

基于 Spark 的并行 K-means 算法研究

许明杰, 蔚承建, 沈 航

(南京工业大学 计算机科学与技术学院, 江苏 南京 211816)

摘 要: 针对 K-means 算法在海量数据的处理过程中, 由迭代计算次数加大导致的内存不足的问题, 提出 Spark 并行化的 K-means 算法. 将粒子群优化(PSO)与 K-means 结合, 利用 PSO 来提高 K-means 的全局搜索能力, 得到初始聚类中心. 利用 Spark 的迭代计算能力, 将 K-means 算法与 Spark 并行框架结合, 提升该算法模型对数据的处理速度, 缩短算法的整体运行时间. 通过疾病检测数据进行实验, 结果表明 Spark 并行化的 PSOK-means 算法在保证准确率的同时大大提高了算法的效率, 本算法对于海量数据的聚类研究有着很好的应用场景.

关键词: Spark; K-means; PSO; 迭代计算

中图分类号: TP181; TP273

文献标识码: A

文章编号: 1000-7180(2018)05-0095-05

Research on K-means Algorithm of Spark Parallelization

XU Ming-jie, WEI Cheng-jian, SHEN Hang

(College of Computer Science and Technology Nanjing Technical University, Nanjing 211816, China)

Abstract: In view of the problem of insufficient memory caused by the increase of iterative computation in the process of mass data processing in K-means algorithm, this paper proposes K-means algorithm of Spark parallization. the algorithm uses particle swarm optimization (PSO) to improve the global search ability of K-means to get the initial clustering center. Through the iterative computing power of Spark, the K-means algorithm is combined with the Spark parallel framework to improve the processing speed of the model and reduce the overall running time of the algorithm. The experiment was carried out by disease detection data, the experimental results show that the Spark parallelized PSOK-means algorithm greatly improves the efficiency and accuracy of the algorithm, It will be good application scenarios for the clustering of massive data.

Key words: spark; k-means; PSO; iterative computing

1 引言

K-means 是解决聚类问题的典型算法, 并且在数据挖掘等众多领域得到广泛应用. 随着大数据时代的来临, 各业界的数据量在体积、速度及多样性方面都呈现飞速的增长. 这样的环境下, 许多传统算法无法适用, 新兴的 Spark 并行计算框架因其处理海量数据的巨大潜力而成为业界争相研究的热点. 面对大量数据, K-means 由于其在串行场景中的时间复杂性进而对此处理的能力不足.

针对 K-means 存在的问题, 许多专家提出了不同

的策略来提高其性能. 总的来说, 这些方法着重于优化初始中心. K. Alsabti^[1] 提出一种 k-d 结构改进 K-means 的聚类策略. 随着群体智能的发展, GA 和 PSO 被引入到 K-means 中. 文献[2]中通过使用 GA 来阻止 K-means 收敛到局部最优. H. Modares^[3] 则提出了一种将 PSO 与 K-means 相结合以增强聚类的方法. 在文献[4]中, 总结了 PSO 和 K-means 聚类结合大致分为三种: (1) K-means + PSO; (2) PSO + K-means; (3) K-means + PSO + K-means. 在三种不同的方法中, 后两种方法具有比较好的效果.

由于 K-means 算法在海量数据方面有缺陷, 所

以本文将并行化的思想引入进来. 本文结合 PSO 和 Spark 并行计算框架, 提出了一种 Spark 并行化的 PSOK-means 聚类算法.

2 Spark 技术

2.1 Spark 系统总览

Apache Spark 是一个通用并行计算框架, 允许通用编程语言进行集群大数据处理. Spark 除了支持迭代机器学习算法、交互式数据分析等外, 也保留了 MapReduce 的可扩展性和容错性.

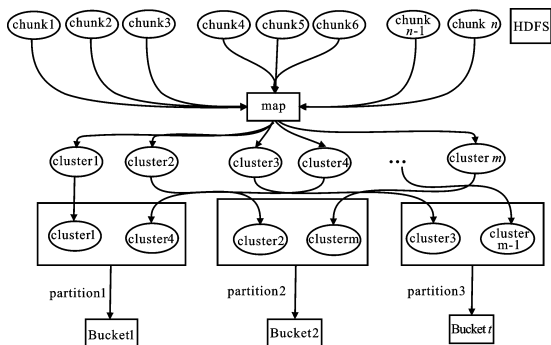


图1 Spark 基于哈希函数的数据分布机制

图1所示的架构中, chunks 是分布在 HDFS 中的数据片段, 每个数据片段都分配给一个 map 任务. map 任务对其进行分割并产生中间输出.

2.2 弹性分布式数据集(RDD)

弹性分布式数据集(RDD)是一个数据集抽象, 其是被 Spark 分散在 Hadoop 集群上的对象集合. RDD 是只读数据集, 任务或作业不能改变 RDD 的元素. RDD 中的所有元素都不会存储在物理存储中; 相反, RDD 的抽象包含了足够的信息来计算 RDD. RDD 可通过读取共享文件系统文件、使用 parallelize() 并行化驱动程序中的集合对象、用户定义已有 RDD 执行 transformation 操作三种方式构建.

RDD 具有显示 RDD 存储级别的持久性状态. 其拥有多个存储级别. 默认情况下, 每次对其执行操作时, 都会重新计算 Spark RDD. 如果某 RDD 将在下一次执行操作中重复使用, 可通过在 RDD 上执行 persist() 改变持久性级别以提高算法性能.

2.3 SparkContext 和并行操作

SparkContext 是在主程序中定义的驱动程序. 其主要负责调度、内存管理、与存储系统的交互及并行任务的故障恢复. SparkContext 主要和集群管理器 Yarn 进行连接. 然后, Spark 尝试获取集群中所有节点上的执行程序, 再接收应用程序代码. 最后,

所有执行者并行运行.

Spark 提供了 Transformations 和 Action 在 RDD 上执行并行操作. 在运行并行任务完成对 RDD 的 action 和 transform 操作中的任务数等于 RDD 的分区数.

3 Spark 并行化的 K-means 算法

3.1 PSOK-means

在 PSO 中, 每个粒子的位置 $x_i(t) = (c_1, c_2, \dots, c_K)$ 代表 K 个聚类中心. 如果每个样本是 N 维, 则每个粒子的位置是 $K \times N$ 维. 并且粒子的速度也是 $K \times N$ 维. 当 PSO 达到最大迭代次数时, 当前全局最佳位置 $g(t)$ 是最优初始聚类中心.

已经证明, 惯性权重 ω 越大, PSO 的全局搜索能力越强, 而 ω 较小时, 其更适用于局部搜索. 在文献[5]中, 随着迭代次数的增加, ω 被设置为线性减少. 在式(1)中, $\omega_{\max}$ 是最大惯性权重, 一般定为 0.9; $\omega_{\min}$ 是最小惯性权重, 一般定为 0.4. 为了防止粒子过快收敛为局部最优, 每个粒子的速度被限制在 $[-v_{\max}, v_{\max}]$.

$$\omega = \omega_{\max} - \text{iter} \times \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} \quad (1)$$

评估聚类结果的关键是建立适应度函数. 已有一个适应度函数, 如下:

$$F(z_i) = 1 / \sum_{j=1}^K \sum_{x_i \in G_j} d(x_i, c_j) \quad (2)$$

该适应度函数表示所有簇的群内距离之和. 在文献[6]中, c_j 是集群 G_j 的中心, 并且 $d(x_i, c_j)$ 是一个样本与集群 G_j 中心之间的距离.

$$J_c = \sum_{j=1}^K \sum_{x_i \in G_j} d(x_i, c_j) \quad (3)$$

公式(3)所示的 J_c 表示所有样本和相应聚类中心之间的距离之和. 为了获得良好的聚类结果, 一个集群中的样本应尽量紧凑. 因此在本文中, J_c 被用作测度聚类结果的适应度函数. 较小的 J_c 是聚类结果越好.

总体而言, 使用 PSO 优化 K-means 初始聚类中心的过程如下:

(1) 初始化一个粒子群. 假设给定的数据集 $S = \{x_1, x_2, \dots, x_n\}$ 具有 n 个 N 维样本. 从 S 中随机选择 K 个样本作为粒子的初始位置 $x_i(t)$. 为粒子设置随机速度 $v_i(t)$ 并设置 $p_i(t) = x_i(t)$. 进行 M 次这样的迭代, 以形成具有 M 个粒子的群;

(2) 设置 ω 为式(1)并在迭代次数增加时进行

更新;

(3) 对于每个粒子,算出所有样本和由 $x_i(t)$ 表示的聚类中心之间的距离. 然后根据算出来的距离将样本分配给最近的聚类;

(4) 对于每个粒子,计算聚类的新中心并更新粒子的位置. 然后根据式(3) 计算适应度;

(5) 对于每个粒子,将其当前适应度与其个人最佳适应度进行比较. 如果当前值更好,则更新其个人最佳位置和个人最佳适应度;

(6) 对于每个粒子,将其当前的个人最佳适应度与全局最佳适应度进行比较. 如果当前值更好,则更新全局最佳位置和全局最佳适应度;

(7) 对于每个粒子,根据(1) 和(2) 更新其速度和位置. 并限制其速度 $[-V_{\max}, V_{\max}]$;

(8) 循环到步骤 2 直达到达 $iter_{\max}$ 或全局最佳位置 F 次保持不变;

(9)使用全局最佳位置作为初始聚类中心,并启动并行化的 K-Means.

3.2 PSOK-means 算法的并行化

Spark 并行化的 PSOK-means 算法通过对 RDD 重复调用 Map 函数和 Reduce 函数实现程序中多次 MapReduce 操作,在多个工作节点上对分割后的数据片段进行操作.

Spark 并行化的 PSOK-means 架构如图 2 所示. 由图 2 可知,该算法在执行时,首先将输入数据按照 batch size 分成多个数据片段,接着将这些数据片段转换成 RDD 缓存到内存中去,根据 Yarn 的任务分配将 RDD 分配到不同的 work 节点上进行算法操作,最终通过主节点汇聚各计算节点的结果.

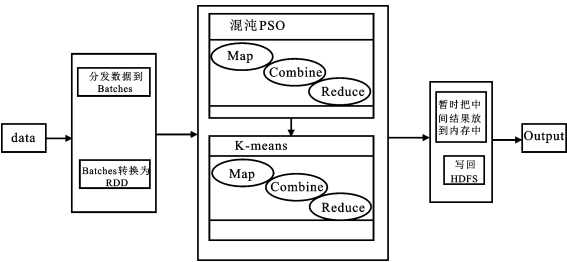


图 2 Spark 并行化的 PSOK-means 策略框架

该算法在执行时,先运行 PSO,得到 K-means 算法的初始聚类中心. 接着再进行 K-means 并行化. K-means 并行化详细流程设计如图 3 所示.

由图 3 可知,该算法并行化设计主要由 Map, Combine 和 Reduce 三个部分组成. Map 和 Reduce

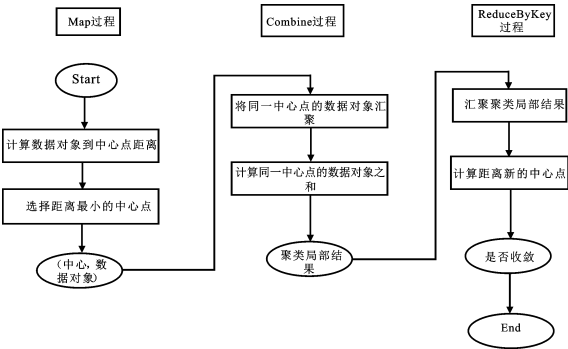


图 3 K-means 算法部分并行化详细流程设计

过程详细算法如下.

```
算法 1:K-means 并行化 map 过程算法
input:<聚类中心,数据对象>
output:<新的聚类中心,数据对象>
通过 PSO 得到初始聚类中心链表 list.
For:(vector<string,kmeans_vector>cluster:list){
    dis tance=cluster.getV2() . dist(∀value);
    if (dis tance<min){
        min=distance;
        nearest_cluster = cluster. map(parseVector(_)). cache
    ();}}key_new=nearest_cluster
context. write(key_new,value)
End

算法 2:K-means 并行化 Reduce 过程算法
input:<聚类中心,数据对象>
output:<聚类中心,新的数据对象>
初始化一个集合来存储新的中心点
For;kmeans_Vector;value){
    计算 value 的均值,存储在临时的集合中;
}
将临时集合赋值给数据对象
中心点是否收敛
context. write(center,value_new)
End
```

4 实验验证与结果分析

4.1 实验环境

本次实验采用多台 PC 设备搭建 Spark 并行计算所需的集群环境. Spark 集群中的所有节点都给出相同的配置,操作系统均为 ubuntu14、1 个 CPU、4 GB 内存和 50 GB 硬盘. 集群环境中版本为 Hadoop—2. 6. 1,jdk1. 8 和 Spark1. 6. 1,一共拥有 1 个主节点和 4 个从节点.

4.2 实验设计

为了对比验证本文提出的算法和其他聚类算法

的在性能上的优越性,本次实验选取 UCI 机器学习部分数据集进行实验. 在 <http://archive.ics.uci.edu/ml/datasets.html> 可以找到这些数据集,数据特征如表 1 所示.

表 1 机器学习数据集

数据集	对象数目	属性值	标准聚类数
天平心理实验	625	4	3
淋巴检测	148	18	4
听力检测	226	69	9
乳腺癌检测	286	9	2

实验测试的算法包括标准的 K-means 和 PSOKmeans. 为了考查并行化 PSOK-means 算法在准确率和效率方面性能情况,设计如下的两组实验.

实验一:准确率对比分析. 采用本文的聚类算法与 UCI 数据集的标准聚类对比和误差平方和指标 (SSE)两者来评估本文所提算法的准确率,其中误差平方和指标如下:

$$SSE = \sum_{j=1}^K \sum_{d \in C_j} || \mu_j - x ||^2$$

(4)

式中, x 代表数据点; μ_j 代表类的中心.

实验二:主要对不同框架下聚类算法效率情况的不同进行分析.

4.3 实验结果分析

本文将采用 UCI 机器学习数据集中的天平心理实验 (Balance Scale)、淋巴检测 (Lymphography Data Set)、听力检测 (Audiology (Standardized)) 和乳腺癌检测数据集 (Breast Cancer),这几个数据集分别包含有 625,148,226 和 286 个对象数目. 在测试过程中需要将计算所需的所有输入数据都预先存储在 HDFS 中.

对于实验一,不同方法标准对比准确率定义如下式所示.

$$f_g = \frac{\sum_{m=1}^u a_m}{n} \times 100\%$$

(5)

式中, u 代表类别个数; a_m 是算法下第 m 个类别的对象数目;分母 n 代表总的对象数目. 通过医疗数据集对不同的算法进行单点实验,实验结果如表 2 和表 3 所示.

表 2 算法准确率

数据集	K-means/%	PSOK-means/%
天平心理实验	82.7	83.36
淋巴检测	77.6	79.5
听力检测	80.3	81.8
乳腺癌检测	92.4	94.3

表 3 SSE 指标

数据集	K-means	PSOK-means
心理实验	58 193	55 184
淋巴检测	201 752	190 049
听力检测	90 134	86 934
乳腺癌检测	558	537

实验一的结果分析:通过以上的表格可以知道,在相同的迭代次数情况下,本文提出的 PSOK-means 算法在求解 Balance Scale、Lymphography Data Set、Audiology (Standardized) 和 Breast Cancer 时,准确率明显优于 K-means 算法. 从表 1 可知,通过 UCI 的四种数据集测试,PSOK-means 算法的 SSE 指标值小于 K-means,另外其准确率明显优于 K-means,故本文提出的算法在医疗数据分析方面相比于 K-means 有明显的优势. 总体说来,PSOK-means 算法在求解结果的准确率上高于 K-means 算法.

在实验二中,采用前面所提及的天平心理实验 (Balance Scale) 作为实验数据,将其随机生成多组数据集 a 、 b 和 c . 在多组数据集下,分别测试 K-means 算法和 PSOK-means 算法在单机环境和 Spark 环境下的计算运行时间,进而计算加速比.

从图 4 和图 5 中可以看出,上述的三种算法在处理天平心理实验数据集 (Balance Scale) 时,随着节点数目的增加,加速比曲线趋向于平缓. 由于数据量不大, a 、 b 在拥有少量节点时,数据集并行处理. 紧接着,继续添加节点 S 的改变就不会很明显. 进而随着 Spark 集群运行的过程中,集群启动、任务转换调度等开销会导致计算性能的部分下滑. 在处理 a 数据集时,Spark 并行化的算法的 S 是基本符合线性增长的. 这显示在大数据量情况下,集群并行化能够有效地提高算法的效率. 由于实验条件,可以预见,在亚马逊云环境下,效果更加明显. 另外也由此可以看到在大数据集下,并行算法相比于单机效果更为明显.

为了进一步测试 Spark 平台的并行化算法对海量数据的加速比性能,本文采用 Car Evolution 数据集随机生成 1 000 个样本点,该数据共有 6 个属性,称为 d . 在 Spark 平台上,运行并行化的 K-means 和 PSOK-means,从而测出算法并行化解决海量数据问题的能力. 对于 D5 数据集的测试结果如图 6 所示.

通过图 6 可以看出,基于 Spark 并行化的

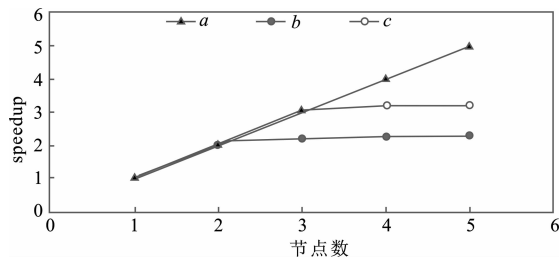


图 4 K-means 算法加速比

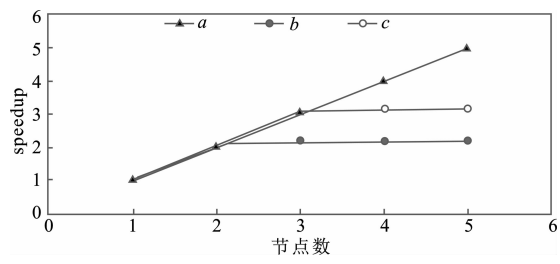


图 5 PSOK-means 算法加速比

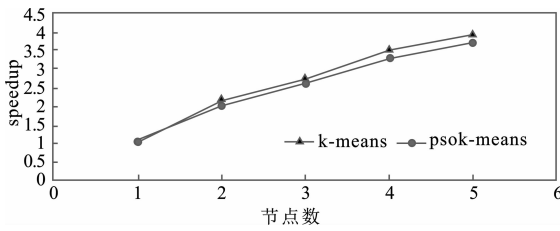


图 6 d 数据集下 Spark 并行化算法加速比

PSOK-means 算法在面对海量数据的处理时,随着节点数的增加,加速比基本线性增长,一直维持算法执行的高效性。

为了充分测试 Spark 平台并行化算法所带来的优势,将 PSOK-means 算法分别在 Spark 和 hadoop 平台下进行并行化算法对比实验. 实验结果如图 7 所示。

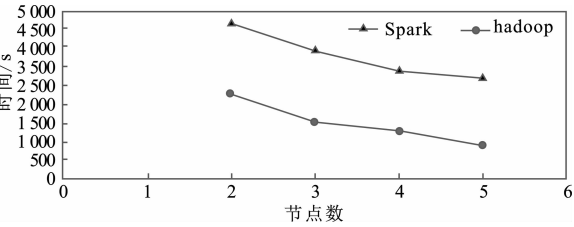


图 7 Spark 与 hadoop 性能对比

通过图 7 所示的实验结果可知,Spark 并行化的 PSOK-means 算法明显优于 hadoop,因为 Spark 使用 RDD 在迭代计算方面的优势,使得中间数据无需在磁盘上进行 I/O 操作,性能更为优越,在实际环境中效果更加明显。

5 结束语

当今时代,各行业的数据量从体积、速度及多样性都呈现飞速增长,传统算法已经无法充分挖掘大规模数据集特征,所以引入了 Spark 并行计算框架. 本文充分利用其在处理迭代计算和大数据计算方面的优势,提出了 Spark 并行化的 PSOK-means 算法. 该算法将粒子群 (PSO) 算法与 K-means 结合,利用改进的 PSO 来提高 K-means 的全局搜索能力,得到初始聚类中心,最后充分利用 Spark 的迭代计算能力,将 PSOK-means 算法并行化,提升该算法模型对海量数据的处理能力。

将该算法运用在乳腺癌检测、淋巴检测等医学研究中,与其他算法对比,本文的方法在准确率、运行效率方面都有更好的表现,在现实生产过程中具有一定的指导意义。

参考文献:

[1] Alsabti K, Ranka S, Singh V. An efficient K-Means clustering algorithm [J]. Electrical Engineering and Computer Science, 1997,23(11):1067-1080.

[2] Shboul B A, Myaeng S H. Initializing K-Means using Genetic Algorithms[C]//World Academy of Science, Engineering and Technology . IEEE,2009.

[3] Ahmadyfard A, Modares H. Combining PSO and k-means to enhance data clustering[C]// International Symposium on Telicommunications. IEEE,2008.

[4] Li J J , Yang X, Lu Y M, et al. Survey of particle swarm clustering algorithms[J]. Application Research of Computers, 2009 , 26 (12) :4423-4427.

[5] Liu J M, HAN Lichuan, HOU Liwen. Cluster analysis based on particle swarm optimizaiton algorithm[J]. System Engineering Theory and Practice, 2005,6:118-123.

[6] Pal S K, Wang P P. Genetic algorithm for pattern recognition[M]. CRC Press, 1996.

作者简介:

许明杰 男,(1992-),硕士研究生. 研究方向为云计算、机器学习、人工智能等. E-mail:1582677459@qq. com.

蔚承建 男,(1957-),博士,教授,硕士生导师. 研究方向为分布式计算、人工智能等.

沈 航 男,(1984-),博士,讲师. 研究方向为云计算、移动互联网、无线多媒体通信协议等.