文章编号:1671-4598(2016)01-0252-03 DOI:10.16526/j.cnki.11—4762/tp.2016.01.070 中图分类号:TP312 文献标识码:A

ABC_Kmeans 聚类算法的 MapReduce 并行化研究

袁小艳
(四川文理学院 计算机学院,四川 达州 635000)

摘要:随着数据的海量增长,数据聚类算法的研究面临着海量数据挖掘和处理的挑战;针对 K-means 聚类算法对初始聚类中心的依赖性太强、全局搜索能力也差等缺点,将一种改进的人工蜂群算法与 K-means 算法相结合,提出了 ABC_Kmeans 聚类算法,以提高聚类的性能;为了提高聚类算法处理海量数据的能力,采用 MapReduce 模型对 ABC_Kmeans 进行并行化处理,分别设计了 Map、Combine 和 Reduce 函数;通过在多个海量数据集上进行实验,表明 ABC_Kmeans 算法的并行化设计具有良好的加速比和扩展性,适用于当今海量数据的挖掘和处理。

关键词:K-means;聚类;人工蜂群;MapReduce

Map Reduce Parallel Study of ABC_Kmeans Clustering Algorithm

Yuan Xiaoyan
(College of computer, Sichuan University of Arts and Science, Dazhou 635000, China)

Abstract: With the massive growth of data, data clustering algorithm research is facing the challenge of mass data mining and processing. For K-means clustering algorithm to the dependence of the initial clustering center is too strong, and poor global search ability shortcomings, will be an improved artificial colony algorithm combined with K-means algorithm, ABC_Kmeans clustering algorithm is proposed, in order to improve the performance of clustering. In order to improve the clustering algorithm's ability to deal with huge amounts of data, uses the MapReduce model for parallel processing to ABC_Kmeans, design the Map, Combine and Reduce function respectively, Through the experiments on several huge amounts of data collection, show ABC_Kmeans parallel design of algorithm has good speedup and scalability, applicable to today's huge amounts of data mining and processing.

Keywords: K-means; clustering; artificial bee colony; MapReduce

0 引言

聚类分析是当今数据挖掘研究的一个重要领域,其目的是把数据集按照规则分成若干个类别,使得同类别的数据尽量高内聚,不同类别的数据尽量低耦合,它是一种无监督学习技术[1]。

K_means 是常用的一种数据聚类算法,具有高效而简单的特性,但其 K 值要靠经验确定,结果也容易受初始中心点影响,易陷入局部最优解[2],全局搜索能力较差,鲁棒性也低。群体智能优化算法是一种把所有个体信息进行交互,从而得到最优结果的算法,其全局搜索能力较强,效率也较高,因此很多人都将其融入到 K_means 算法中进行研究,效果都较理想。

人工蜂群(ABC)算法是 2005 年根据蜂群觅食的行为提出的一种群体智能算法,其结构简单、收敛速度快、容易实现,更适合于计算机编程[3]。本文将一种改进的人工蜂群算法融入到 K_means 中进行迭代,降低了对初始聚类中心点的依赖,提高了全局寻优能力,但在当今海量数据的情况下,效率还是堪忧,因此将本文提出的 ABC_Kmeans 算法采用 Hadoop 平台的 MapReduce 模型进行并行化实现,因为 MapReduce 模型本身具有并行化结构,编程人员不需要考虑并行化的细节知识,实现起来也较容易,能够减少节点间的通信时间,这样不仅仅是提高时间效率,同时还可以防止早熟现象。

1 改进的 ABC 与 K_means 聚类算法的融合

在基本 ABC 算法中,蜜蜂有 3 种类型,即采蜜蜂、观望蜂和侦察蜂,采蜜蜂与蜜源相对应,负责对相应的蜜源采蜜,观望蜂通过采蜜蜂的共享蜜源信息选择最优蜜源,若蜜源被采蜜蜂和观望蜂放弃,则将该蜜源对应的采蜜蜂转化为侦察蜂,进而搜索新的蜜源。本文将改进的 ABC 算法与 K_means 迭代相结合,每次迭代都利用改进的 ABC 算法来优化聚类中心点,再利用 K_means 迭代更新每个聚类中心点,两种算法交替执行,直到聚类达到最优为止。下面介绍两种算法融合后的聚类过程。

(1) 设置参数,并输入样本数据集。采蜜蜂与观望蜂的个数,均为 SN,聚类数为 K,最大迭代次数为 LIN,控制参数为 $limit$。

(2) 聚类中心的初始化,本文的聚类中心就是蜜源。在基本 ABC 算法中,蜜源的初始化随机性太大,不能保证初始蜜源的均匀分布。本文采用混沌算子和反向学习算子对聚类中心进行初始化,从而保证初始聚类中心的多样性,以提高全局搜索的能力。

1) 首先生成具有 SN 个初始解的搜索空间。搜索空间是一个二维矩阵 $DF=K*d$,行数为 K,即聚类个数,列数为 d,即样本数据集的维数。搜索空间的值计算如下:

$$df[1,j]=rand(0,1)$$

收稿日期:2015-07-22; **修回日期**:2015-09-07。
基金项目:四川省教育厅一般项目(15ZB0318);四川文理学院一般项目(2014Z012Y);四川文理学院智能计算与物联网工程技术中心资助。

作者简介:袁小艳(1982-),女,重庆永川人,硕士,讲师,主要从事软件技术及开发、云教育、知识工程方向的研究。

$$df[s+1,j]=n\times df[s,j]\times(1-df[s,j])$$

其中 $j=1,2,3,\cdots\cdots,d$，$s=1,2,3,\cdots\cdots,K-1$，n 是一个调节因子。

2）计算每个初始解的混沌算子和反向学习算子，计算如下：

$$nc[i,j]=\lg[j]+df[s,j]\times(hg[j]-\lg[j])$$

$$rt[i,j]=\lg[j]+hg[j]-nc[i,j])$$

其中 $i=1,2,3,\cdots\cdots,K$，$hg\ [j]$、$lg\ [j]$ 是搜索空间第 j 维的上、下限，$nc\ [i,\ j]$ 为混沌算子，$rt\ [i,\ j]$ 为反向学习算子，混沌算子组成混沌集合 NC，反向学习算子组成反向集合 RT。

3）最终的初始聚类中心就是 NC 和 RT 两个集合中适应度最高的 SN 个最优解。本文各聚类中心的适应度采用欧式距离进行计算。

（3）采蜜蜂进行邻域搜索，采用贪婪机制选择适应度高的新聚类中心。基本 ABC 中，邻域搜索范围的随机性太大，导致全局搜索能力较好，但局部搜索能力欠缺。本文受 DE/best/1 的启发，邻域搜索方程改为由最优解引导的式子，具体如下：

$$p_{i,j}=df_{best,j}+random(-1,1)\times(df_{best,j}-df_{i,j})\quad(1)$$

其中 best 是适应度最优的采蜜蜂的 ID。

（4）采蜜蜂完成邻域搜索后，观望蜂根据轮盘赌机制选择跟随的聚类中心，其选择概率为：

$$F_i=\frac{dfit(i)}{\sum_{n=1}^{K}dfit(n)}$$

其中：$dfit(i)$ 是第 i 个解的适应度。

观望蜂选择聚类中心后，继续在聚类中心的邻域采用公式（1）进行搜索，查找适应度更高的聚类中心，以替换旧的聚类中心。

（5）若聚类中心经过连续 limit 次迭代后，没有被新的适应度高的新聚类中心替代，说明其陷入了局部最优，应该舍弃，该聚类中心对应的采蜜蜂也应该转换为侦察蜂，让其搜索新的聚类中心。基本 ABC 中，新蜜源的随机性太大，很容易被淘汰。受粒子群算法的启示，本文采用一种线性时变检索空间的策略选择新聚类中心，即随着迭代次数的增加而线性减小侦察蜂的检索空间，这样利于扩大探寻边界[3]，更利于迭代后期快速找到最优解，公式如下：

$$R_{i,j}=R_{\max}-\frac{t}{M_{\text{iter}}}(R_{\max}-R_{\min})$$

其中：$df'_{i,j}$ 是侦察蜂产生的新聚类中心，$df_{i,j}$ 是观望蜂产生的聚类中心，$R_{\max}$、$R_{\min}$ 是侦察蜂领域检索半径的最大、最小值，t 是当前迭代次数，M_{iter} 是最大迭代次数。

（6）对各聚类中心进行一次 K _ means 迭代，重新计算每个类别的聚类中心，并采用贪婪机制更新蜂群。

（7）将找到的最优聚类中心记录下来，若迭代次数小于 LIN，则转到（3）执行下一次迭代；否则将此结果作为最优聚类类别进行输出。

2　MapReduce 分布式模型

MapReduce 模型是一种并行计算模型，主要用于海量数据的并行化处理。该模型能对文件自动分块，能自动处理节点间的传输、节点失效[4]和负载均衡等优点，并有较好的扩展性和容错性[4]。MapReduce 模型输入的是一组键值对，输出的是另一组键值对，整个过程分成 Map（映射）和 Reduce（规约）。

Map 映射函数首先利用 Partion 过程将输入的键值对（key－value）分割成多个键值对，然后对这多个键值对进行处理，将处理的结果输出。Reduce 规约函数以 Map 函数的输出作为输入，并处理这些数据，输出一个较小的键值对集合。数据由 Map 函数到 Reduce 函数时，中间还要经过复制、排序、合并等操作，统称为 Shuffle 过程。合并是将具有相同键（key）的中间结果归并在一起，提供给 Reduce 处理。MapReduce 模型的整个处理过程如图 1 所示。

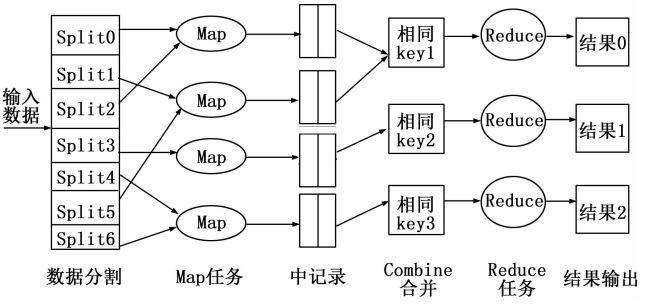

图 1　MapReduce 模型处理流程

3　ABC _ Kmeans 聚类算法的 MapReduce 并行化设计

ABC _ Kmeans 聚类算法的并行化采用 MapReduce 并行模型设计，输入的数据是以行的形式存储，让数据按行分片。ABC _ Kmeans 的每一次迭代都对应一次 MapReduce 过程，完成聚类中心适应度和观望蜂选择概率的计算，以得到新的聚类中心，具体设计如图 2 所示。

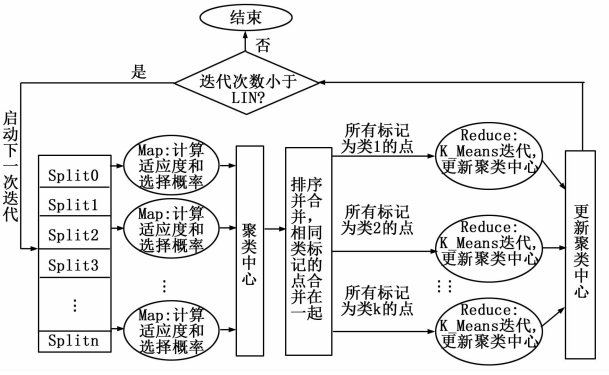

图 2　ABC _ Kmeans 的 MapReduce 并行化设计

3.1　Map 函数的设计

Map 函数的目标是计算每个蜜源的适应度，由此得到观望蜂的选择概率，并标记出观望蜂属于的新类别，其输入是样本数据或者上一次迭代的聚类中心，输入数据的形式为（行偏移量，记录行）[5] 这样的（key，value）键值对，其输出是（key’，value’），key’ 是选择的聚类中心的索引号，value’ 是当前样本点的坐标，其函数的伪代码如下：

```
void mapper(key,value)
{
从 value 中解析出样本数据,记为 foods
```

```
For iter=0 to k-1 do{
计算每个蜜源的适应度;
计算每个观望蜂的选择概率;
观望蜂选择聚类中心,并查找更优聚类中心;
侦察蜂查找更优聚类中心;
if(foods[maxoptidx]. opttimes>optmaxtime)
{
Sfoodinit(maxoptidx);
}
}
key'=maxoptidx;
value'=value;
Emit(key',value')
}
```

为了降低算法传输过程中的数据量和通信代价，ABC _ Kmeans 算法在 Map 操作后设计了一个 Combine 的函数，把每个 Map 函数的最终结果进行本地合并。

3.2 Combine 函数的设计

Combine 函数的输入（key'，value'）中，key' 是聚类中心的下标，value' 是分配给 key' 类别的样本数据的坐标值组成的字符串链表[6]；输出的（key，v）中，key 是聚类中心的下标，v 是包含了样本总数和 key 对应的各坐标值组成的字符串，其伪代码如下：

```
void Combiner(key',value')
{
Count=0;
While(value'. hasNext){
Point curret=value'. next();
Count+=current. getNum();
for(i=0;i<phydim;i++){
total[i]+=current. point[i];
}
}
key=key';
```

将前面得到的 count 和 total 数组各个分量的信息组成一个字符串，为 v；

```
Emit(key,v);
}
```

3.3 Reduce 函数的设计

Reduce 函数输入的（key，v）中，key 是聚类类别的下标，v 是从所有 Combine 函数传过来的结果，其作用是将 key 对应的样本点坐标值相加，再除以样本总数，得到新的聚类中心坐标。函数伪代码如下：

```
void Reducer(key,v)
{
While(v. hasNext){
Point current=v. next();
total+=current. getValue();
count=current. getCounter();
New=total/count;
}
}
```

根据 Reduce 的结果，得到新的聚类中心坐标，判断迭代次数是否小于 LIN ，若是的话就进入下一次迭代，否则算法收敛。

4 实验结果与分析

4.1 实验环境

本文中的实验是在我们的实验室搭建的云平台上运行的。平台由 12 台机器组成，其中一台是主控节点 master，一台是 JobTracker 服务节点，剩余的 10 台是 slaves，配置 DataNode 数据节点和 TaskTracker 服务节点。每台机器的配置为：CPU 是 2 核，内存是 8 GB，硬盘是 2TB，网卡为板载千兆以太网卡，操作系统是红帽 Linux AS 6.0，Hadoop 平台是 0.21.0 版本，JDK 是 1.6.0.29 版本。

4.2 单机处理实验结果分析

本实验是将 K-means 和 ABC _ Kmeans 聚类算法在相同的数据条件下，完成聚类的各自需要的迭代次数。处理的数据是分为 10 类且具有 30 000 个数据集的二维数据，两者数据处理的收敛速度如表 1 所示。

表 1 收敛速度的比较

编号	K-means	ABC_Kmeans
1	7	5
2	6	4
3	8	6
4	7	6
5	5	5
6	9	7
7	6	6
8	8	6

从表 1 中可以看出，相较于基本 K-means 聚类算法，本文的 ABC _ Kmeans 聚类算法具有更好的收敛速度，并且减小了聚类中心对初始值的依赖。

4.3 集群加速比性能实验结果分析

加速比是用来比较并行系统或者并行化程序的性能和扩展性的重要指标，是指相同的任务在单处理器上消耗的时间，与并行处理器系统中消耗的时间的比率。

实验用的数据集是通过程序生成的数据，生成了 1 G、2 G、4 G、8 G 等 4 个数据集，编号分别是 A、B、C、D，每个数据集由 30 维数据组成，要求生成 10 个聚类类别。

实验 1 将 1、2、4、8、10 个节点参与实验，观察节点在逐渐增加时，算法完成聚类任务的时间，如图 3 所示。从图中可以看到，随着节点的增加，A、B、C、D 这 4 个数据集运行的时间反而降低，同时相同规模的数据集处理时间也呈线性减少，这说明在 MapReduce 上执行 ABC _ Kmeans 算法的加速比较好，且其加速比随着数据规模的增大，性能也越来越好。

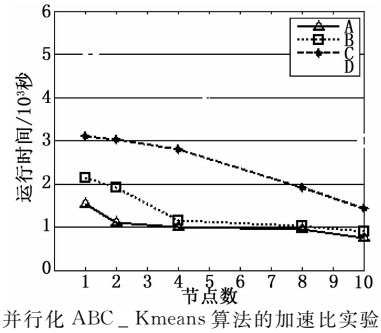

图 3 并行化 ABC _ Kmeans 算法的加速比实验结果图

（下转第 258 页）

表 3 三类接口测试实验结果

正确识别概率	K	HDSL	G. SHDSL
K	50	0	0
HDSL	0	55	5
G. SHDSL	0	5	35

由表 2 可以看出，利用本文提出的自动识别方法，基本上能够有效识别 K、HDSL 和 G. SHDSL 三类接口。其中 K 口识别正确率达到 100%，可靠性比较高。HDSL 口识别正确率达到 91%以上，误识别为 G. SHDSL 接口主要是因为接口阻抗不匹配以及线路衰减导致电平峰峰值的下降。G. SHDSL 口识别正确率达 87%以上，误识别为 HDSL 接口主要是因为测试过程中噪声干扰比较严重，以及传输速率设置错误导致频带宽度下降。HDSL 口和 G. SHDSL 口信号分别是 2B1Q 编码（四电平脉冲幅度调制码）和 16TC－PAM 编码（十六电平脉冲幅度调制码）。为提高 G. SHDSL 口和 HDSL 口识别的准确性，下步考虑使用参数均值归一化包络方差 R[7]，利用信号包络变化程度的不同来进一步区分，确保识别的正确性。

4 结论

本文介绍了一种有线远传接口自动识别方法，通过对接口特征进行深入分析，明确了采用限幅滤波法和基于 Burg 算法 AR 模型的谱估计方法，准确提取信号时域峰峰值和频域频带宽度的特征。通过采用双阀值判决方法，最终实现了对有线远传接口的有效识别。此方法基本上能够实现三类远传接口的识别，但是接口阻抗不匹配、线路衰减以及速率设置错误会导致识别率出现一定程度的下降，下一步还会有针对做进一步深入研究。该方法可以同样用于其它类型接口信号的特征提取与分类识别，为下步有线远传接口的智能测试提供了基础和前提，可以有效提高有线远距离传输接口的自动化测试水平[8]，在推广和应用上具有一定的实际价值。

参考文献：

[1] 邵怀宗，袁祥荆，吴 颖，等. 小型化通信电台综合测试系统的设计研究 [J]. 兵工学报，2009，30 (10)：1389－1395.

[2] 梁 艳，梁昔明，廖力清. 模拟信号调制方式自动识别仿真 [J]. 计算机测量与控制，2006. 14 (1)：117－119，127.

[3] 韦静涵. 接口检测及时域波形分析软件设计与实现 [D]. 成都：电子科技大学，2013.

[4] 杜挺克，杨俊峰，宋克柱，等. 一种计算数据相关性抖动峰峰值的方法 [J]. 中国科学技术大学学报，2009，39 (6)：608－611.

[5] 罗 丰，段沛沛，吴顺君. 基于 Burg 算法的短序列谱估计研究 [J]. 西安电子科技大学学报：自然科学版，2005，32 (5)：724－728.

[6] 黄春琳，邱 玲，沈振康. 数字调制信号的神经网络识别方法 [J]. 国防科技大学学报，1999，21 (2)：58－61.

[7] AAAAM R M A. Division － of － amplitude photopolarimeter (DOAP) for the simultaneous measurement of all four Stokes parameters of light [J]. Optica Acta，1982，29 (5)：685－689.

(上接第 254 页)

实验 2 分别选择 2、4、6、8 个节点，对应采用 A、B、C、D 这 4 个数据集计算，实验结果如图 4 所示。在图中可以看到，当节点数与数据集的规模同比例增长时，MapReduce 处理数据的水平也基本上保持一致，这说明了其有良好的扩展性。

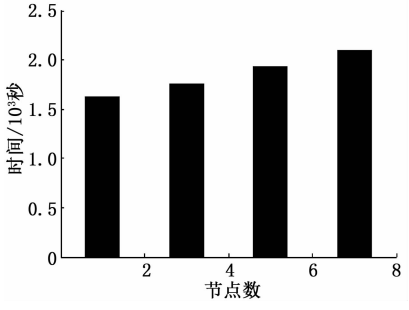

图 4 并行化 ABC _ Kmeans 算法的扩展性实验结果图

5 结束语

鉴于基本 K－means 聚类算法过于依赖初始聚类中心，全局搜索能力较差的原因，本文将全局搜索能力较好的人工蜂群算法（ABC）融入其中，并改进了人工蜂群算法的初始蜜源，让 ABC _ Kmeans 不再依赖于初始中心。为了提高 ABC _ Kmeans 算法的执行效率，本文将其利用 MapReduce 并行化模型实现。实验表明 ABC _ Kmeans 并行聚类算法比一般的 K－means 聚类算法，具有更好的聚类性能和更高的时间效率，数据量越大，时间效率的优势越好。随着云计算的兴起，数据挖掘越演越烈，其中的聚类算法研究也越来越热烈，本文的研究仅仅起到抛砖引玉的作用。

参考文献：

[1] 曹永春，蔡正琦，邵亚斌. 基于 K _ means 的改进人工蜂群聚类算法 [J]. 计算机应用，2014，34 (1)：204－208.

[2] 管玉勇. K－means 算法与智能算法融合的研究 [D]. 合肥：安徽大学，2014.

[3] 李海生. 蜂群算法及其在垂直 Web 检索中的应用 [D]. 广州：广州大学，2010.

[4] 杨国营. 基于 MapReduce 模型文本分类算法的研究 [D]. 沈阳：辽宁大学，2013

[5] 虞倩倩，戴月明，李晶晶. 基于 MapReduce 的 ACO－K－means 并行聚类算法 [J]. 计算机工程与应用，2013，49 (16)：117－121.

[6] 赵卫中，马慧芳，傅燕翔，等. 基于云计算平台 Hadoop 的并行 k－means 聚类算法设计研究 [J] . 计算机科学，2011，38 (10)：166－169

[7] 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

[8] 张石磊，武 装. 一种基于 Hadoop 云计算平台的聚类算法优化的研究 [J]. 计算机科学，2012，39 (10)：115－118

[9] 莫 赞，罗世雄，杨清平，等. 基于 K－means 算法的改进蚁群聚类算法及其应用 [J]. 系统科学学报，2012，20 (3)：91－94

[10] 江小平，李成华，向 文，等. k－means 聚类算法的 MapReduce 并行化实现 [J]. 华中科技大学学报，2011，39 (6)：120－124