文章编号：2096-1472(2021)-09-13-04
DOI:10.19644/j.cnki.issn2096-1472.2021.09.003

大数据平台下实时电影推荐算法研究

牛路帅，彭 龑

(四川轻化工大学自动化与信息工程学院，四川 宜宾 644000)
✉644056822@qq.com; 2634932795@qq.com

摘 要：随着互联网数据量的不断扩大，数据的实时推荐需求已经不能被传统的推荐模型所满足，协同过滤推荐算法的不足也越来越明显。为此，通过大数据计算框架Spark平台构建基于模型的推荐算法来更好地应对海量数据实时推荐的问题。首先，通过预先设定的计算方法进行模型的构建；同时将一种改进的余弦相似度算法应用到模型中，不仅可以缩短推荐实现的时间，而且可以提高推荐性能。实验结果表明，该算法和传统协同过滤算法相比，提高了准确率和时效性，验证了系统可较好地满足用户的实时需求。

关键词：Spark；实时；推荐算法；协同过滤
中图分类号：TP399 **文献标识码**：A

Research on Real-time Movie Recommendation Algorithm based on Big Data Platform

NIU Lushuai, PENG Yan

(*School of Automation and Information Engineering, Sichuan Light Chemical Technology University*, *Yibin* 644000, *China*)
✉644056822@qq.com; 2634932795@qq.com

Abstract: With the continuous expansion of the amount of Internet data, traditional recommendation model can no longer meet the demand for real-time recommendation. The deficiency of collaborative filtering recommendation algorithm is becoming more and more obvious. For this reason, this paper proposes to build a model-based recommendation algorithm based on the Spark platform of big data computing framework, in order to better deal with the problem of real-time recommendation of massive data. First of all, the model is constructed through the preset calculation method, and an improved cosine similarity algorithm is applied to the model, which can not only shorten the time of recommendation implementation, but also improve the performance of recommendation. Experimental results show that compared with the traditional collaborative filtering algorithm, the proposed algorithm improves the accuracy and timeliness, and verifies that the system can better meet the real-time needs of users.

Keywords: Spark; real time; recommendation algorithm; collaborative filtering

1 引言(Introduction)

在信息爆炸的今天，信息超载的问题[1]逐渐显现出来，人们有许多种方式和途径去获取信息，但是很难从这些信息中找到自己所感兴趣的东西。推荐系统则是解决这个问题的关键所在[2-4]。推荐系统能够通过分析用户的兴趣和行为，智能地向用户推荐所需信息，其既需要筛选出大量有用的数据，又要实时地满足用户的个性化需求[5]，所以大数据处理是推荐系统所要具有的能力[6-7]。Spark是一种具有大数据处理能力的内存计算框架[8]，运行于Spark平台的推荐系统可以发挥更高的处理能力。本文旨在利用Spark平台对基于模型的推荐算法进行优化与并行化实现，和传统的推荐算法相比，实时性与准确性也都有较大的提高。

基金项目：自贡市科技局科技计划(2018GYCX33).

2 Spark分布式计算框架(Spark distributed computing framework)

2.1 Spark运行架构原理

Apache Spark是一种基于内存的快速、通用、可扩展的大数据计算引擎[9]。当执行一个用户编写的Spark应用程序时(SparkContext)，Driver程序会创建SparkContext向集群申请资源，并在此阶段中生成DAGScheduler和TaskScheduler两个模块，资源管理器启动Executor进程。DAGScheduler将任务解析成Stage，之后把一个个TaskSet提供给底层调度器TaskScheduler处理[10]。

Executor向应用程序申请Task，TaskScheduler将Task发放给Executor运行并提供应用程序代码。具体流程如图1所示。

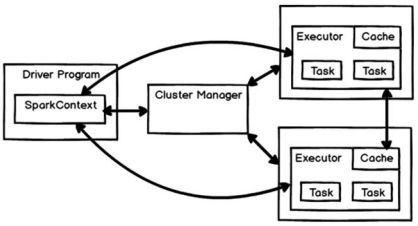

图1 Spark工作流程

Fig.1 Spark workflow

2.2 Spark平台部署

本文搭建的Spark集群[11]采用3 台普通的PC，其中包括1 个Master节点和2 个Slave节点。3 个节点均采用CentOS 6 Linux系统，节点之间采用局域网连接，具体情况如表1所示。

表1 节点具体说明

Tab.1 Node specification

主机名	IP 地址	作用
Linux	192.168.1.102	NameNode，Master
Linux1	192.168.1.103	DataNode，Worker
Linux2	192.168.1.104	DataNode，Worker

Java JDK安装包采用jdk-8u144-linux-x64.tar.gz，Scala安装包采用scala-sdk-2.11.8.tar.gz，Hadoop安装包采用hadoop-2.7.2.tar.gz。安装过程中配置安全外壳协议(SSH)以便实现远程无密码登录与管理。完成Hadoop集群的配置安装后进行Spark的安装，Spark安装包采用Spark-2.1.1-bin-hadoop2.7.tgz，软件开发环境采用IntelliJIDEA 2019.2.4。

3 推荐系统与协同过滤推荐算法(Recommendation system and collaborative filtering recommendation algorithm)

推荐系统需要大量的数据作为支撑，和搜索引擎[12]一样，它需要挖掘和解析用户历史的行为数据，尽可能地知晓用户的个性化需求，进而主动为用户提供所需信息，给用户的兴趣建模[13-14]。

协同过滤推荐算法[15-17]是根据用户行为数据设计的较早的著名推荐算法。协同过滤推荐算法可以分为基于用户的协同过滤算法[18]和基于物品的协同过滤算法[19]两类。基于用户的协同过滤算法的主要思路是首先确立目标用户并找出与他相似兴趣的用户，在不考虑他们皆感兴趣部分的前提下，将相似用户感兴趣的内容推荐给目标用户，适用于用户较少的场合，比如新闻推荐。基于物品的协同过滤算法主要是通过分析用户的历史行为记录找出与之相似的物品，将这一类物品推荐给用户，适用于物品数远小于用户数的场合，比如图书、电子商务。

4 基于模型的实时推荐算法(Model-based real-time recommendation algorithm)

实时推荐结果反映了用户最近一段时间的偏好[20-21]，用协同过滤推荐算法已经不足以满足实时的需求，本文提出了基于模型的实时推荐算法，其主要思路是根据预先设定的模型来完成推荐。算法设计如下：

当电影q被用户u评分时，将对u的推荐结果进行实时更新。此时，推荐结果会产生改变，把和电影q极为近似的K个电影挑选出来，将它们标记为备选电影。每个备选电影被推荐的先后顺序取决于“推荐优先级”的权重，它们会通过用户u最近时间内的评分，分别计算出各自的推荐优先级，根据计算出来的结果同上一次的推荐结果进行归并和更换，得出最终的实时推荐结果。详细过程如下：

按照时间先后顺序，得到用户u最近的K个评分，表示为RK；对于电影q，将S记为与其最为近似的K个电影的集合；之后计算每个电影$q \in S$的推荐优先级，公式如式(1)所示。

$$E_{uq}=\frac{\sum_{reRK}\mathrm{sim}(q,r)\times R_r}{\mathrm{sim\_sum}}+\lg\max\{incount,1\}-\lg\max\{recount,1\} \quad (1)$$

电影q和电影r之间的相似度用$sim(q,r)$来表示，为了计算简便，将最小相似度的阈值设置为0.7。两个电影之间相似度如果低于0.7，则表示它们没有关联，不予考虑；$incount$代表的是在RK中，不但与电影q相似，而且自身评分大于等于3的电影数量；RK中评分低于3，与电影q相似的电影数量则用$recount$来表示；sim_sum是在RK中符合最小阈值条件的电影数量；其中在计算$sim(q,r)$过程中用到了一种改进的余弦相似度公式，计算公式如式(2)所示。

$$W_{qr}=\frac{\sum_{u\in N(q)\cap N(r)}\frac{1}{\log 1+|N(u)|}}{\sqrt{|N(q)||N(r)|}} \quad (2)$$

其中，W_{qr}表示电影的两两相似度，u代表用户。由于存在热门电影，很多用户都很喜欢，这样计算的相似度就会很大，因

此会造成任何电影都会和热门电影有很大的相似度。还有一种情况是，用户看很多电影并不都是出于自身的兴趣，虽然活跃，但是有可能是职业需要，而且这些电影覆盖了很多领域，所以这个用户对于他所看过的电影的两两相似度的贡献应该远远小于一个只看了几部自己喜欢的电影但是不活跃的用户。公式针对热门电影和活跃用户进行了一定的惩罚。

在每个备选电影q中，从用户u的最近K个评分之中，发现和电影q相似度大于等于0.7的电影，将发现的每个电影r与备选电影q之间的相似度乘上r自身的评分，然后进行加权求和，得出用户u对电影q评分的预测值。在用户u评分的K个电影中，将与电影q相似并且评分值大于等于3的电影数量标记为$incount$，而评分值小于3的电影数量标记为$recount$。$\lg\max\{incount,1\}$用作电影q的“强化因子”，其作用是表示在用户u评分过的K个电影当中，评分值大于等于3且与电影q相似的个数，那么需要根据相似的个数相应地去增加电影q的推荐优先级，如果其中相似的高评分电影越多，那么电影q被推荐的可能性越大；反之其被推荐的可能性比较小。$\lg\max\{recount,1\}$用作电影q的“弱化因子”，作用是表示在用户u评分过的K个电影当中，评分值小于3且与电影q相似的个数，则需要根据近似的个数相应地去减少电影q的推荐优先级，如果其中相似的低评分电影有很多，那么不推荐电影q的可能性较大。

最终通过公式(1)计算出用户u的每个备选电影q的推荐优先级，并且形成一个电影<ID，推荐优先级>的推荐列表。因为在本次推荐之前已经有了一个旧的推荐列表，所以将两个推荐列表按照推荐优先级的大小进行归并与替换，得到最终的推荐列表。

在Spark上实现算法时，首先计算电影相似度矩阵，会为用户、电影分别生成最终的特征矩阵。用户特征矩阵为$U(m\,x\,k)$矩阵，每个用户由k个特征描述；物品特征矩阵为$V(n\,x\,k)$矩阵，每个物品由k个特征描述，将它们保存在MongDB中。再计算每一个用户最近对电影的K次评分，存储在Redis中。接下来读取数据库的数据，计算备选电影的推荐优先级，更新对userId的实时推荐结果。流程图如图2所示。

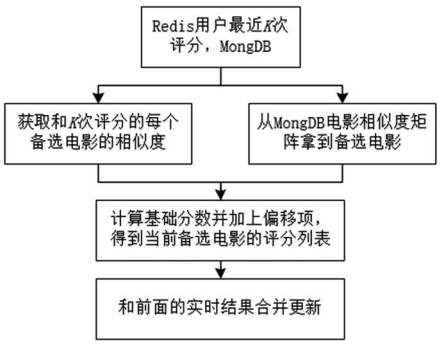

图2 算法流程图

Fig.2 Algorithm flow chart

5 实验结果分析(Analysis of experimental results)

5.1 实验数据集

本文使用公开MovieLens数据集，在Spark平台和单机系统分别对比实时推荐算法在相似度修改前和修改后(实时推荐算法-1)，以及基于物品的协同过滤算法在相似度修改前(ItemCF)和修改后(ItemCF-1)的推荐效果。此处选择的数据大小为10 MB，共10,000部电影、72,000名用户及1,000万条评分数据。评分取值为1—5，得分越高，用户对电影的评价越高。

5.2 评测标准

实时推荐算法的评价标准一般包括推荐时间效率、推荐质量和推荐多样性，它们分别反映了推荐算法的实时响应、推荐性能和推荐过程中用户的个性化请求。其中，推荐时间效率通过执行算法的时间来衡量，推荐的质量可以通过准确率和召回率来衡量[22-23]。

准确率：推荐正确的电影数量占推荐电影总数的比例，公式如式(3)所示。

$$Precision=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|R(u)|} \tag{3}$$

召回率：在推荐结果中满足用户要求的电影数量占所有满足要求的电影数量之比，公式如式(4)所示。

$$Recall=\frac{\sum_{u\in U}|R(u)\cap T(u)|}{\sum_{u\in U}|T(u)|} \tag{4}$$

其中，$R(u)$是基于用户在训练集上的行为向用户做出的推荐列表，$T(u)$是测试集上的用户行为列表。

5.3 实验结果与分析

论文选择$Recall$和算法执行时间作为推荐算法的评测标准。实验随机取数据集中的80%作为训练集，20%作为测试集。分别计算四种算法的$Recall$值和执行时间，经过八次计算，结果如图3和图4所示。

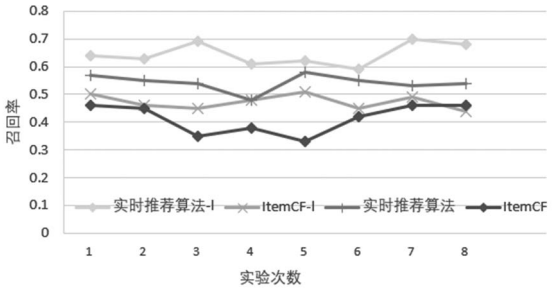

图3 实时推荐算法与ItemCF修改相似度前后的召回率对比图

Fig.3 Comparison graph of recall rate before and after modifying similarity between real-time recommendation algorithm and ItemCF

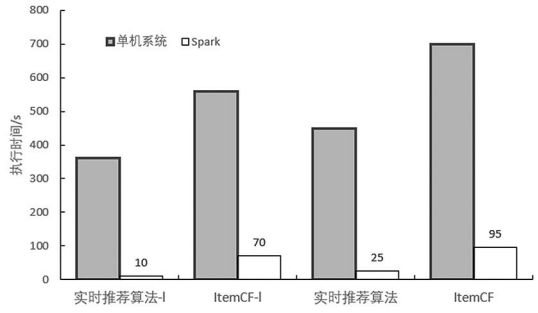

图4 算法执行时间对比图

Fig.4 Comparison graph of algorithm execution time

由此可以看出，本文提出的算法效果要比传统的协同过滤算法在性能上有较大的提升。Spark平台推荐算法并行化实现，实时推荐算法的部分运行结果如图5所示。根据显示的结果，推荐算法成功地将用户的隐式行为与实际行为结合起来，可以根据用户的当前反应进行实时推荐，推荐结果更符合用户的当前意图。

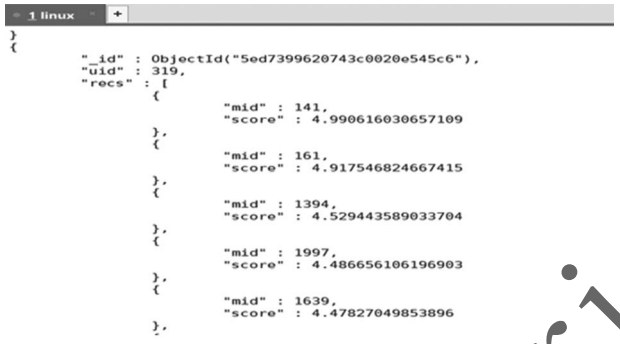

图5 部分推荐结果

Fig.5 Partial recommendation results

6 结论(Conclusion)

本文提出了实时推荐算法并且设置了一种改进的余弦相似度公式，以减少活跃用户对相似度的干扰。与传统推荐算法相比，真实数据集的实验检验了该算法在电影实时推荐中的准确性与时效性。但是，模型在评价指标的提升上没有达到预期，这可能是模型的调参问题，或推荐模型的选择问题，或特征工程不够完善的问题，有待于进一步研究。

参考文献(References)

[1] 陈玮瑜.互联网时代信息超载问题研究[J].传播力研究,2019,3(08):243.

[2] 李学超,张文德,曾金晶,等.推荐系统领域研究现状分析[J].情报探索,2019,26(01):112-119.

[3] 刘君良,李晓光.个性化推荐系统技术进展[J].计算机科学,2020,47(07):47-55.

[4] 周万珍,曹迪,许云峰,等.推荐系统研究综述[J].河北科技大学学报,2020,41(01):76-87.

[5] 严磊,汪小可.基于Spark流式计算的实时电影推荐研究[J].软件导刊,2019,18(05):44-48.

[6] CHEN M, MAO S W, LIU Y H. Big data: A survey[J]. Mobile Networks and Applications, 2014, 19(2):171-209.

[7] 蔡江辉,杨雨晴.大数据分析及处理综述[J].太原科技大学学报,2020,41(06):417-424.

[8] 孙丽.常见大数据处理框架比较研究[J].电脑知识与技术,2020,16(12):3-5.

[9] 宋泊东,张立臣,江其洲.基于Spark的分布式大数据分析算法研究[J].计算机应用与软件,2019,36(01):39-44.

[10] 须成杰,肖喜荣,张敬谊.基于Spark的大数据分析平台的设计和应用[J].中国卫生信息管理杂志,2019,16(05):633-637.

[11] 唐未香,吴学杨,刘科峰.Spark分布式集群的搭建[J].福建电脑,2020,36(02):102-104.

[12] GUEVARA S. Improve your search engine experience[J]. Information Today, 2020, 37(7):20.

[13] 周雪梅.用户兴趣建模支持下的行为推荐算法特性分析[J].现代信息科技,2019,3(09):11-13.

[14] 杨李婷,陈翰雄.用户兴趣建模综述[J].软件导刊,2015,14(10):20-23.

[15] 翁小兰,王志坚.协同过滤推荐算法研究进展[J].计算机工程与应用,2018,54(01):25-31.

[16] ZHANG F, GONG T, LEE V E, et al. Fast algorithms to evaluate collaborative filtering recommender systems[J]. Knowledge-Based Systems, 2016, 96(15):96-103.

[17] TAO J H, GAN J H, WEN B. Collaborative filtering recommendation algorithm based on Spark[J]. International Journal of Performability Engineering, 2019, 15(3):930-938.

[18] 蒋宗礼,于莉.基于用户特征的协同过滤推荐算法[J].计算机系统应用,2019,28(08):190-196.

[19] 焦富森,李树青.基于物品质量和用户评分修正的协同过滤推荐算法[J].数据分析与知识发现,2019,3(08):62-67.

[20] 李亚欣,蔡永香,张根.结合实时推荐与离线推荐的推荐系统[J].计算机系统应用,2019,28(10):45-52.

[21] 刘宇,周虎.基于Spark Streaming实时推荐系统的研究与设计[J].计算机与数字工程,2020,48(05):1172-1175.

[22] 朱郁筱,吕琳媛.推荐系统评价指标综述[J].电子科技大学学报,2012,41(02):163-175.

[23] DUBE K, KAVU T D, RAETH P E, et al. A characterisation and framework for user-centric factors in evaluation methods for recommender systems[J]. International Journal of ICT Research in Africa and the Middle East, 2017, 6(1):1-16.

作者简介：

牛路帅(1997-)，男，硕士生.研究领域：大数据，推荐系统.

彭　龑(1967-)，男，博士，教授.研究领域：计算机应用.