自适应布谷鸟搜索的并行 K-means 聚类算法*

王 波, 余相君
(重庆大学 计算机学院, 重庆 400044)

摘 要: 针对 K-means 聚类算法受初始类中心影响,聚类结果容易陷入局部最优导致聚类准确率较低的问题,提出了一种基于自适应布谷鸟搜索的 K-means 聚类改进算法,并利用 MapReduce 编程模型实现了改进算法的并行化。通过搭建的 Hadoop 分布式计算平台对不同样本数据集分别进行 10 次准确性实验和效率实验,结果表明:a)聚类的平均准确率在实验所采用的四种 UCI 标准数据集上,相比原始 K-means 聚类算法和基于粒子群优化算法改进的 K-means 聚类算法都有所提高;b)聚类的平均运行效率在实验所采用的五种大小递增的随机数据集上,当数据量较大时,显著优于原始 K-means 串行算法,稍好于粒子群优化算法改进的并行 K-means 聚类算法。可以得出结论,在大数据情景下,应用该算法的聚类效果较好。

关键词: 聚类; K-均值算法; 布谷鸟搜索算法; Hadoop; MapReduce
中图分类号: TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2018)03-0675-05
doi:10.3969/j.issn.1001-3695.2018.03.008

Parallel K-means clustering algorithm based on adaptive cuckoo search

Wang Bo, Yu Xiangjun
(*College of Computer Science, Chongqing University, Chongqing* 400044, *China*)

Abstract: The original K-means clustering algorithm is seriously affected by initial centroids of clustering and easy to fall into local optima. So this paper proposed an improved K-means clustering based on adaptive cuckoo search, and achieved the parallelization of the improved algorithm using MapReduce programming model. It implemented accuracy experiments and efficiency experiments 10 times respectively on Hadoop platform for every different data sets, the experimental results show that:a) compared with the original K-means algorithm and PSO-Kmeans, the average accuracy of clustering improved in the experiments which test on four UCI standard data sets;b) tested the average execution efficiency of clustering in the experiments which test on five random incremental data sets, when the amount of data was very large, significantly better than original K-means algorithm, slightly better than PSO-Kmeans. It can be concluded that the algorithm can be applied to large data clustering, and will play a significant effect.

Key words: clustering; K-means algorithm; cuckoo search algorithm; Hadoop; MapReduce

0 引言

K-means 聚类算法是一种基于划分思想的聚类算法,具有思路简单、聚类速度快、局部搜索能力强等特点;但同时也因为其全局搜索能力弱、类中心初始化敏感,从而导致效率不足、准确率低等缺点[1]。很多学者针对 K-means 聚类算法的局限性展开研究与改进,为了得到质量较好的初始聚类中心,文献[2]采用谱图理论思想先通过相似性函数计算出样本的密度,然后再利用启发式规则动态生成初始聚类中心,但没有解决全局搜索能力差的问题;文献[3]假设每个类中都包含一个样本稠密区,然后基于最小生成树算法来初始化类中心,有效提高了 K-means 聚类算法的准确率,但降低了算法的效率;为了增强 K-means 聚类算法的全局搜索能力,文献[4]将改进的粒子群优化(particle swarm optimization,PSO)算法与 K-means 聚类算法相结合,并在运行过程中引入小概率随机变异操作来丰富种群的多样性;文献[5]为了解决 K-means 聚类算法易陷入局部收敛的问题,将遗传算法(genetic algorithm,GA)的编码、交叉和变异思想与 K-means 聚类的局部寻优能力相融合,提出基于遗传算法的 K-means 聚类算法,但这两种算法在数据量较大时效率都较低。文献[6]针对 K-means 聚类算法对初始聚类中心选择敏感而导致的聚类结果不稳定、聚类平均准确率低的问题,提出一种改进的粒子群优化的 K-means 聚类算法,并在 Hadoop 分布式框架上实现了算法的并行化处理,使算法的效率在数据量较大时有了显著的提升。

文献[7~10]经过多种测试实验,将布谷鸟搜索(cuckoo search,CS)算法与人工蜂群算法、萤火虫算法、粒子群算法等群体智能仿生算法进行比较,结果表明 CS 算法的性能均接近或优于其他经典的优化算法。布谷鸟搜索算法中的步长因子对算法的搜索精度有很大的影响[7],本文采用自适应步长调整策略来改进基本布谷鸟搜索算法,使其能够在局部搜索与全局搜索之间保持平衡,并利用 MapReduce 编程模型对改进后的自适应布谷鸟搜索(adaptive cuckoo search,ACS)算法进行并

收稿日期: 2016-10-27; **修回日期**: 2016-12-14 **基金项目**: 国家科技重大专项资助项目(2012ZX07-307-002)

作者简介:王波(1960-),男,辽宁丹东人,副教授,主要研究方向为智能建筑与智慧城市、物联网与网络安全(wangbo@cqu.edu.cn);余相君(1992-),男,硕士,主要研究方向为物联网与网络安全、数据挖掘算法并行化.

行化处理。

1 相关算法与技术

1.1 K-means 聚类算法

设样本数据集为 $X=\{x_i \mid i=1,2,\cdots,n\}$，$C_j(j=1,2,\cdots,k)$ 表示聚类的 k 个类别，$c_j(j=1,2,\cdots,k)$ 表示初始聚类中心。聚类 $C_1,C_2,\cdots,C_k$ 满足[1]：

a) $C_i \neq \varnothing, i=1,2,\cdots,k$。

b) $C_i \cap C_j = \varnothing, i,j=1,2,\cdots,k; i \neq j$。

c) $\sum_{i=1}^{k} C_i = \{x_1,x_2,\cdots,x_n\}$。

K-means 聚类算法步骤[1]如下：

a)随机选 k 个样本$(c_1,c_2,\cdots,c_k)$作为初始聚类中心。

b)将样本数据集$\{x_i\}$中各个样本按照最近距离原则分配给 k 个聚类中心 c_i。

c)计算新聚类中心 $c'_i(i=1,2,\cdots,k)$，即 $c'_i=\frac{1}{n}\sum_{x\in S_i} x$，其中 n 为 S_i 聚类域包含样本个数。

d)若 $c'_i \neq c_i(i=1,2,\cdots,k)$，转步骤 b)；否则算法收敛，结束。

K-means 聚类算法通过迭代的方法计算结果，为防止步骤 d)的终止条件不能满足而导致无限循环，算法通常会设置最大迭代次数。

1.2 自适应布谷鸟搜索算法

1.2.1 基本布谷鸟搜索算法

布谷鸟在自然界中通过随机或类似随机的方式搜索适合自己产卵的宿主鸟窝。为抽象出布谷鸟搜索宿主鸟窝的行为，需要设定三个假设条件[7]：

a)每只布谷鸟随机选取宿主鸟窝产卵，并且每次只产卵一枚。

b)所有布谷鸟选取的宿主鸟窝中存在一个最优鸟窝位置，将其保留到下一代。

c)所有布谷鸟每次选取宿主鸟窝总数固定为 n，每个宿主鸟窝主人发现一枚外来鸟蛋的概率固定为 $P_a \in [0,1]$。

在这三个假设条件基础上，布谷鸟搜索鸟窝位置的路径更新公式[7]为

$$x_i^{(t+1)} = x_i^t + \partial \oplus L(\lambda) \quad i=1,2,\cdots,n \tag{1}$$

其中：x_i^t 表示第 i 个鸟窝在第 t 代所处的位置；$\oplus$为点对点乘法；∂ 表示步长控制量，可通过求解问题设定；$L(\lambda)$ 表示服从参数$(1<\lambda\leqslant 3)$的 Lévy 分布产生的一个随机搜索向量，即 $L(\lambda) \sim u = t^{-\lambda}(1<\lambda\leqslant 3)$。鸟窝位置更新后，生成随机数 $r \in [0,1]$与发现概率 P_a 比较，当 $r<P_a$ 时，对 $x_i^{(t+1)}$ 位置随机改变，否则不变；最后保留最优的一组鸟窝位置。

1.2.2 自适应步长布谷鸟搜索算法

在基本布谷鸟搜索算法中，通过 Lévy 飞行随机产生的步长时长时短，步长越长则搜索精度越低，越容易搜索全局最优；步长越短则搜索精度越高，但会严重影响搜索速度[8,11,12]。所以基本布谷鸟搜索算法通过 Lévy 飞行产生步长虽然具有随机性，不容易陷入局部最优，但步长缺乏自适应性。为了能够动态自适应调整步长大小，必须按照不同搜索阶段的结果，不断在全局搜索速度与搜索精度之间寻求平衡。首先由文献[8]引入与最优鸟窝位置距离相关的自适应步长调整策略公式，如式(2)所示。

$$\text{step}_i = \text{step}_{\min} + (\text{step}_{\max} - \text{step}_{\min}) d_i \tag{2}$$

其中：$\text{step}_{\max}$表示步长的最大值；$\text{step}_{\min}$表示步长的最小值；d_i 表示步长调整因子，由文献[13]引入，如式(3)所示。

$$d_i = \frac{|\text{nest}_i - \text{nest}_{\text{best}}|}{d_{\max}} \tag{3}$$

其中：nest_i 表示第 i 个鸟窝位置；$\text{nest}_{\text{best}}$表示此时最优鸟窝位置；$d_{\max}$表示最优鸟窝与其余鸟窝最大距离。由此动态地根据上一次迭代结果更新本次搜索的步长，当某个非最优鸟窝离最优鸟窝位置越近时，步长移动越小；否则步长移动越大，从而较好地实现步长变化的自适应性。

1.3 Hadoop 分布式计算平台

Hadoop 使用成本较低的计算节点构成服务集群来并行地处理和分析数据，具有低成本、高可靠性、高扩展性、高效率等优点[14]。其理论基础起源于 Google 三大论文：

a)2003 年发表的 GFS(google file system)论文[15]，是一个用于访问大规模数据的大型分布式文件系统，运行于廉价的普通硬件上，文件被分割成很多块，使用冗余的方式储存于上千台机器上，可扩展性强，且提供容错功能。

b)2004 年发表的 MapReduce 论文[16]，描述了大数据情景下的分布式计算方式，主要思路是将计算任务分解到多台处理能力较弱的计算节点中并行处理，然后通过合并结果完成大数据处理，以其编写的执行程序分为 Map 和 Reduce 两个阶段，数据处理流程如图 1 所示。

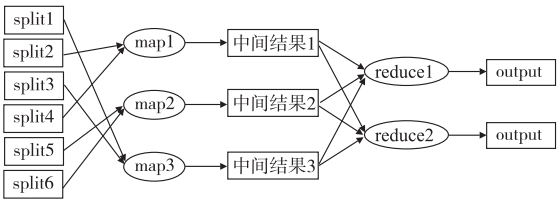

图1 MapReduce工作原理

c)2006 年发表的 Bigtable 论文[17]，是 Google 设计的一个稀疏、分布式、非关系型、持久化的键值(key-value)存储数据库。

2 基于 ACS 的 K-means 并行算法

2.1 编码方法和适应度函数

假设原始样本数据集需聚集成 k 类，每个样本包含 d 维特征，则算法的编码方法是选取 k 组向量坐标作为一个解。这里鸟巢的位置由 k 个聚类中心表示，样本向量的维数为 d，所以鸟巢的位置由 $k\times d$ 维的矩阵组成，编码方式如式(4)所示。

$$A=\begin{bmatrix} \text{nest}_{11} & \text{nest}_{12} & \cdots & \text{nest}_{1d} \\ \text{nest}_{21} & \text{nest}_{22} & \cdots & \text{nest}_{2d} \\ \vdots & \vdots & & \vdots \\ \text{nest}_{k1} & \text{nest}_{k2} & \cdots & \text{nest}_{kd} \end{bmatrix} \tag{4}$$

其中：$\text{nest}_{11},\text{nest}_{12},\cdots,\text{nest}_{1d}$表示第 1 个类的聚类中心 d 维向量；$\text{nest}_{21},\text{nest}_{22},\cdots,\text{nest}_{2d}$表示第 2 个类的聚类中心 d 维向量；$\text{nest}_{k1},\text{nest}_{k2},\cdots,\text{nest}_{kd}$表示第 k 个类的聚类中心 d 维向量。本文在 K-means 聚类过程中采用最近距离(欧氏距离)来确定样本所属类别，向量 x 与 y 之间距离公式为

$$D_{xy}=\sqrt{\sum_{i=1}^{d}(\text{nest}_{xi}-\text{nest}_{yi})^2} \tag{5}$$

其中:$1 \leqslant x \leqslant k, 1 \leqslant y \leqslant k$, $\text{nest}_{xi} - \text{nest}_{yi}$ 表示向量 x 与 y 在第 i 维的差值。

在群体仿生算法中,个体的优劣程度用适应度值大小来评价,适应度值越大表示个体越好,反之越差;群体进化的方向则由适应度函数来实现,适应度函数直接决定了仿生群体的进化行为,不同的适应度函数得到解的迭代次数、优劣程度等都不会相同。适应度函数常以类内距离或者聚类点数表示,但如果仅以类内距离或仅以点数表示适应度函数都会存在缺陷[18],影响解的迭代次数以及优劣度。所以,结合布谷鸟搜索过程以及 K-means 算法思想将聚类点数和类内距离融合构造以下适应度函数,如式(6)所示。

$$\text{Fit}(i) = \frac{CN_i}{DS_i} \quad i = 1, 2, \cdots, k \tag{6}$$

其中:CN_i 表示第 i 个聚类中样本数据点的个数;$DS_i = \sum_{x \in y} D_{xy}$ ($x = 1, 2, \cdots, CN_i; x \neq y$)表示第 i 个类中所有数据点到聚类中心 y 的距离之和。$\text{Fit}(i)$ 的倒数描述 CN_i 个样本聚集成某个类时各个点到类中心的平均距离。显然,$\text{Fit}(i)$ 的值越大,表示聚类效果越好。本文使用函数 $\text{Fit}(i)$ 作为 CS 算法的适应度函数。通过对 k 个类的适应度值求和,则可以用来表示本次聚类总体结果的好坏,结果度量 G 越大,则本次聚类结果越好,反之越差,如式(7)所示。

$$G = \sum_{i=1}^{k} \text{Fit}(i) \tag{7}$$

2.2　ACS-KMeans 算法的并行化实现步骤

基于布谷鸟搜索的 K-means 聚类算法可以很好地解决 K-means 算法受初始聚类中心影响、容易陷入局部最优等问题,例如文献[19]采用的 ACS-KMeans 算法在一定程度上弥补了 K-means 算法的不足,提高了算法的准确度,但其适应度函数只考虑了类内距离,效果仍可改进[18],并且其串行化的程序思想限制了算法对于大数据样本集的处理效率。本文提出一种将基于改进的自适应布谷鸟搜索算法与 K-means 聚类算法进行并行化融合处理的思想,其实现步骤如下:

a)输入待聚类样本数据集和待聚类个数 k、最大迭代次数 $\text{Iterator}_{\max}$、发现概率 P_a、最大步长 $\text{step}_{\max}$、最小步长 $\text{step}_{\min}$。

b)在样本数据集中随机初始化 k 个鸟巢的位置。

c)进行 K-means 聚类划分。由于每个点到最佳鸟窝的计算是相互独立的,使得可以利用 MapReduce 并行计算框架挖掘其并行度,以此大幅度缩短最佳簇中心(鸟巢)的计算时间,ACS-MapTask1 以数据点的行号 id + 簇中心 id 作为 key,以欧氏距离作为 value 输出,ACS-ReduceTask1 以 ACS-MapTask1 的输出作为输入,找出每个样本数据点所属最优鸟巢的位置。

d)根据式(6)计算每个鸟巢的适应度值,ACS-MapTask2 以鸟窝 id 作为 key,鸟窝适应度值作为 value 输出,ACS-ReduceTask2以 ACS-MapTask2 的输出作为输入,找出最佳鸟窝位置,由式(7)保留聚类结果度量 G。

e)保留上次迭代的最佳鸟窝,并根据式(2)(3)计算出步长控制量 step_i,然后对剩余的鸟窝进行更新,ACS-MapTask3 以鸟窝 id 作为 key,以 step_i 作为 value 输出,ACS-ReduceTask3 以 ACS-MapTask3 的输出作为输入,根据式(1)更新剩余鸟窝位置。

f)更新剩余鸟窝位置过程中,生成随机数 $r \in [0,1]$ 并与发现概率 P_a 进行对比,若 $r < P_a$,则抛弃该鸟巢,随机构建新鸟巢;否则,保留步骤 e)操作更新的鸟巢。

g)根据更新后的鸟巢重复步骤 c)操作,求出 k 个鸟巢新的适应度值,根据式(7)将更新后的聚类结果度量 G 与上代结果度量进行对比,若更好则取更新后的鸟巢组合,否则回滚不更新鸟巢。

h)如果已经达到最大的迭代次数 $\text{Iterator}_{\max}$,则输出结果,程序结束;否则重复步骤 e)操作。

算法流程如图 2 所示。

3　实验和结果

3.1　实验环境

本实验各类串行算法运行环境为一台 CentOS 7.0 操作系统的虚拟机,单核处理器,4 GB 内存,80 GB 硬盘。本实验各类并行算法运行环境为由四台虚拟机组成的 Hadoop 分布式集群,其中一台虚拟机机作为主节点,另外三台作为从节点;四台虚拟机都是单核处理器,1 GB 内存,20 GB 硬盘。每台虚拟机环境均为 CentOS 7.0、Hadoop 2.6.0、jdk 1.7.0。具体集群节点分布如表 1 所示。

表 1　集群分布详情

	Hadoop0	Hadoop1	Hadoop2	Hadoop3
NameNode	是	是	否	否
DataNode	否	是	是	是
JournalNode	是	是	是	否
ZooKeeper	是	是	是	否
ZKFC	是	是	否	否

3.2　算法准确性实验

为了验证基于本文聚类算法的准确性,选用四个已知所属类别的 UCI 标准数据集 4k2_far、Iris、Wine、leuk72_3k,分别进行 10 次实验取平均准确率。平均准确率计算公式如式(8)所示。

$$\bar{A} = \frac{\sum_{i=1}^{N} A_i}{N} \tag{8}$$

其中:N 表示实验次数,本文 N 取 10;A_i 表示每次独立实验的准确率,如式(9)所示;n 表示样本数量;flag_j 表示某个样本实验结果与标准数据集所属样本一致性标志,一致时为 1,否则为 0。

$$A_i = \frac{\sum_{j=1}^{n} \text{flag}_j}{n} \tag{9}$$

作为实验测试数据,四个 UCI 标准数据集的类数、样本数、维数信息如表 2 所示。

表 2　准确性实验数据集描述

数据集名称	类数	样本数	维数
4k2_far	4	400	2
Iris	3	150	4
Wine	3	178	13
leuk72_3k	3	72	39

分别采用串行的原始 K-means 聚类算法、并行的原始 K-means 聚类算法、基于改进的并行粒子群优化算法的 K-means 聚类算法(PSO-Kmeans)[6]、自适应串行布谷鸟搜索的 K-means 算法(ACS-Kmeans)[19]、本文所实现的基于改进的自适应布谷鸟搜索的并行化 K-means 算法。各类算法平均准确率如图 3 所示。

由图 3 可知,对于四组 UCI 标准数据集,原始的 K-means 算法的准确率始终最低,并且在实验过程中发现其在同一实验条件下多次实验的准确率差别较大,这是由于每次初始聚类中

心都是随机产生的,反映了 K-means 聚类算法受初始聚类中心影响较大。实验结果表明,本文算法的准确率高于原始 K-means 算法和文献[6]中改进的 PSO-Kmeans 算法,且与文献[19]中串行 ACS-K-means 算法相比,准确率也稍有提高。

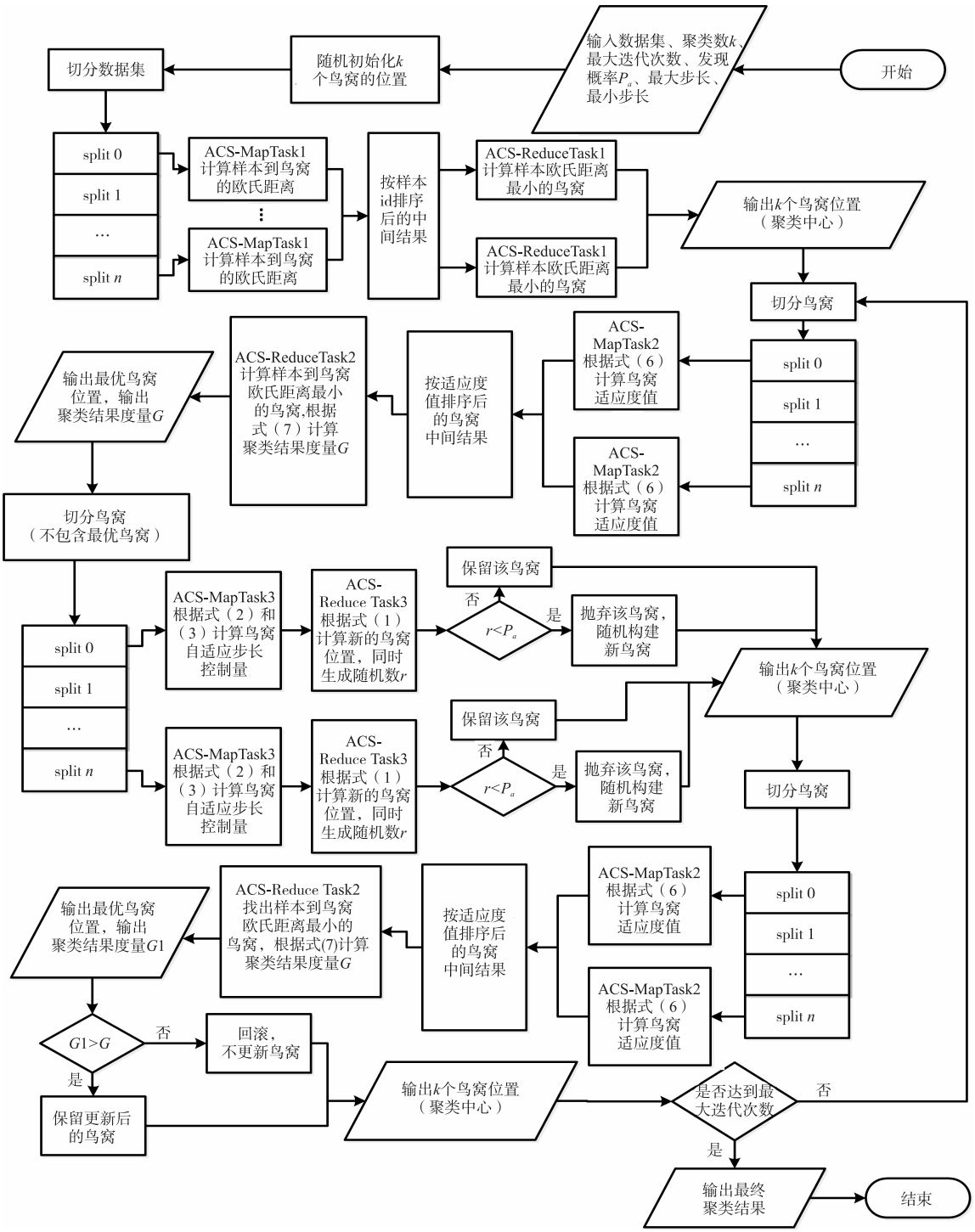

图2 算法流程

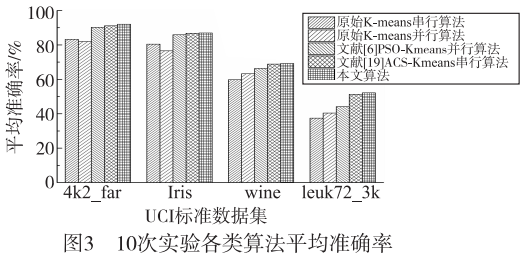

图3 10次实验各类算法平均准确率

3.3 算法效率实验

在算法的准确性实验基础上,为了验证各类算法在不同数据量情景下的效率,随机生成五个类数、维数相同,文件大小逐个增加的数据集,如表 3 所示。

表 3 效率实验数据集描述

数据集	文件大小/KB	类数	样本数	维数
A	1 270	5	10 万	6
B	63 477	5	500 万	6
C	634 766	5	5000 万	6
D	1 269 532	5	1 亿	6
E	6 347 657	5	5 亿	6

考虑到数据量较大时单次实验时间较长,同样分别采用上述准确性实验中的各类算法进行 10 次重复实验取平均时间,平均时间的计算公式如式(10)所示。

$$\overline{T}=\frac{\sum_{i=1}^{N}T_i}{N} \tag{10}$$

其中:N 表示实验次数,本文 N 取 10;T_i 表示每次独立实验所消耗的时间,为方便获取,本文直接以程序执行时间来表示。各类算法平均执行时间如表 4 所示。

表 4　10 次实验各类算法平均执行时间　　/s

数据集	原始 K-means 串行算法	原始 K-means 并行算法	文献[6] PSO-Kmeans 并行算法	文献[19] ACS-Kmeans 串行算法	本文算法
A	13.53	29.42	27.92	12.98	28.67
B	48.89	132.52	121.53	44.32	123.56
C	1 325.12	782.23	688.32	1 252.34	652.89
D	2 892.56	1 428.31	1 203.71	2 366.38	1 138.33
E	N/A	9 721.66	7 892.76	N/A	7 523.99

由表 4 可知,对于五组样本数递增的随机数据集,当样本数较少时,Hadoop 分布式平台的并行算法处理效率比单机串行算法效率低;但当样本数增多时,Hadoop 分布式平台的并行处理效率逐渐高于单机串行算法,特别是对于数据集 E,串行算法的处理效率过低,程序已不能在本次实验机器上正常执行(内存溢出),这是由于样本较少时,Hadoop 分布式平台需要不断地读写和传输数据占用较多时间,实际计算时间占比较小[20],而单机的串行算法不需要与其他机器交互所以效率更高;但当样本数量达到一定规模时,单机系统资源有限,所以串行算法执行时间很长,而 Hadoop 分布式平台并行处理由于可利用的资源更多所以效率表现更好。并且与文献[6]中提出的 PSO-Kmeans 并行算法相比,在数据量较大时本文算法执行耗时更少。

4　结束语

本文提出了一种基于自适应布谷鸟搜索的并行 K-means 聚类算法,解决了原始 K-means 聚类算法全局搜索能力差,以及在样本数据量较大时单机串行环境下效率低等问题。通过在 Hadoop 分布式计算平台上进行实验对比分析,结果表明相对于原始 K-means 算法和基于粒子群优化算法的 K-means 算法,本文改进算法的聚类准确性和大数据情景下的执行效率均有所提高。但本文研究也存在一些局限性,样本数据间仅考虑了欧氏距离作为 K-means 聚类算法的测度,未考虑集群节点数量对算法效率的影响,初始聚类中心采用随机获取的方式会影响算法的稳定性,算法还有优化的空间。总体来说,布谷鸟搜索算法作为一种新的元启发式群体智能与仿生算法,不仅可以运用于 K-means 聚类算法的改进,而且为如何从海量数据中快速准确地发现有用信息提供了一种新的研究思路。

参考文献:

[1] Han Jiawei, Kamber M, Pei Jian, *et al*. Data mining concepts and techniques [M]. 3rd ed. San Francisco: Morgan Kaumann, 2011: 451-456.

[2] 汪中,刘贵全,陈恩红. 一种优化初始中心点的 K-means 算法[J]. 模式识别与人工智能,2009,22(2):299-304.

[3] 李春生,王耀南. 聚类中心初始化的新方法[J]. 控制理论与应用,2010,27(10):1435-1440.

[4] 陶新民,徐晶,杨立标,等. 一种改进的粒子群和 K-均值混合聚类算法[J]. 电子与信息学报,2010,32(1):92-97.

[5] Lu Bin, Ju Fangyuan. An optimized genetic K-means clustering algorithm[C]//Proc of International Conference on Computer Science and Information Processing. Piscataway: IEEE Press, 2012:1296-1299.

[6] 马汉达,郝晓宇,马仁庆. 基于 Hadoop 的并行 PSO-Kmeans 算法实现 Web 日志挖掘[J]. 计算机科学, 2015,42(s1):470-473.

[7] Yang Xinshe, Deb S. Cuckoo search via Lévy flights[C]//Proc of World Congress on Nature & Biologically Inspired Computing. 2009: 210-214.

[8] 郑洪清,周永权. 一种自适应步长布谷鸟搜索算法[J]. 计算机工程与应用,2013,49(10):68-71.

[9] Raveendra. DE based job scheduling in grid enviroments[J]. Journal of Computer Networks, 2013, 1(2): 28-31.

[10] 陈乐,龙文. 求解工程结构优化问题的改进布谷鸟搜索算法[J]. 计算机应用研究,2014,31(3):679-683.

[11] Fister I, Yang Xinshe, Fister D, *et al*. Cuckoo search: a brief literature review[M]//Cuckoo Search and Firefly Algorithm, Volume 516 of the Series Studies in Computational Intelligence. Berlin: Springer-Verlag, 2013: 49-62.

[12] Yang Xinshe, Deb S. Cuckoo search: recent advances and applications [J]. Neural Computing and Applications, 2014, 24(1): 169-174.

[13] 欧阳喆,周永权. 自适应步长萤火虫优化算法[J]. 计算机应用,2011,31(7):1804-1807.

[14] White T. Hadoop: the definitive guide[M]. 4th ed. Sebastopol: O'Reilly Media, 2015: 3-15.

[15] Ghemawat S, Gobioff H, Leung S T. The Google file system[C]// Proc of the 19th ACM Symposium on Operating Systems Principles. 2003: 19-43.

[16] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[C]//Proc of Conference on Symposium on Opearting Systems Design & Implementation. 2004: 107-113.

[17] Chang F, Dean J, Ghemawat S, *et al*. Bigtable: a distributed storage system for structured data[C]//Proc of USENIX Symposium on Operating Systems Design and Implementation. [S. l.]: USENIX Association, 2006: 15.

[18] 喻金平,郑杰,梅宏标. 基于改进人工蜂群算法的 K-均值聚类算法[J]. 计算机应用,2014,34(4):1065-1069,1088.

[19] 杨辉华,王克,李灵巧,等. 基于自适应布谷鸟搜索算法的 K-means 聚类算法及其应用[J]. 计算机应用,2016,36(8):2066-2070.

[20] 周婷,张君瑛,罗成. 基于 Hadoop 的 K-means 聚类算法的实现[J]. 计算机技术与发展,2013,23(7):18-20.

(上接第 674 页)

[13] 刘芳. 基于 SOM 聚类的可视化方法及应用研究[J]. 计算机应用研究, 2012,29(4):1300-1303,1306.

[14] Gärtner T. A survey of kernrls for structured data[J]. ACM SIGKDD Explorations Newsletter, 2003,5(1):49-58.

[15] Hammer B, Micheli A, Sperduti A, *et al*. Recursive self-organizing network models[J]. Neural Networks, 2004,17(8):1061-1085.

[16] Tsutsumi K, Nakajima K. Maximum/minimum detection by a module-based neural network with redundant architecture[C] //Proc of International Joint Conference on Neural Networks. 1999: 558-561.

[17] Deng Zhidong, Mao Chengzhi, Chen Xiong. Deep self-organizing reservoir computing model for visual object recognition[C]//Proc of International Joint Conference on Neural Networks. 2016: 1325-1332.

[18] Qiu Lin, Xu Jungang. A Chinese word clustering method using latent dirichlet allocation and K-means [C]//Proc of the 2nd International Conference on Advances in Computer Science and Engineering. 2013: 267-270.

[19] Yan Danfeng, Hua Enzheng, Hu Bo. An improved single-pass algorithm for Chinese microblog topic detection and tracking [C]//Proc of IEEE International Congress on Big Data. 2016:251-258.

[20] 郑飞,张蕾. 基于分类的中文微博热点话题发现方法研究[C]// 第 29 次全国计算机安全学术交流会论文集. 2014: 311-314.