基于 Spark 的密度聚类算法并行化研究

朱子龙,李玲娟
(南京邮电大学 计算机学院,江苏 南京 210023)

摘 要: 聚类分析目前是数据挖掘研究领域中热门的研究课题,DBSCAN 算法则是聚类分析中较为重要的一种基于密度的算法。Apache Spark 扩展了广泛使用的 MapReduce 计算模型,提出了基于内存的并行计算框架。通过将中间结果缓存在内存中减少 I/O 磁盘操作,使其能够更高效地支持交互式查询、迭代式计算等多种计算模式。为了更好地进行大数据聚类挖掘,研究如何对基于当今主流的大数据处理框架 Spark 对 DBSCAN 算法进行并行化。设计了基于 Spark 的 DBSCAN 算法并行化方案,通过合理利用 RDD 和设计 Sample 算子、map 函数、collectAsMap 算子、reduceByKey 算子,实现了对寻找核心对象的密度可达数据点过程的并行化。在 Spark 平台上运用 DBSCAN 算法对 UCI 的 Wine 数据集、Car Evaluation 数据集和 Adult 数据集的并行化聚类结果表明,并行化的 DBSCAN 算法具有较好的准确性和时效性,适用于大数据聚类。

关键词: DBSCAN; 聚类; Spark; 并行化
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1673-629X(2018)06-0080-05
doi: 10.3969/j.issn.1673-629X.2018.06.018

Research on Parallelization of Density Clustering Algorithm Based on Spark

ZHU Zi-long, LI Ling-juan
(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: Clustering analysis is currently a hot research topic in data mining, and DBSCAN algorithm is a density-based algorithm which is more important in clustering analysis. Apache Spark extends the widely used MapReduce computing model and proposes a memory-based parallel computing framework. It reduces I/O disk operations by caching intermediate results in memory, enabling it to more efficiently support multiple computing modes such as interactive queries, iterative calculations and so on. In order to mine the large data well, we study how to parallelize the DBSCAN algorithm based on large data processing framework Spark, and design a scheme on parallelization of density clustering algorithm based on Spark. Through the rational use of RDD and design of Sample operator, map function, collectAsMap operator, reduceByKey operator, it realizes the parallelization of the process of finding the density-reach data points for the core object. With DBSCAN algorithm on the Spark platform on the data set of UCI Wine, Car Evaluation and Adult, the parallel clustering results show that the parallelized of DBSCAN algorithm has better accuracy and timeliness, and it is suitable for large data clustering.

Key words: DBSCAN; clustering; Spark; parallelization

0 引 言

聚类是将未知数据集合分组成由较高相似度的对象组成的若干类或者簇的过程[1]。聚类算法大致分为划分方法、层次方法、基于密度的方法、基于网格的方法和基于模型的方法等五类[2-3]。DBSCAN 算法作为典型的基于密度的方法,具有聚类速度快、有效处理“噪声”点并且能够发现任意形状的簇等优点[4],因此研究用该算法高效并行化处理海量数据具有重要的现实意义。

Hadoop 作为近年来比较流行的大数据处理平台,利用 HDFS 分布式海量存储和 MapReduce 分布式计算框架并行处理,其每次 map 计算过程中产生的中间结果需要反复读写本地磁盘,在进行大量迭代计算时,MapReduce 计算模型将会耗费大量读写时间[5]。2009 年加州大学伯克利分校创立基于内存的 Spark 大数据处理计算框架,更好地支持交互式查询和迭代算法,扩展了 MapReduce 计算框架,并且支持内存式存储和高效的容错机制。

收稿日期: 2017-07-10 **修回日期:** 2017-11-15 **网络出版时间:** 2018-02-24
基金项目: 国家自然科学基金(61302158,61571238)
作者简介: 朱子龙(1991-),男,硕士研究生,研究方向为数据挖掘;李玲娟,教授,研究方向为数据挖掘、信息安全、分布式计算。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20180224.1521.068.html

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

文中研究了 DBSCAN 算法在 Spark 平台的并行化实现方案,并通过实验验证了方案的高效性。

1　DBSCAN 聚类算法原理

DBSCAN 算法是一种典型的基于密度的聚类算法,它将簇定义为高密度相连的点的最大集合。该算法能够将高密度的数据区域分为不同的簇,能够在具有"噪声"的数据集中识别出任意形状的聚类[3]。对于数据量为 n 的数据集合,按照空间索引方法,DBSCAN 的计算复杂度是 $O(n\log n)$,否则其计算复杂度为 $O(n^2)$。

(1) DBSCAN 算法涉及到的定义。

定义 1(Eps 邻域): 对指定数据集 D 中以 x 为圆心的半径 Eps 内的球形区域称为该点 x 的 Eps 邻域。

定义 2(核心对象): D 中任意一点 x 的 Eps 邻域内包含大于或等于最小数目 MinPts 个对象,则称该点 x 为核心对象。

定义 3(边界对象): D 中任意一个不是核心对象的点,当其在其他核心对象的 Eps 邻域中时,称它为边界对象。不同核心对象的 Eps 邻域中可能会有相同的边界对象。

定义 4(直接密度可达): 如果 D 中的点 y 在点 x 的 Eps 邻域中而且点 x 是核心对象,则称点 y 是从点 x 关于参数 Eps 和 MinPts 直接密度可达的[4]。

定义 5(密度可达): 在给定半径 Eps 和 MinPts 的数据集 D 中,存在对象链 $m_1,m_2,\cdots,m_n$,其中 $m_1=x$,$m_n=y$,对 $m_i\in R(1\leqslant i\leqslant n)$,如果 m_{i+1} 是从 m_i 直接密度可达的,则点 y 是从点 x 关于 Eps 和 MinPts 密度可达的。密度可达关系是不对称的。

定义 6(密度相连): 如果 D 中的点 x 和点 y 是从点 p 关于 Eps 和 MinPts 密度可达的,则称点 x 和点 y 是密度相连的[4]。

定义 7(簇): 基于密度的 DBSCAN 算法的簇是密度相连的数据点的最大集合[7]。对给定数据集 D 的任意一个非空子集 R,如果称之为簇,必须满足如下条件:

①最大性: 任意对象点 x,$y\in D$,若 $x\in R$,且点 y 是从 x 密度可达的,则 $y\in R$。

②连通性: 任意对象点 x,$y\in R$,则点 x,y 是密度相连的。

定义 8(噪声): 如果 D 中的某点不被包含在任意簇中,则称为噪声。

综上所述: 当基于密度聚类时,数据集中的簇看作是被低密度区域分隔开的高密度数据区域[8]。数据集中的核心对象一定属于某簇,而且密度相连的点在同一簇中。噪声点是数据集中的干扰数据,会被舍弃。

(2) DBSCAN 算法的基本思想。

首先从给定数据对象集中随机选定一个点 x,在该点给定半径 Eps 的区域寻找聚类。如果点 x 的 Eps 邻域内至少包含 MinPts 个对象,那么以点 x 为核心对象创建一个新簇,接着反复基于这些核心对象查找直接密度可达的数据对象,查找过程可能会涉及密度可达簇的相关合并[3]。直到没有新的点被合并到其他簇时,算法结束。

(3) DBSCAN 算法的具体步骤。

输入: 包含 n 个数据对象的数据集 $D=\{x_1,x_2,\cdots,x_n\}$,半径 Eps 和最小对象数目 MinPts。

输出: 簇集合 $\{R_1,R_2,\cdots,R_n\}$。

①输入待处理数据集 D 后,任意选择一个数据点 x,检查该对象的 Eps 邻域;

②如果数据点 x 的 Eps 区域内至少包含最小对象数 MinPts,则以点 x 为核心对象形成新簇并从点 x 出发寻找所有密度可达的数据点,随之更新簇;

③如果数据点 x 不是核心对象,将点 x 当作噪声点处理;

④repeat 以上步骤;

⑤until 无新的数据点加入任何簇。

2　Spark 大数据计算框架

不同于 Hadoop 的 MapReduce 计算模型,Spark 能够使 job 中间结果保存在内存中,减少对磁盘的大量读写操作,从而可以高效低延迟处理大型数据集[9-11]。Spark 引入了弹性分布式数据集(resilient distributed dataset,RDD)概念,实现了任务调度、分发和处理等,同时提供了更多计算模式组件,如 SparkSQL、Spark Streaming、MLib 和 GraghX 等,可以适用于多种分布式平台场景。

Spark 的 RDD 核心概念,让其能够以基本一致的操作方式去处理不同的应用场景。RDD 本质上是一个不可变只读的分布式元素集合,每个 RDD 包含不同的分区,这些分区就是多个 dataset 片段,它们分别运行在不同的集群节点上可被同时并行处理。实际上,Spark 并行框架计算流程就是通过待处理数据创建 RDD、转化成新的 RDD 和调用 RDD 行动操作求值得到结果[12]。RDD 支持两种操作类型: 转化(transformation)和行动(action)。其中转化操作是将现有的 RDD 转化成一个全新的 RDD,Spark 中转化操作都是惰性求值,只有在行动操作实际用到这些 RDD 时才会被计算。而行动操作会触发实际计算,并且向驱动器程序返回结果或者将结果存入外部存储系统中。通常情况下,每个转化过的 RDD 会在对其执行行动操作时被重新计算,但是大数据迭代算法中经常会多次使用

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

同一 RDD，为避免多次调用行动操作对同一 RDD 计算而带来的开销，Spark 支持对 RDD 进行持久化，计算出 RDD 的节点会分别保存其所求得的分区数据。

Spark 在分布式环境中采用了主从结构计算模型，其中包含驱动器(driver) 节点和执行节点(见图 1) 。

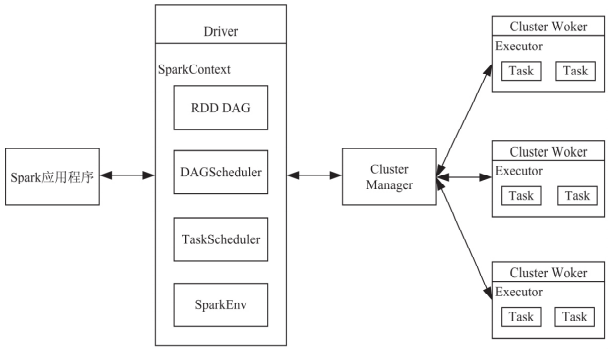

图 1 Spark 计算模型

驱动器节点运行 Application 中的 main( ) 方法，创建 SparkContext、创建 RDD 和对 RDD 转化、执行行动操作，它作为应用逻辑执行起点负责将用户程序转化为多个物理执行单元，然后将其分发到不同执行器节点执行处理，并且会根据执行器节点任务处理情况，负责执行器节点的任务调度以尽可能地提高效率。执行器节点接收到驱动器节点指令后，主要负责分区中任务执行并将任务执行结果反馈给驱动器节点，执行器节点可以为需要缓存的 RDD 进行内存式存储，从而可以提高运行效率。

3 基于 Spark 的 DBSCAN 算法并行化方案

DBSCAN 算法定义类簇是密度相连点的最大集合，需要反复迭代寻找核心对象的密度可达数据点[13]，所以可以使用 Spark 大数据计算框架 RDD 实现。在 Spark 应用中一个驱动器(Driver) 程序定义了集群中执行器(executor) 节点上的分布数据集，并实现任务分发、调度、执行和聚合结果等操作。文中设计的 DBSCAN 算法并行化方案如下:

(1) 配置 Spark。

首先，驱动器程序创建 SparkConf 对象，配置 Spark 如何连接到相关集群中，然后创建 SparkContext 对象来连接访问 Spark。如前文所说，Spark 并行框架计算流程实际上是通过待处理数据创建 RDD，转化成新的 RDD，并调用 RDD 行动操作求值得到结果。一般可以通过两种方式创建 RDD: 读取外部文件系统中的数据集或者在驱动器程序中对数据集进行并行化，同时 Spark 支持常用文件格式和文件存储系统，比如 HDFS、Amazon S3、HBase 等。读取待处理数据集创建 RDD 后，分发到集群中各个执行器节点中，转化为 Spark 中数据块保存在内存或者磁盘中，并通过 BlockManager 进行管理。RDD 中 partition 是逻辑数据块，分别对应 BlockManager 管理的物理分区中相应的 Block。由于 Spark 采用惰性求值的方式节省集群中的内存使用，只有 RDD 行动操作才能触发 Job 的提交计算。RDD 的 Action 算子触发 Job 提交，Spark 收到 Job 后生成具有逻辑性的有向循环图(RDD DAG) ，随后 DAGScheduler 会对 DAG 进行 Stage 划分。对应的每个 Stage 都会生成一组 Task 集合并提交到 TaskScheduler 中，会由 TaskScheduler 将 Task 调度分发到各个执行器节点的线程池中执行。

(2) 多执行器节点并行执行 DBSCAN 算法。

DBSCAN 算法基于 Spark 的并行化流程见图 2。

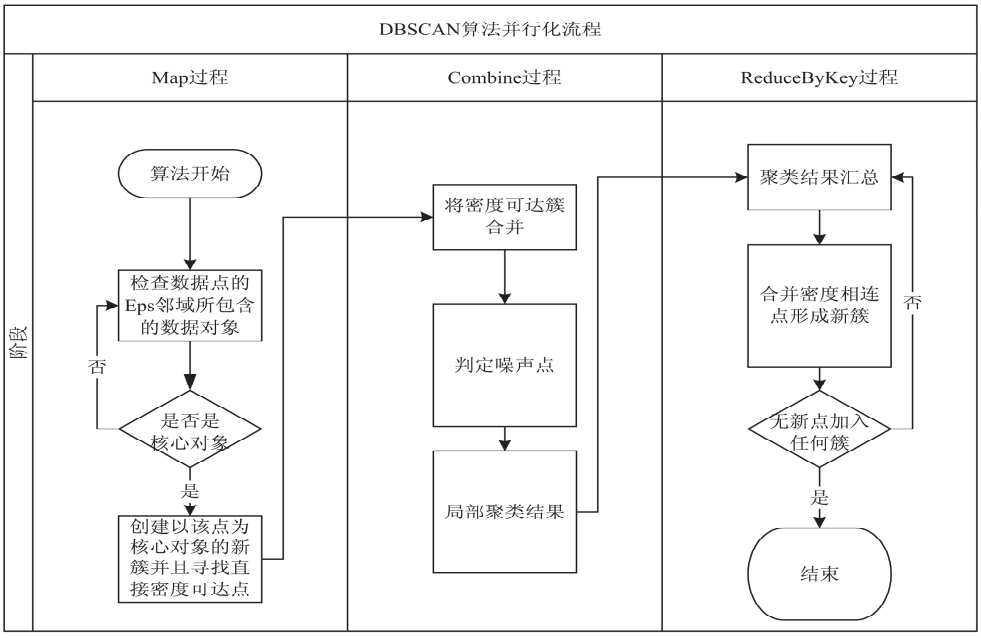

图 2 DBSCAN 算法并行化流程

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

每个执行器节点通过多线程方式对其 RDD 分区内容中的数据使用 DBSCAN 算法进行计算,首先读取数据集形成 RDD_1,RDD_1 启动 Sample 算子,逐一随机选取某数据点 x 作为起始点并转化为 RDD_2。设计 map 函数,计算点 x 的 Eps 邻域内是否包含大于或等于 MinPts 个数据对象,以判断其是否为核心对象,如果是则形成新簇,生成 RDD_3。RDD_3 启动 collectAsMap 算子,将已处理的相同类簇汇总到 RDD_4 同一数据分片中,RDD_4 启动 reduceByKey 算子,寻找从核心对象出发直接密度可达的数据点,此过程中可能存在密度可达的类簇合并,重复迭代,直到无新的数据点加入任何簇时输出聚类结果。

执行过程中,执行器节点会将需要缓存的 RDD 缓存在内存中,并将各自处理数据汇总到驱动器节点,并再次迭代计算后终止。此时驱动器节点调用 saveAsTextFile 算子,将结果存储到分布式存储系统 HDFS 中。最后,通过调用 SparkContext 的 stop 方法退出 Spark 应用。由于 Spark 大数据计算框架的核心数据模型 RDD 是分发到不同 executer 节点上并行计算的,并将中间结果缓存到内存中,因此对于需要大量迭代计算的 DBSCAN 算法可以节省大量时间。

4　实验与结果分析

为了测试和分析基于 Spark 的并行化 DBSCAN 算法的性能,分别用单机 DBSCAN 算法和基于 Spark 的并行 DBSCAN 算法对实验数据集进行聚类操作,在不同运行模式下,对聚类效果准确度和时间效率进行对比。

4.1　实验环境和实验数据

实验搭建的 Spark 集群包含 1 台驱动器节点,2 台执行器节点。每个节点的 CPU 为 Intel CORE i5-4210H,每个节点配有 2 个处理器,硬盘数据读写速度为 600.00 MB/s,其中驱动器节点拥有 6 G 运行内存,执行器节点拥有 2 G 运行内存。操作系统为 centos 6.5; Java 版本为 JDK1.7.0_13; Spark 版本为 1.6.0; Scala 版本为 2.10.4。

具体节点配置见表 1。

表 1　节点配置

编号	角色	内存	处理器
Spark-1	驱动器节点	6 GB	2 Core
Spark-2	执行器节点	2 GB	2 Core
Spark-3	执行器节点	2 GB	2 Core

实验采用了 UCI 实验室提供的 Wine 数据集、Car Evaluation 数据集和 Adult 数据集[14-15],详情见表 2。

表 2　数据集

数据集	属性特征	属性数目	记录数
Wine	整数、实数	13	178
Car Evaluation	类别型	6	1 728
Adult	类别型、整数	14	48 842

4.2　聚类准确度对比实验

以正确聚类的样本数占总样本数的百分比作为准确率,对 DBSCAN 算法的聚类结果进行评估,得到的结果如表 3 所示。

表 3　准确度对比

数据集	属性数目	准确度/%	
		单机 DBSCAN 算法	Spark-DBSCAN 算法
Wine	13	73.10	73.22
Car Evaluation	6	71.86	72.97
Adult	14	77.55	78.33

可以看出,对 3 个数据集,基于 Spark 的并行 DBSCAN 算法比传统单机运行模式的 DBSCAN 算法的聚类准确率都有所提高。算法准确度不会因为数据分布式处理而影响聚类结果,说明分布式数据处理具有良好的稳定性和准确度。

4.3　聚类时效对比实验

图 3 展示了传统单节点 DBSCAN 算法和基于 Spark 的并行 DBSCAN 算法处理不同数据集时的耗时情况。

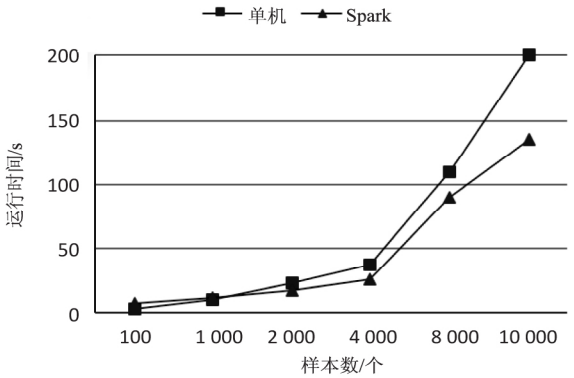

图 3　算法运行效率对比

可以看出,在数据量比较少的情况下,传统单节点 DBSCAN 算法运行时间比基于 Spark 的 DBSCAN 算法运行时间要少,因为 Spark 平台启动时的初始化需要消耗一定时间。伴随着数据量逐渐增大,单节点 DBSCAN 算法执行时间涨幅明显,因为数据逐步增多所消耗的处理器和内存等资源也在逐步增多,单个工

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

作节点由于资源限制,运行速度会逐步降低导致消耗更多时间;相反,基于 Spark 的 DBSCAN 算法运行时间随着数据量增加涨幅明显低于单节点运行模式,此时分布式集群优势逐渐显示出来。可以得出这样的结论:在处理规模较大的数据时,基于 Spark 平台的 DBSCAN 算法聚类时效性更好。

5 结束语

对 DBSCAN 算法如何在 Spark 平台进行并行化实现进行了研究。通过设计该算法在 Spark 集群中的并行化实现方案和对 3 个数据集进行聚类处理,验证了在大规模的数据处理中,基于 Spark 平台的 DBSCAN 算法并行化能够有效完成聚类工作,并且具有更高的准确度和更好的执行效率。

参考文献:

[1] HAN Jiawei,KAMBER M.数据挖掘概念与技术[M].范明,译.北京:机械工业出版社,2001:232-236.

[2] 张 丽.无参数网格聚类算法的研究[D].郑州:郑州大学,2009.

[3] 林建仁.聚类算法的研究与应用[D].上海:复旦大学,2007.

[4] CHEN Yixin,TU Li.Density-based clustering for real-time stream data[C]//Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining.San Jose,California,USA:ACM,2007:133-142.

[5] 李 帅,吴 斌,杜修明,等.基于 Spark 的 BIRCH 算法并行化的设计与实现[J].计算机工程与科学,2017,39(1):35-41.

[6] 张世安.分布式环境下的数据挖掘算法的研究与实现[D].上海:复旦大学,2004.

[7] 范 敏,李泽明,石 欣.一种基于区域中心点的聚类算法[J].计算机工程与科学,2014,36(9):1817-1822.

[8] 高 兵.基于密度的数据流聚类方法研究[D].哈尔滨:哈尔滨工程大学,2014.

[9] 张 泉.面向云计算数据中心的存储服务质量技术研究[D].武汉:华中科技大学,2014.

[10] JAIN A K.Data clustering:50 years beyond K-Means[J].Pattern Recognition Letters,2010,31(8):651-666.

[11] PAN Donghua,ZHAO Lilei.Uncertain data cluster based on DBSCAN[C]//Proceedings of IEEE international conference on multimedia technology.Hangzhou,China:IEEE,2011:3781-3784.

[12] KELLNER D,KLAPPSTEIN J,DIETMAYER K.Grid-based DBSCAN for clustering extended objects in radar data[C]//Intelligent vehicles symposium.Alcala de Henares,Spain:IEEE,2012:365-370.

[13] 唐振坤.基于 Spark 的机器学习平台设计与实现[D].厦门:厦门大学,2014.

[14] 罗启福.基于云计算的 DBSCAN 算法研究[D].武汉:武汉理工大学,2013.

[15] ZAHARIA M,CHOWDHURY M,FRANKLIN M J,et al.Spark:cluster computing with working sets[C]//Proceedings of the 2nd USENIX conference on hot topics in cloud computing.Boston,MA:[s.n.],2010:1765-1773.

(上接第 79 页)

1795-1801.

[11] 韦明祥,陈 俊.基于 C-V 模型的医学图像分割方法[J].电子科技,2012,25(5):101-104.

[12] GENG Hao,LUO Min,HU Feng.Improved self-adaptive edge detection method based on Canny[C]//5th international conference on intelligent human-machine systems and cybernetics.Hangzhou,China:IEEE,2013:527-530.

[13] SU Baofeng,NOGUCHI N.Discrimination of land use patterns in remote sensing image data using minimum distance algorithm and watershed algorithm[J].Engineering in Agriculture,Environment and Food,2013,6(2):48-53.

[14] ZNANATY E A,AFIFI A.A watershed approach for improving medical image segmentation[J].Computer Methods in Biomechanics and Biomedical Engineering,2013,16(12):1262-1272.

[15] ZENG Weili,LU X,TAN X,A local structural adaptive partial differential equation for image restoration[J].Multimedia Tools and Applications,2015,74(3):743-757.

[16] 李雅梅,任婷婷.自适应分数阶微分小波图像增强方法的研究[J].微电子学与计算机,2015,32(6):130-133.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net