安徽农业大学学报, 2018, 45(4): 768-771
Journal of Anhui Agricultural University
[DOI] 10.13610/j.cnki.1672-352x.20180825.002　网络出版时间：2018/8/27　17:05:10
[URL] http://kns.cnki.net/kcms/detail/34.1162.s.20180825.1656.003.html

面向 Spark 的图书借阅数据关联模型的研究

高琪娟[1]，刘　锴[2]，陈　佳[3]
(1. 安徽农业大学信息与计算机学院，合肥 230036；2. 安徽农业大学现代教育信息中心，合肥 230036；
3. 中国电信芜湖分公司高端聚类服务中心，芜湖 241003)

摘　要：为了方便读者能在海量的图书资源中快速有效的找到需要的书籍，利用 MapReduce 框架分块处理，结合关联分析 Apriori 算法，将数据挖掘技术应用到图书管理系统中。但需要多次扫描数据库和产生大量候选集，对 Hadoop 平台处理速度带来了巨大挑战，因此，针对传统的 Apriori 算法，提出基于内存计算、弹性分布式数据集处理的 Spark 平台为读者推荐书籍，指引读者的借阅行为。

关键词：Apriori 关联规则；Spark 平台；图书借阅行为模式；频繁项集
中图分类号：TP391.3　　**文献标识码：**A　　**文章编号：**1672−352X (2018)04−0768−04

Research of associative model for libraries' book lending data based on the spark

GAO Qijuan[1], LIU Kai[2], CHEN Jia[3]
(1. School of Information and Computer Science, Anhui Agricultural University, Hefei 230036;
2. Modern Educational Technology Center of Anhui Agricultural University, Hefei 230036;
3. High-standard clustering Service Center Department of China Telecom Co., Ltd., Wuhu 241003)

Abstract: In order to search the required books from a tremendous amount of resources immediately for authors, we tried to use the method of MapReduce for dealing the process of block data, combining the algorithm of Apriori, and applying data mining technology to the library management system. But the method referring to above need scan database many times and emerge a large number of candidate set, which will produce tremendous challenge to the speed with Hadoop processing method. Thus, compared to the tradition method of Apriori, there is a new method based on the memory and RDD to compute in Spark platform to recommending books for readers and guiding their borrowing behavior.

Key words: apriori associative rules; spark platform; the borrow behavior model of books; frequent itemset

高校学生面对日益庞大的图书资源，如何有效选择适合的书籍，需要消除信息孤岛，解决内部信息系统缺乏连通性和联动性等问题，为高校图书馆提供综合性和个性化服务，从而实现了从传统的查询图书名目的模式向自动推荐方式发展。在传统数据技术编辑保存的图书采购、流通和编目数据中，通过设计关联模型，挖掘出图书借阅行为模式，指引学生借阅行为，能够迅速查看有价值的图书资源。

近年来，随着国内高校图书馆的规模越来越大，图书推荐服务在理论方面有了一些进展，相关算法被相继的提出和改进，其中王景艳[1]对关联规则经典算法 Apriori 进行了改进，减少候选项集的数量及实际考虑的事物数，利于从大量数据中找到隐藏的规则，进而挖掘出读者的借阅兴趣。

通常，利用数据挖掘技术如关联规则、聚类分析方法，基于并行处理数据的开源平台对高校图书管理系统的业务流程进行研究，例如基于 Hadoop 平台提出一种 MapReduce 方式并行处理的 Apriori 算法，改进和优化频繁项集的产生，在海量数据挖掘中大大提高效率。不足之处是该算法需要大量迭代运算[2]，而大量多年保存及当前使用的图书借还信息给这类算法带来巨大的计算挑战。为了解决该问题，其中以经典的 Apriori 关联规则挖掘算法，基于 Spark 框架进行改进，提出分布式并行化算法[3]，

收稿日期：2018-01-05
作者简介：高琪娟，中级工程师。E-mail：grace@ahau.edu.cn

适用于大数据关联规则的挖掘。以图书馆借阅数据为基础，挖掘学生个性化阅读信息，通过改善评分数据稀疏性，对学生借阅行为的数据分析，提炼出学生的阅读偏好，提高推荐效果，从而能为提供个性化的知识推荐做好基础工作（见图 1）。

目前研究使用较多的方法是基于开源集群计算框架，分布式内存计算以及弹性分布式数据集的RDD，非常适合于迭代型的计算[4]。本研究针对图书数据的关联稀疏性以及处理速度的问题，提出基于 Spark 平台，利用关联规则的布尔矩阵 Apriori 算法，剔除无效规则，提高挖掘质量，减少算法运行时间。将所得的规则导入数据库形成知识库，提供读者个性化服务。

图 1　安农大图书借阅检索系统

Figure 1　The system of AHAU library searching

1　Spark 的关键技术

Spark 基于 Map-Reduce 算法模式实现的分布式计算，作为一个通用的大规模数据快速处理引擎，提供了非常容易使用的编程接口将节点集群数据集缓存到内存中。提供 4 个范畴的计算框架，分别为：支持流处理、并行图计算、底层分布式机器学习库和机器学习功能，支持结构化数据的 SQL 查询及分析查询，适合各种迭代算法和交互式查询分析，缩短访问延迟，提供交互式查询。相对于 Hadoop，Spark 在性能方面更适用于需要多次操作特定数据集的应用场合，同时还支持内存是存储和高效的容错机制。

1.1　弹性分布式数据集 RDD

Spark 是基于内存计算，且核心抽象模型 RDD（弹性分布式数据集），是一个不可变的分布式对象集合。每个 RDD 都被分为多个分区，这些分区运行在集群中不同节点上[5]，图 2 显示了 RDD 转换流程。

Spark 平台能兼容处理 Hadoop 的 HDFS 数据文件，通过转换-transformation（例如 map/filter/groupBy/join 等，区别于 action）创建新 RDD，且具有不可变性，无法更改。其次，RDD 通过转化操作从已有的 RDD 创建出新的 RDD，由于每个 RDD 都保留计算至当今数据的全部历史记录，而且其他进程无法对其做出更改。因此，当某个节点丢失数据时，只需要对该节点 RDD 重新计算即可，并不影响其他节点运行[6]。

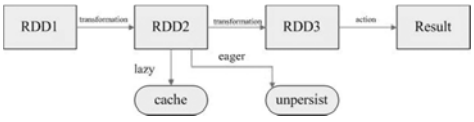

图 2　RDD 转换流程

Figure 2　RDD transformation process

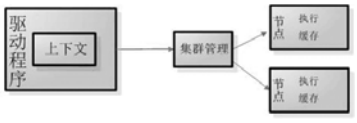

图 3　分布式 Spark 应用中的组件

Figure 3　The component of spark

RDD 的基本操作包括转换和动作，RDD 的转换操作是返回一个新的 RDD 的操作，例如：map、filter 或已有的 RDD 产生新的 RDD，例如 groupBy 和 reduceBy。Action 通过 RDD 计算得到一个或者一组值，例如 count()，take()和 collect()[7]。Spark 中 RDD 的转化操作都是惰性求值的，被调用行动操作之前不会开始计算，只是会在内部记录所要求执行的操作的相关信息。

1.2　Spark 应用程序框架

Spark Application 的运行架构由 2 部分组成：

driver program（SparkContext）和 executor[8]。图 3 显示了分布式 spark 应用的组件结构，其中 SparkContext 封装了 spark 执行环境信息。在分布式环境下，driver 节点负责中央协调，调度各个分布式工作节点，执行器 Executor 是与之对应的工作节点，驱动节点可以和大量的执行器的节点通信。Spark Application 一般都是在集群中运行，如 standalone、yarn 和 mesos 等。通过集群管理器（Cluster Manager）自带的独立集群管理器 Standalone 或外部服务在集群中机器上启动，如 Hadoop YARN 和 ApacheMesos。这些集群中提供了计算资源和资源管理，既可以给 executor 执行，也可以给 driver program 运行。

2 基于 Spark 并行化 Apriori 算法

Apriori 算法是关联规则挖掘中经典算法之一，图书借阅数据记录了读者选择图书种类的多条记录，通过挖掘频繁项集，可以从大量的数据集发现隐藏的有意义的联系，从中找出读者选择书籍的关联性。但 Apriori 算法在判断其是否为频繁项集时，需要逐层扫描每个候选项集，多次迭代才能确定。

2.1 传统的 Apriori 算法设计

关联分析是 Agrawal 于 1993 年提出的，对顾客在商店购物零售数量进行分析，从中找出那些很可能被同时购买的商品的集合，帮助销售商有目的为顾客推荐商品，有效引导购物[9]。

Apriori 算法是挖掘关联规则的频繁项集。首先定义 2 个互相独立的集合 A 和 B，假设 A 和 B 之间有一定的关联性，存在关联规则。用支持度和置信度来说明：（1）支持度 support $(A \Rightarrow B)$=P()，即的同时出现的概率；（2）置信度 A 和 B 在一定条件下出现的概率，即 confidence$(A \Rightarrow B)$ = P(B|A)，揭示了 A 出现时，B 是否也出现或有多大的概率出现。

传统的 Apariori 算法采用逐层迭代方法，从 k 项集推出 k+1 项集，找到频繁项集，生成关联规则。

输入：数据集 Dataset，最小支持度阈值 minSup=0.4 和置信度 support=0.7。

输出：K-项频繁集 Lk；

求频繁项 1 项集 L1，先扫描数据集 D，以集合 I 作为候选项集 C1，获取 C1 的支持度，通过逐层扫描统计候选项集中每个项集 X 的支持度，删除没有达到阈值的项集 X。找到 minSup 的元素作为频繁 1 项集 L1[8]；

继续逐层扫描扫描数据集 D，频繁集 L1 再进行自身连接生成候选集 Ck 的支持度，删除没有达到阈值的项集 X。找到 minSup 的元素作为频繁集 Lk；

通过频繁项集 k 项集 Lk 产生 k+1 候选集 Ck+1；

通过迭代以上步骤，直到不能找到 k+1 项候选集，最终得出最大频繁项集 Lk。

该方法每次搜索需要完整扫描一次数据库，挖掘海量数据时，CPU 时间和内存消耗大；关联规则挖掘多次迭代搜索候选集，产生巨大数量的候选集，该模型较复杂无法适应大数据环境。

2.2 基于 Spark 并行化 Apriori 算法

结合 Spark 的框架技术，把数据库中的事务数据并行均衡分发给多个子节点，设计以局部查找频繁项集、剪枝取代全局操作，防止内存无法容纳的问题，实现实时数据集出现的次数、过滤支持度的项集以及排序等，实现对整个挖掘频繁集和生成规则的过程并行化，从而提高关联规则挖掘的效率。根据 Spark 建立的生态环境，Spark 框架搭建主要是基于对 RDD 的多种操作。方法如下：

（1）Master 利用 Spark 提供的算子获取总事物集，对数据源进行预处理扫描存储在 HDFS 上的事务数据库，即 RDD；Worker 利用 count 求 1 项集的集合 L1 和候选 11 项集 C1。

（2）RDD 被分成 n 个数据块，被分配到 m 个 worker 节点进行操作处理，根据 worker 节点上 1-项集 Itemsets，方式生成局部 K-项集 subitemset，通过函数 f(i) =>i.filter(_>= minSup)对 worker 数据进行过滤。

（3）设置关联规则最小阈值支持度 minSup，利用局部剪枝性质，删除局部支持度小于阈值的项集。

（4）利用 map、reduceByKey、filter 组合进行每一轮局部剪枝，针对剪枝触发 foreach，进行全局连接。

（5）结合频繁项集时序规则进行 filter 产生有序规则 Lk

3 基于 Spark 的图书借阅数据并行化 Apriori 算法的实现

以安徽农业大学图书馆 2010—2015 年 6 年图书借阅历史数据为实验数据，对上文提出的读者借阅行为的关联规则模型进行建模。

以 Spark 平台为基础，结合 Apriori 算法并行化实现的核心是在内存中迭代调用 transformation 和 action 操作，每次迭代是在上一次迭代结果进行求解，算法并行化伪代码实现步骤如下：

输入：数据源路径 input path，数据集 Dataset（一般存储在本地文件系统或存储在 Hadoop 分布

式文件系统中）最小支持度阈值 minSup=0.4 和置信度 support=0.7。

输出：K-项频繁集 Lk；输出路径 output path。

（1）求 L1

```
val rdd = sc.textFile(“/”); //构造函数，
L1=rdd.map(_,1).reduceBykey(_+_).filter(_>min_sup)
val nums = rdd.count();//计算 1 项集总类别数。
```

（2）每行记录进行分割，求出每一行事物的所有项集候选项集

```
Valitemsets = rdd.flatMap(x =>x).filter(_.size>0)
Val minSup = 0.4
Val combined=itemsets.reduceByKey(_ +_).map (x => x/nums)
If (1 int) {
  result
} else { for (0<i< maxL) {
Result ++= L
}
```

（3）计算同组置信度

```
Val rules=subitemsets.groupByKey() // 将 Key 相同归为一组
Val assocRules = rules.map (x =>
listX = x._1
listY= ._2.toList
if( newList .isEmpty){
result
) else {
Val result =newList.map()
Result
}//将同一组出现的次数和规则做不相及出现的次数计算置信度
Val minConf=0.7//设置置信度阈值
Val finalResult = assoRules.flatMap (x=>x) filter(_>=minConf)//flatMap 和 filter 生成规则过滤掉置信度没有达到置信度阈值的规则。
return L//最后返回所有关联规则的列表。
```

图 4 显示是基于 spark 平台，通过并行化 Apriori 算法实现的结果。

图 4 图书关联规则运行结果

Figure 4 The result of apriori-rules

4 结论

以读者借阅图书历史记录为基础，建立基于读者身份、读者专业、图书类型等多维属性读者借阅关联规则挖掘模型[10]，利用 Spark 框架技术，运用 Apriori 算法的剪枝思想改进频繁项，所建模型可以挖掘读者借阅图书类型之间关联关系，提高关联效率。

参考文献：

[1] 王景艳. 基于改进关联规则挖掘算法的图书推荐服务[J]. 福建电脑, 2008,24 (5): 72-73.

[2] 王青, 谭良, 杨显华. 基于 Spark 的 Apriori 并行算法优化实现[J]. 郑州大学学报(理学版), 2016, 48(4): 60-64.

[3] 车晋强, 谢红薇. 基于 Spark 的分层协同过滤推荐算法[J]. 电子技术应用, 2015, 41(9): 135-138.

[4] 牛海玲, 鲁慧民, 刘振杰. 基于 Spark 的 Apriori 算法的改进[J]. 东北师大学报(自然科学版), 2016 ,48(1): 84-89.

[5] 陈方健. 图书借阅行为模式挖掘方法在学生借书管理系统中的应用[D]. 苏州:苏州大学, 2014.

[6] 景民昌, 于迎辉. 基于借阅时间评分的协同图书推荐模型与应用[J]. 图书情报工作, 2012, 56(3): 117-120.

[7] 金瑶. 数据挖掘技术在高校图书馆管理系统中的应用[D]. 上海:华东师范大学, 2010.

[8] 王飞. 基于数据挖掘的高校图书馆个性化推荐服务的应用研究[D]. 呼和浩特: 内蒙古工业大学, 2015.

[9] 李现伟. 基于 Spark 的推荐系统的研究[D]. 杭州: 浙江理工大学, 2016.

[10] 王家胜, 牟肖光. 读者借阅多维关联规则挖掘模型的建立与分析[J]. 计算机应用, 2011, 31(11): 3084-3086.