基于 Hadoop 平台的 K-means 聚类算法①

刘宝龙, 苏 金

(西安工业大学 计算机科学与工程学院, 西安 710021)

摘 要: 传统的 K-means 算法虽然具有很多优点, 但聚类准则函数对簇密度不均的数据集分类效果较差. 文中在加权标准差准则函数的基础之上, 增加了收敛性判定, 并在 Hadoop 平台上提出了一种基于 MapReduce 编程思想设计与优化的 K-means 并行算法. 与传统的 K-means 算法相比, 设计的并行算法在聚类结果的准确性、加速比、扩展性、收敛性等方面都有显著的提高, 降低了因簇密度不均引起误分的概率, 提高了算法的聚类精度, 并且数据规模越大、节点越多, 优化的效果就越明显.

关键词: K-means; 簇密度; 聚类精度; MapReduce; Hadoop

K-Means Clustering Algorithm Based on Hadoop

LIU Bao-Long, SU Jin

(School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China)

Abstract: Although there are many advantages in traditional K-means algorithm, the clustering criterion function has poor efficiency on classification of the data set with uneven cluster density. On the basis of weighted standard deviation criterion function, this paper proposes a K-means parallel algorithm which is designed and optimized based on MapReduce programming. And it also increases the convergence judgment. Compared with the traditional K-means algorithm, the designed parallel algorithm has a significant improvement in the aspects of accuracy, speedup ratio, scalability and the convergence of clustering results. It also reduces the probability of misclassification caused by the uneven cluster density, and improves the clustering accuracy of the algorithm. What's more, the optimization effect will be more obvious when it deals with lager data size and more nodes.

Key words: K-means; cluster density; clustering accuracy; MapReduce; Hadoop

随着电商、物联网、云计算等一系列新型技术的发展与应用, 如今的数据增长已不再是线性的、缓慢的, 它所呈现的是海量的、复杂的、实时的与爆炸性的. 目前, 聚类挖掘技术已经越来越受到行业内的关注. 随着研究的深入, 聚类分析主要被划分为以下五类: 基于模型的聚类算法, 基于划分的聚类算法, 基于密度的聚类算法, 基于层次的聚类算法, 基于网格的聚类算法. 常用的 K-means 算法是一种基于划分的聚类挖掘算法, 该算法的思路简单、收敛速度快, 使用广泛且易于实现, 但其在 K 值及中心点的选取上仍然存在很大的随机性, 容易使聚类结果陷入局部最优, 对簇密度不均的数据集处理效果较差, 且并行处理能力差, 易产生内存泄漏, 缺乏可伸缩性.

为了进一步提高算法的运行效率, 许多研究人员提出了不同的解决方案. 文献[1]利用多次随机采样的方式来确定 K 值, 初步解决了聚类中心点个数选择问题; 文献[2]为了解决传统 K-means 算法在 Hadoop 平台上需要多次遍历数据集的问题, 提出了一种基于初始聚类中心点优化的选择算法 M+Kmeans, 可一次遍历所有数据集, 大幅度减少了原算法的遍历时间, 提高了算法的加速比; 文献[3]对 K-means 算法在迭代计算过程中容易产生的内存泄漏问题做了进一步的优化

① 基金资助:西安市未央区科技计划(201609);陕西省科技计划(2015KTCXSF-10-11)
收稿时间:2016-09-06;收到修改稿时间:2016-10-19 [doi:10.15888/j.cnki.csa.005779]

© 中国科学院软件研究所 http://www.c-s-a.org.cn

工作; 文献[4]基于 MapReduce 利用重组技术提出了一种新的聚类算法 Mrk-kmeans, 提高了原算法的运行效率和时间复杂度. 文献[5]将 Spark 框架和 Hadoop 技术进行了混合式设计, 实现了分布式 K-means 聚类算法, 提高算法的吞吐量和可扩展性; 文献[6]为了避免噪声和孤立点对 K-means 算法的影响, 提出了一种新的聚类有效性函数, 提高了数据聚类质量. 文献[7] Banerjee 和 Ghosh 等人提出了一种比例均衡的聚类算法研究, 提高了类簇的聚类效果. 文献[8]针对 K-means 算法易受到异常点干扰的缺点, 使得聚类精度及收敛速度都有大幅度的提高. 文献[9]在算法迭代过程中使用平均误差准则函数来达到较优的聚类效果, 保证了聚类运行结果的有效性和可靠性.

鉴于以上对 K-means 算法优化的优点, 利用文献[10]的加权标准差聚类准则函数原理, 在 Hadoop 云计算平台上提出了一种基于 MapReduce 分布式编程模型优化的 K-means 算法.

1 传统的K-means算法

传统的 K-means 算法是首先从 n 个数据对象中任意选择 k 个对象作为初始聚类中心, 对于剩余的其它对象, 根据它们与各个簇(类)中心的相似度, 分别将它们分配到相似度距离最近的那个中心点所在的簇内, 然后再重新计算每个新聚类的聚类中心,不断迭代这一过程直到准则函数收敛为止. K-means 算法具体描述如下所示:

1) 对于任意的数据集合 X, 事先设定聚类数目为 K, 并从 X 中随机选取任意的 K 个数据对象作为聚类的初始质心点 u_k, 其中 $u_1,u_2,\cdots,u_k \in U$.

2) 对于每一个数据点 $x_p \in X$ 且 $x_p \neq u_k$, 计算其到各个质心的欧式距离并将其分配到距离最小的那个类簇中. 欧式距离计算公式如下:

$$d(x_p,u_i)=\sqrt{\sum_{i=1}^{k}(x_p-u_i)^2} \tag{1}$$

3) 对于新得到每一个簇 U_k, 重新计算其中心点坐标 u_k .

4) 循环执行(2)、(3)步, 直到聚类准则函数:

$$J_{SSE}=\sum_{i=1}^{K}\sum_{x_p\in U_i}\arg\min|x_p-u_i|^2 \tag{2}$$

收敛为止. 其中: J_{SSE} 是所有数据对象的平方误差总和, K 代表类簇的个数, U_k 代表第 k 个簇, u_k 代表簇 U_k 的中心点, x_p 代表任意的数据对象.

由公式(2)可知, 该函数主要目的是尽量使簇内总的误差平方和达到最小, 从而来获得最优的聚类结果. 显然, 若 J_{SSE} 值越小, 说明误差越小, 那么聚类效果就会越好, 反之, 则效果就会越差. 由此可知, 其只适用于各类样本大体上为球形、簇密度均匀且样本数目之间悬殊不大的样本分布, 却不能有效处理簇密度不均且样本类型多的数据集, 经常导致许多大的类簇会被拆分, 从而影响聚类的质量. 因此, 针对以上存在的问题, 本文主要对 K-means 算法做了如下的优化.

2 优化的K-means算法

为了解决传统 K-means 算法聚类准则函数 J_{SSE} 的不足, 文献[10]提出了一种加权的标准差聚类准则函数, 具体如公式(3)所示. 其用标准差来反映一个数据集的离散程度, 标准差越高, 表示实验数据越离散, 也就是说越不精确, 反之, 标准差越低, 代表实验数据越精确.

$$\varepsilon=\sum_{i=1}^{K}\frac{m_i}{M}\sigma_i \quad \text{即} \quad \varepsilon=\sum_{i=1}^{K}\frac{m_i}{M}\sqrt{\frac{\sum_{j=1}^{m_i}\arg\min|x_{ij}-u_i|^2}{m_i}} \tag{3}$$

由公式(3)可得到如下结论: 为了使 ε 能够取得最小值, 对于任意的一个数据点 x, 在 K-means 算法迭代的过程中, 都应该选择一个使聚类准则函数 ε 的值增加最小的簇 C_i 加入. 也即是选择使得加权距离 $z_i\cdot\arg\min|x-u_i|$ 达到最小的簇加入(其中 z_i 为加权系数). 这样, 数据点在每一次迭代过程中, 就会更容易分配到数据点个数比较少的簇内. 当不同大小和密度的簇相邻而间距又比较小时, 处于稀疏的大簇边界上的数据被错分到与之相邻的高密度小簇中的可能性就会减少.

若进一步对加权的标准差聚类准则函数 ε 分析可知, 不难发现该函数值在迭代计算过程中理想状态下是一个单调下降的曲线. 具体如图 1 所示.

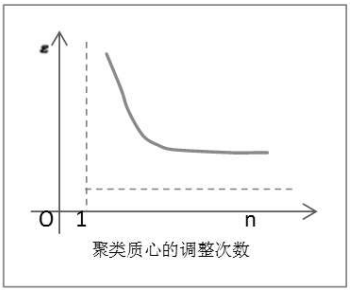

图 1 加权的聚类准则函数曲线图

© 中国科学院软件研究所 http://www.c-s-a.org.cn

这是因为对于任意数据对象 x，其在聚类迭代计算的过程中，数据对象都会被分配到离自己最近的簇中. 随着聚类质心的不断调整，数据对象都将会越来越向着有利于自己的簇进行靠近，ε 的值也将会慢慢趋近于一个固定的值，最终当 ε 的值不再发生变化时，整个算法达到最优聚类. 然而，传统的 K-means 算法在每次迭代计算过程中并未明确指出当算法收敛到何种程度时，可以结束整个算法的迭代计算过程，其也并未将准则函数的值 ε 作为算法是否结束的判定标志. 本文在公式(3)的基础之上，采用了最小加权距离来重新确定数据点应该被分配的类簇，并增加了 K-means 算法的收敛性判定. 因此，可将 K-means 算法进一步优化为:

(1) 给定数据集合 $X=\{x_1,x_2,\cdots,x_n\}$ 且 $x_n\in X$，并从 X 中随机选取任意的 K 个初始聚类质心点 $u_1,u_2,\cdots,u_k\in R^n$，并令 $q=1$.

(2) 逐个计算每个数据对象 x_p 与聚类质心的加权距离 $\mathrm{d}(x_p,u_i)=z_i\cdot\arg\min|x_p-u_i|$，其中 $i=1,2,\cdots,k$；如果满足以下式子 $d(x_p,u_i)=\min\{d(x_p,u_i)\}$，则将数据点 x_p 分配到中心点 u_i 所在的簇 U_i 中，即 $x_p\in U_i$. 这样，最终将会生成 K 个新的簇.

(3) 令 $q=q+1$，对新得到的每一个簇 U_i 重新计算其聚类质心:

$$u_i(q+1)=\frac{\sum_{p=1}^{n}x_p^{(U_i)}}{n}$$

其中 n 代表被分配到簇 U_i 中数据点的总个数.

(4) 计算加权的标准差聚类准则函数 ε 的值:

$$\varepsilon(q+1)=\sum_{i=1}^{K}\frac{m_i}{M}\sqrt{\frac{\sum_{j=1}^{m_i}\arg\min|x_{ij}-u_i(q+1)|^2}{m_i}}$$

若 $0\le\varepsilon(q)-\varepsilon(q+1)<\delta$ (其中 δ 为一个很小的常数)，则当前各簇内已没有数据对象被调整，结束本算法，已达到最优聚类；否则 $q=q+1$，返回循环执行第(2)步，直到聚类准则函数收敛为止.

3　K-means算法的设计与实现

由于 K-means 算法在分类时，各个类簇是可以相互独立的，本文在 Hadoop 平台上基于 MapReduce 设计 K-means 算法的主要思路是将串行 K-means 算法的每次迭代过程转化为一次能够独立执行的 MapReduce 计算，且多台计算机上的 MapReduce 计算任务可以协同工作. 每次 K-means 算法迭代主要分为以下三个阶段: Map 阶段, Combine 阶段和 Reduce 阶段. 具体如图 2 所示.

在 Map 阶段，首先随机选择 K 个中心点，并将这 K 个中心点存储在 HDFS 中，作为起始的全局变量. 利用最小加权距离对数据集中的数据点进行划分；在 Combine 阶段，由于在 Map 阶段每个数据节点输出的中间结果比较大，势必会造成多次溢写的发生，消耗大量的系统资源，为了进一步提高算法的运行效率，本算法在不改变最终计算结果的情况下，引进了 Combine()函数来优化 MapReduce 的中间结果，其将键值对中具有相同 key 值的 value 合并起来，可大幅度地减少溢写到磁盘的数据量. 另外，本阶段还把最后由 Combine()函数合并的结果作为 Reduce 阶段的输入，进一步减轻了 Reduce 阶段再次合并中间结果的负担，提高了原算法的执行速度；在 Reduce 阶段，首先读取出从每个 Combine 中处理的数据样本个数和每个数据样本各维的坐标值，并将对应的各维累加值分别对应相加，再除以总的样本个数，所得的结果即为新的中心点坐标. 然后把 Reduce 阶段输出的结果，作为新一轮迭代计算的中心点坐标并将其上传到 HDFS 中进行下一轮迭代，最后直到算法收敛为止.

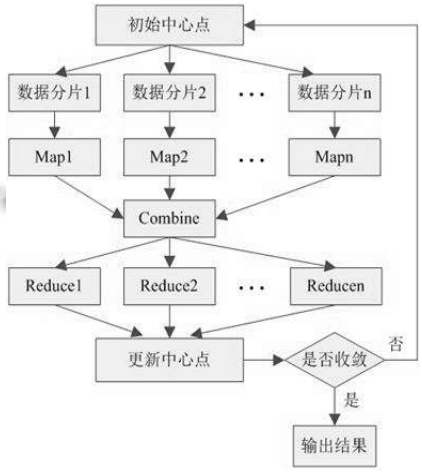

图 2　K-means 算法的并行化设计

4　实验结果及分析

4.1 实验环境

实验共选用六台计算机搭建 Hadoop 集群，其中一台作为 Jobtracker，硬件配置为: CPU 型号为 Intel(R)core(TM) p8400，2.26GHz，内存 8GB，硬盘 1TB，其它五台电脑作为 TastTracker，硬件配置为:

© 中国科学院软件研究所 http://www.c-s-a.org.cn

CPU 型号为 Pentium4 dual-Core 631, 内存 4GB, 硬盘 500GB. 六台计算机统一安装操作系统 Ubuntu 12.04. Hadoop 版本为 2.20, 代码编译采用 JDK1.8. 实验中的测试数据均来自于 UCI 数据集中的真实数据.

4.2 单机性能测试

本实验中主要从 UCI 数据集中随机选取了 6 组实验数据, 在单机节点下, 分别对本文算法和传统 K-means 算法进行了测试, 实验结果如下表 1 所示, 其中用 T_1 表示传统 K-means 算法的运行时间, 用 T_2 表示优化 K-means 算法的运行时间. 由表 1 可知, 在小数据规模下, 传统的 K-means 算法运行效率要高于优化的 K-means 算法, 这是由于在 Combine 和 Reduce 任务阶段, 需要不断的拉取数据, 会造成系统时间消耗过大. 但随着数据规模的不断增加, 传统的 K-means 算法在迭代计算过程中将会由于内存不足而抛出异常, 而优化的 K-means 算法却一直能够很好的进行大规模的数据处理, 初步体现了该算法在 Hadoop 平台上具有处理大规模数据的能力和优势.

表 1 两种算法单机性能测试对照表

实验次数	样本个数	T_1/s	T_2/s
1	5 万条	5.2	6.3
2	10 万条	8.3	9.1
3	20 万条	11.2	11.7
4	40 万条	18.6	12.5
5	80 万条	32.4	14.7
6	160 万条	内存溢出	16.2
7	320 万条	内存溢出	21.8

4.3 算法的准确性测试

为了测试算法的准确性, 从 UCI 数据集中随机挑选了三组实验数据, 其分别为: 数据集 1: Buzz in social media(100M, 2 维), 数据集 2: BlogFeedback (300M, 4 维), 数据集 3: Census-Income(KDD)(900M,8 维).

表 2 两种算法准确性测试对照表

UCI 数据集	Hadoop 平台算法	聚类准则函数收敛值	收敛时间(s)	准确率(%)
数据集 1	K-means	$J_{SSE}=98.5$	4.2	59.3
	优化的 K-means	$\varepsilon=72.1$	1.7	76.6
数据集 2	K-means	$J_{SSE}=69.6$	7.6	83.7
	优化的 K-means	$\varepsilon=60.5$	3.4	88.4
数据集 3	K-means	$J_{SSE}=48.2$	17.3	85.8
	优化的 K-means	$\varepsilon=37.6$	8.2	95.2

由表 2 统计的实验结果可知: 从总体上来看, 优化的 K-means 算法准确性明显高于传统的 K-means 算法, 这个容易理解, 当数据规模与维数成线性增加时, 这对于一个数据向量来讲, 其在相似性分组时被误分的概率也在逐渐增大, 而基于标准差聚类准则函数优化的 K-means 算法将数据对象分配给使加权距离最小的中心点所在的簇中, 明显减少了相对较大且稀疏的簇中数据对象被错误地分配到相邻的小而密集的可能性, 其既继承了传统 K-means 的优点, 又改善了聚类效果, 减少了迭代计算中质心被调整的次数, 缩短了原算法的收敛时间, 大幅度提高了聚类的精度且具有良好的稳定性.

4.4 集群加速比测试

为了测试本文算法在 Hadoop 集群环境下的加速比, 在 Covertype 数据集随机选取了 3 个样本数据, 其样本容量分别为: 100M, 200M, 400M. 从图 3 到图 5 可以看出, 数据规模越大, 优化的 K-means 算法加速比就越接近线性增长. 另外, 从实验结果也很容易看出, 无论数据规模与任务节点增加与否, 本文算法的加速比均优于传统的 K-means 算法, 这是因为优化的算法在 Map 阶段之后增加了 Combine 操作, 使得各个节点之间相互拉取数据的效率提高了, 尤其是当数据规模越大时, 这种效率提高的就越明显.

4.5 集群扩展性测试

为了测试算法的扩展性, 实验从 UCI 数据集中随机选取了三组测试数据, 其大小分别为: 100M, 300M, 900M. 从图 6 可知, 对于相同规模的数据集, 在任务节点逐渐增加的情况下, 大数据集的运行效率明显优于小数据集的运行效率, 尤其在第 3 个节点时, 运行 900M 数据时明显比运行 300M 的数据运行效率高. 由此可见, 随着任务节点不断增加, 本文算法可以显著提高系统对相同规模数据的处理能力, 具有良好的扩展性.

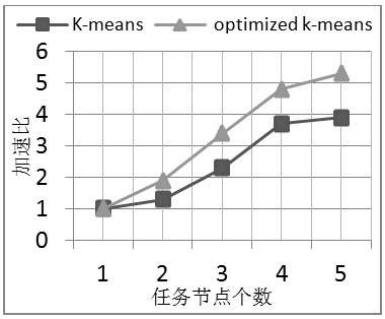

图 3 数据容量为 100M 时的加速比曲线图

© 中国科学院软件研究所 http://www.c-s-a.org.cn

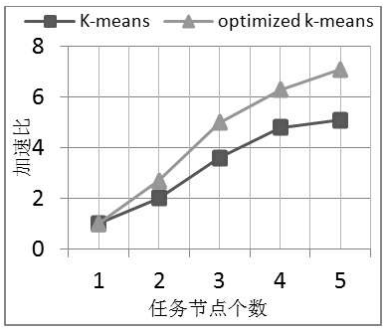

图 4　数据容量为 200M 时的加速比曲线

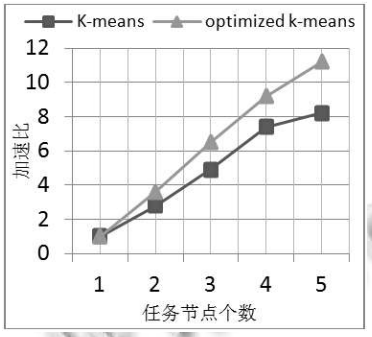

图 5　数据容量为 400M 时加速比曲线图

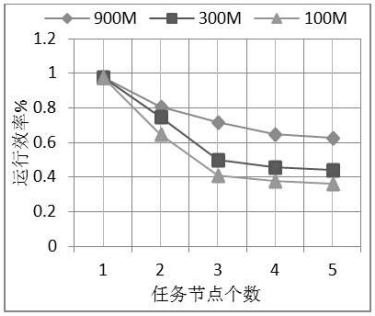

图 6　Hadoop 平台上的扩展率曲线图

5　结语

文中利用加权聚类准则函数的优点, 优化了传统 K-means 算法的聚类准则函数, 增加了算法收敛性约束判定, 并利用 MapReduce 分布式编程模型实现了 K-means 算法的并行化设计. 通过对算法的准确性、加速比、扩展性与收敛性等对比实验进一步验证了新算法的可行性, 与传统的 K-means 算法相比, 优化的聚类算法可更有效地应用于海量的数据挖掘中. 本文算法对于具有噪声或孤立点的数据分布, 及在 K 值的确定上, 效率还比较低, 需要进一步的研究.

参考文献

1 雷小锋,谢昆青,夏征义.一种基于 K-means 局部最优性的高效聚类算法.软件学报,2008,19(7):1683–1972.

2 武霞,董增寿,孟晓燕.基于大数据平台 Hadoop 的聚类算法 K 值优化研究.太原科技大学学报,2015,36(2):1673–2057.

3 虞倩倩,戴月明,李晶晶.基于 MapReduce 的 ACO-Kmeans 并行聚类算法.计算机工程与用,2013,49(6):117–120.

4 Shahrivari S, Jalili S. Single-pass and linear-time K-means clustering based on MapReduce, in: Information Systems, 2016, 60(3): 12–36.

5 唐振坤.基于 Spark 的机器学习平台设计与实现[硕士学位论文].厦门:厦门大学,2014.

6 孙秀娟,刘希玉.基于新聚类有效性函数的优化 K-means 算法.计算机应用,2008,12(28):1256 –1263.

7 Banerjee A, Ghosh J. On scaling up balanced clustering algorithms. Proc. of the 2nd SIAM ICDM, Arslington, VA. 2002. 333–349.

8 叶于林,夏秀渝,莫建华,刘帅.对 K-means 及势函数聚类算法的研究与优化.计算机系统应用,2015,24(4):124–135.

9 石云平.使用平均误差准则函数 E 的 K-means 算法分析.计算机与信息技术,2004,3(21):978–997.

10 张雪凤,张桂珍,刘鹏.基于聚类准则函数的优化 K-means 算法.计算机工程和应用,2011,47(11):165–172.

11 赵卫中,马慧芳,傅燕翔,史忠植.基于云计算平台的并行 K-means 聚类算法设计研究.计算机科学,2011,38(10):1065–1072.

12 周丽娟,王慧,王文伯.面向海量数据的并行 K-means 算法.华中科技大学学报(自然科学报),2012,(SI):150–152.

13 贾瑞玉,管玉勇,李亚龙.基于 MapReduce 模型的并行遗传 K-means 聚类算法.计算机工程与设计,2014,2 (2):31–35.

14 温程.并行聚类算法在 MapReduce 上的实现[硕士学位论文].杭州:浙江大学,2011.

15 Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Proc. of the 6th International Conference on Operation Systems Design & Implementation (OSDI). Berkeley, CA, USA. 2004. 137–150.

16 常晋义,何春霞.基于三角不等式原理的 K–means 加速算法.计算机工程与设计,2007,28(21):5094–5096.

17 Andrew WM. The anchors hierarchy: Using the triangle inequality to survive high dimensional data. Proc. of the 16th Conference on Uncertainty in Artificial Intelligence. 2000. 156–177.

18 江小平,李成华,向文,张新访,颜海涛.K-means 聚类算法的 MapReduce 并行化实现.华中科技大学报,2011,39(1):120–124.

© 中国科学院软件研究所 http://www.c-s-a.org.cn