值、默认值 0）来回填数据，由于用户的未评分项不可能完全相同，导致信任度不高。本文就近邻评分数据过少，先通过原始数据得到初步的用户相似度和每个用户的近邻，利用 slope one 算法计算评分预测值来填充数据，并基于填充后的数据修正相似度和优化近邻选取集合，最终给出目标用户的推荐列表。   1. 改进Slope One算法   Slope One算法是一种简单实用的协同过滤算法。它基于这样一个假设：用户对两个项目之间的偏好存在着某种线性关系，可以通过简单的线性拟合表示，即可以通过项目*X*预测用户对项目*Y*的偏好值。Slope One对线性函数做了进一步的简化，假设*m*=l， 则项目之间的关系可以表示为  从公式可以看出，不同于传统的基于用户和基于项目的协同过滤算法，该算法并不需要计算用户或者项目之间的相似度，只需要确定*b*，即两个项目之间的平均差异，就能进行预测推荐。  通过上述的算法即可算出用户对项目的预测评分值，但是我们发现，上述算法中没有考虑对同一项目有过评分的用户数量对预测结果的影响。假如共同评价过项目*i 、j*的用户数量为100，而共同评价过项目*i 、k*的用户数量为1000，显然项目*k*和项目*i* 之间的评分偏差对预测结果的影响要大。因此Daniel Lemire等人提出了一种改进加权的Slope One算法(Weighted Slope One，简称WSO)，该算法使用物品间共同评分的用户数量作为评分偏差的权重，具体计算公式如下：  可以看到，Slope One算法在预测用户评分的过程中没有考虑用户与项目之间的相关性以及项目之间相似性。首先，Slope One在预测评分的时候仅仅考虑了用户评价过的项目之间的评分偏差，这使得该算法在通过比较用户相似性而进行个性化推荐的时候存在缺陷。假设己知用户A对项目*i，j*的实际评分需要预测用户A对项目*k*的评分。有50位用户同时对项目*i，k*有过评分，并且用户A与这50位用户之间的平均相似度为0.2，有50位用户同时对项目*j，k*做了评分，并且用户A与这50位用户之间的平均相似度为0.8。显然，用户A对项目*j*的评分以及项目*j，k*之间的评分偏差在最终的预测评分中应该占更大的比重，但是Slope One算法没有做类似的判断。所以在最终的预测中应该考虑用户之间的相似性。其次，Slope One在预测评分时没有考虑项目之间的相似性。虽然WSO算法将共同评价过两个项目的用户数目作为预测过程中两个项目之间评分偏差的权重，但是也没有考虑项目之间的相似性。例如，给出了用户A对项目*i，j*的评分需要预测用户A对项目*k*的评分，有100位用户同时对项目*i，k*做了评分，并且项目*i*与项目*k*之间的相似度为0.2，有50位用户同时对项目*j，k*用过评分，并且项目*j*与项目*k*之间的相似度为0.8。显然，共同评价两个项目的用户数并不能完全描述项目之间的相似性，应该加入项目之间的相似度作为最终预测中项目之间评分偏差值的权重。   1. Spark计算框架的设计和应用   在Hadoop平台中，每个计算任务都需要写一个MapReduce程序，实现一个算法就需要编写多个MapReduce实现并行处理。而在Spark平台中并没有严格的Map和Reduce划分，并行化主要是通过能对其进行分布式并行操作的分布式数据集RDD实现的。在需要并行化的时候，采用RDD的编程模型，将数据按照指定的分片个数封装在RDD中不同的分区，然后对RDD进行map、filter、union等并行数据集的互相转化操作，RDD中不同的数据分区便可实现数据的并行处理。  RDD数据格式的设计与转换是并行化过程中的重点与难点。 |
| 1. **拟解决的关键问题难点** 2. Slope One算法填充稀疏矩阵的邻居选择问题   面对海量的数据，Slope One会遍历整个评分矩阵，计算目标用户和其他用户共同评价过的所有项目之间的平均偏差，这极大的影响了计算效率。传统的协同过滤算法中查找近邻用户的方法通常是先计算用户之间的相似度，再以相似度为基础，采取一定的策略选出目标用户的近邻用户。采取的策略一般有两种，一种是固定选取k个全局和目标用户最相似的用户，另一种是确定一个相似度阀值，并选择所有相似度超过该阀值的用户。固定k个用户的方法由于数据的稀疏性，会导致一些偏好相似性较低的用户参与到预测评分中。因此，设定阈值的方式更能充分利用现有的“用户-项目”评分矩阵所提供的数据。   1. Slope One引入用户项目相关性及项目之间相似度的改进方法   首先，Slope One在预测评分的时候仅仅考虑了用户评价过的项目之间的评分偏差，这使得该算法在通过比较用户相似性而进行个性化推荐的时候存在缺陷；其次，Slope One在预测评分时没有考虑项目之间的相似性。如何在最终预测中结合用户项目的相关性及项目之间相似度进行预测是是本课题重点考虑的问题。   1. Spark RDD流程设计   RDD支持两种操作，转换(Transformation)和动作(Action)，转换就是从现有的RDD创建一个新的RDD，比如map、union、filter等函数都是通过处理RDD中的每一个元素并返回一个新的分布式数据集。所有的转换操作都是惰性的，转换只是对应用到基础数据集上的转换动作做了记录，并没有计算出结果，只有当一个需要返回计算结果给驱动程序的动作被触发的时候，这些转换操作才会被真正的执行。动作是将RDD数据集上运行的计算结果返回Driver或者写到HDFS等外部存储系统中的操作。例如collect()就是一种动作，它将RDD中的所有元素以数组的形式返回给驱动程序。将 MovieLens数据集的csv文件从HDFS文件系统引入，通过RDD操作转换成RDD文件。 |
| **四．研究方法与技术路径**   1. 动态阈值查找用户最近邻居   为了设定合理的阈值，本文将采用动态阈值的策略查找用户最近邻居。具体步骤如下：  第一步，首先计算出目标用户与其他用户之间的相似度，将其他用户按相似度从高到低进行排序，选出前*m*个用户作为近邻集*S1*；  第二步，计算近邻集*S1*中所有用户与目标用户的平均相似度，将该平均相似度作为相似度的阈值*β*；  第三步，以第二步计算出的相似度阈值β为基础，选择*S1*中所有对目标项目有评价的且相似度大于该阈值的用户添加到近邻集*S2*中；  第四步，计算近邻集*S2*的用户数量。如果*S2*的数量小于*n*，则从近邻集*S1*中选择相似度最高的且未添加到*S2*中的用户添加至*S2*中；如果*S2*的数量大于*n*，则近邻集*S2*即为最终需要的用户最近邻居。  该方法中，由于不同目标用户的近邻用户群的相似度不同，因此每个目标用户都有自己的阈值，这种相似度阈值的设定方式相较于传统阈值的设定方式灵活性更强。同时，通过设定阈值*β*，可以保证参与计算目标用户预测评分的近邻用户有一个较高的相似度值，使得目标用户和近邻用户之间的相关度更大，推荐结果的准确率更高。  采用这种近邻选择的策略，可以保证每次在进行评分预测的过程中，均会有大于*n*个近邻用户参与其中，避免出现数据稀疏导致的结果不准确。   1. Slope One算法的改进设计   本文针对该算法存在的不足，引入用户与项目相关性以及项目间相似度作为预测权重，提出了一种综合考虑项目之间相似性以及用户相似性的Slope One改进算法。算法过程如下：  (1)数据预处理。数据集的主要来源为文件或者数据库系统，在进行计算之前需要将数据的格式转换为符合算法需要的格式，这里需要的是一个m行n列的二维矩阵，用R表示。(done)  (2)计算用户之间的相似度。在改进算法中采用了Pearson（余弦相似度，在不存在相似关系时才使用皮尔逊，皮尔逊一般也可以用距离代替）相似度来计算用户之间的相似度，具体公式如下：  其中为用户*u*和用户*v*共同评价过的项目集合，表示用户*u*对项目*i*的评分，表示用户*u*对所有项目评分的平均值。(done)  (3) 代表用户*u*与所有对项目*i*有过评分的用户之间的平均相似度，我们称其为用户项目相关性，具体的计算如下：（done）  建议操作：极大似然估计，直接改为  其中，表示用户*u*与用户*t*之间的相似度，表示共同评价过项目*i*的用户的集合，表示共同评价过项目*i*的用户数目，我们定义一个*m*行*n*列的用户项目相关性矩阵*S*来记录所有的。  (4)计算项目间评分偏差矩阵。改进算法计算评分差异值与Slope One算法所用的方法相似，公式如下：（done）  表示用户*v*对项目*i*的评分，表示对项目*i*和项目*j*都有评分的用户集合，表示集合中包含的用户数量，即共同评价项目*i*和项目*j*的用户数目。  (5)计算项目之间的相似度，公式如下：(done)  (6)预测评分。这是整个算法的核心，需要解决的是如何将项目相似度和用户与项目相关性与Slope One算法相结合。在前面的步骤中，已经完成了、、的计算，接下来根据这些结果来预测用户*u*对项目*j*的评分，具体的计算公式如下：(done)  其中，表示用户*u*评价过的所有项目的集合。  通过上面的计算步骤，就可以对用户*u*没有评价过的项目进行预测评分，然后对测出的评分进行排序，将前面*N*个项目作为推荐列表展示给用户。   1. Spark RDD的编程设计   关键步骤的RDD设计如下：  计算项目评分相似度：本文采用的是Pearson方法计算项目之间的相似度，在计算过程中需要预先计算每个项目获得评分的平均值，在flatMap中通过迭代*List(( Uid , Rat)，…)*计算项目获得的平均评分，List中元组的数对项目有评分的用户数。为了获得共同评价过两个项目的用户评分集合，需要先收集用户评价过的项目列表，然后将项目两两配对。这里为了减少计算量，首先对用户评价过的项目列表按照*lid*升序进行排列，这样在迭代时就可以按照顺序进行配对。例如对于项目配对(10，11)、(11，10)只计算了项目对(10，11)的相似度，没有计算项目对(1l，10)的相似度，减少了一半的计算量。  计算用户与项目相关性矩阵：要计算用户与评价过某项目i的其他用户之间的平均相似度*Sui*，必须知道评价过某项目*i*的用户列表。在flatMap的转换中，将元素键值对中的value部分*(*对项目*i*做过评价的用户列表*List((Uid , Rat)，…))*中的用户两两配对。以每个用户与项目*i*的组合作为key，用户配对作为value，形成元素格式为*<(Uid1，Iid)，(Uid1，Uid2)>*的RDD1。对RDD1进行map操作，在map中使用RDD1中元素的value值*(Uid1，Uid2)*获得广播变量userSim中用户之间的相似度。同时形成元素格式为*<(Uid1，Iid)，simU>*的RDD2。其中，simU为用户*Uid1*与某一个对项目lid有过评分的用户之间的相似度。在combineBykey中通过自定义的creCombSu、mergeSuValue以及mergeSuComb函数对RDD2中具有相同key的元素做聚合计算，形成以元组*(Uid1，Iid)*为键，二元组*(s，n)*为值的键值对。最后通过*mapValues*求解用户项目之间的相关性，*userltemSlm*的元素的格式为*<( Uid1，Iid)，sui>*，其中*sui*代表用户*Uid1*与评价过项目*lid*的其他用户的平均相似度也即用户*Uid1*与项目*Iid*之间的相关性。*sui*由RDD3中每个元素的n除S计算而来。 |
| **五．可行性分析**   1. 理论分析可行性   Slope One协同过滤算法是一种简单高效且易于实现的协同过滤机制，该算法主要依赖同一用户对不同项目评分的差异产生推荐，并且这些差异值可以离线计算，从而减少线上的计算量提升运行效率。相对于其他协同过滤算法，Slope One基于较少的历史数据就可以进行数据分析，向用户提供服务，有着易于实现且预测准确度高的优势，但是它依然有其自身的缺陷。所以本文提出的同时考虑用户项目相关性和项目之间相似度的优化方法可以对原始Slope One算法进行改进。   1. 技术线路可行性   Spark生态系统以RDD运算为基础、以Spark计算框架为核心，建立了一个基于内存计算的大数据平台。其丰富的生态系统为开发提供了极大的便利，MLlib组件提供了一些常见的机器学习算法和实用程序，包括分类，回归，聚类，降维，以及底层优化，同时也提供了相关测试和数据生成器。对于推荐系统领域的诸多算法也提供了方便的接口。此外，Spark拥有多种运行模式，既能以本地或者伪分布式的模式运行在单机上，也能以分布式的方式运行在集群上。当以分布式的方式运行的时候，底层的调度系统可以使用Yam、Mesos，或者Spark自己的资源调度Standalone。  通过RDD的编程设计，将改进的Slope One算法在Spark平台运行，提高了效率和可扩展性。   1. 研究能力可行性   自入读研究生以来，已阅读大量文献，对于基于用户、项目以及slope one协同过滤算法有了比较全面的认识；已经研究了Spark并行化平台的运行原理与使用方法。 |
| **六．特色与创新之处**   1. 引入Slope One算法解决用户稀疏性问题   与其他类似推荐算法相比，Slope One算法的最大优点在于算法简单，易于实现，执行效率高，同时推荐的准确性相对较高。用Slope One算法预测部分用户对物品的评分，进而填充评分矩阵。通过填充评分矩阵，解决数据稀疏性造成的冷启动问题。   1. 提出一种考虑用户项目相关性及项目相似度的slope one改进算法   本文在Slope One算法没有考虑用户之间相似性及项目之间相似性的基础上，基于共同评价的用户相似度，计算用户与特定项目之间的相关性，它代表用户对多有项目进行评分的用户之间的平均相似度，再结合项目之间的相似度，改进Slope One算法。  3. 基于Spark平台实现  不同于大部分推荐系统实验使用的Hadoop技术，本文采取的Spark平台更高效，推荐效果更好。 |
| **七．研究计划与进度安排**  2017.9－2017.12：基础理论学习、文献检索；  2018.1－2018.6：研究推荐系统原理，重点研究协同过滤算法；  2018.6－2019.3：研究基于Hadoop的分布式计算平台，对协同过滤算法进行实现；  2019.3－2019.6：在Hadoop基础上研究Spark计算框架的原理与使用，针对Slope One协同过滤算法未考虑用户项目相似性及项目间相似性的问题提出改进方法；  2019.6－2019.12：处理数据，实验对比；  2019.12－2020.3：撰写论文，准备答辩。 |
| **八．参考文献**  [1] Tariq Mahmood, Francesco Ricci. Improving recommender systems with adaptive conversational stragegies [J]. HT 09 20TH ACM Conference on Hypertext and Hypermedia, Toronto, 2009, ACM Press.  [2] Pooja Vashish, Punam Bedi. Interest-Based Personalized Recommender System[J].World Congress on Information and Communication Technologies of IEEE.2011,11:245-246.  [3] Apache. Apache Spark [EB/OL]. [2019]. http://spark.apache.org.  [4] Apache. Apache Hadoop [EB/OL]. [2019]. http://hadoop.apache.org.  [5] Gu L, Li H. Memory or Time: Performance Evaluation forlterative Operation on Hadoop and Spark: High Performance Computing and Communications &amp; 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC\_EUC),2013 IEEE 10th International Conference on,2013[C].  [6] Patel J A, Sharma P. Big data for better health planning: 2014 International Conference on Advances in Engineering and Technology Research (ICAETR),2014[C].  [7] Goldberg D. Using collaborative filtering to weave an information tapestry: Communications of the Acm,1992[C].  [8] Sarwar B, Karypis G, Konstan J, et al. Item-based Collaborative Filtering Recommendation Algorithms,PROC.10TH INTERNATIONAL CONFERENCE ON THE WORLD WIDE WEB,2001[C].  [9] Pazzani M J, Billsus D. Content-Based Recommendation Systems[M]. Springer-Verlag,2007.  [10] Lin W, Alvarez S A, Ruiz C. Collaborative Recommendation via Adaptive Association Rule Mining: Data Mining and Knowledge Discovery,2010[C].  [11] Paterek A, Paterek A. Improving regularized singular value decomposition for collaborative filtering[J]. Proceedings of Kdd Cup & Workshop,2007.  [12] Netflix. Netflix prize [EB/OL]. [2019]. http://www.netflixprize.com.  [13] Gong S J, Ye H W. Joining User Clustering and Item Based Collaborative Filtering in Personalized Recommendation Services: Industrial and Information Systems,2009.IIS 09.International Conference on,2009[C].  [14] King I, Lyu M R, Ma H. Introduction to social recommendation.: World Wide Web Conference Series,2010[C].  [15]黄义纯. Slope One推荐算法改进[J]. 现代计算机(专业版), 2017(35):24-27.  [16]刘毓, 何锐. 一种改进Slope One的学习推荐算法[J]. 西安邮电大学学报, 2017(4).  [17]李桃迎, 李墨, 李鹏辉. 基于加权Slope One的协同过滤个性化推荐算法[J]. 计算机应用研究, 2017(8).  [18]王冉, 徐怡, 何明慧. 结合三部图综合扩散的Slope One推荐算法[J]. 微电子学与计算机, 2018.  [19]韩影. 基于项目属性相似的加权Slope One算法[J]. 科学技术创新, 2018(29):12-13.  [20]覃幸新, 王荣波, 黄孝喜. 基于多权值的Slope One协同过滤算法[J]. 数据分析与知识发现, 2017, 1(6).  [21]张圣筛, 陈家琪. 一种改进相似度计算的Slope One推荐算法[J]. 电子科技, 2017(10).  [22]王万良, 屠海龙, 朱炎亮. 融合巴氏系数的用户聚类Slope One算法[J]. 小型微型计算机系统, 2018.  [23]冯勇, 徐红艳, 王嵘冰. 融入项目相关性的加权Slope One算法研究[J]. 计算机科学与探索, 2018, 12(10):165-174.  [24] Zhao Z D, Shang M. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop: Knowledge Discovery and Data Mining,2010.WKDD '10.Third International Conference on，2010[C].  [25]李改，李磊. 基于矩阵分解的协同过滤算法[J].计算机工程与应用，2011,47(30):4-7.  [26]郑健. 基于Hadoop的协同过滤推荐系统研究与实现[D].北京邮电大学，2014.  [27]中国IDC圈. 腾讯在Spark上的应用与实践优化[EB/OL].[2019].http://cloud.idcquan.com/yjs/73681.shtml.  [28]钱魏. Spark在腾讯数据仓库TDW的应用[EB/OL].[2019].http://www.biaodianfu.com/spark-tdw.html?utm\_source=tuicool&utm\_medium=referral.  [29]梁化强, 唐坚刚. 基于Spark的Slope One算法优化与实现[J]. 软件导刊, 2018, v.17；No.188(06):96-98+103.  [30]郭霖. 基于Spark的推荐系统的研究[J]. 电信快报, 2018, No.563(05):43-44+47.  [31]李星, 李涛. 基于Spark的推荐系统的设计与实现[J]. 计算机技术与发展, 2018, 28(10):201-205. |
| **九．指导教师意见（对深度、广度及工作量等方面的意见）**  该课题拟将优化的推荐算法应用于并行计算框架，以提高推荐效果，选题具有较大的价值。报告提出的研究目标明确，研究内容具有一定的深度和广度，且工作量饱满，研究计划安排合理。报告提出的研究的关键问题准确，提出的技术方案具有可行性。同意该生按本开题报告开展研究工作。  指导教师：  年 月 日 |