第44卷第7期 中国科学技术大学学报 Vol. 44, No. 7
2014年7月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Jul. 2014

文章编号:0253-2778(2014)07-0537-07

基于 MapReduce 的基因数据密度层次聚类算法

涂金金,杨 明,郭丽娜

(南京师范大学计算机科学与技术学院,江苏南京 210046)

摘要:随着生物信息技术的快速发展,基因表达数据的规模急剧增长,这给传统的基因表达数据聚类算法带来了严峻的挑战.基于密度的层次聚类(DHC)能够较好地解决基因表达数据嵌套类问题且鲁棒性较好,但处理海量数据的效率不高.为此,提出了基于 MapReduce 的密度层次聚类算法——DisDHC.该算法首先进行数据分割,在每个子集上利用 DHC 进行聚类获得稀疏化的数据;在此基础上再次进行 DHC 聚类;最终产生整体数据的密度中心点.在酵母数据集、酵母细胞周期数据集、人血清数据集上进行实验,结果表明,DisDHC 算法在保持 DHC 聚类效果的同时,极大地缩短了聚类时间.

关键词:MapReduce;密度层次聚类;基因表达数据

中图分类号:TP311 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2014.07.001

引用格式:Tu Jinjin, Yang Ming, Guo Lina. A density-based hierarchical clustering algorithm of gene data based on MapReduce[J]. Journal of University of Science and Technology of China, 2014,44(7):537-543.
涂金金,杨明,郭丽娜. 基于 MapReduce 的基因数据密度层次聚类算法[J]. 中国科学技术大学学报,2014,44(7):537-543.

A density-based hierarchical clustering algorithm of gene data based on MapReduce

TU Jinjin, YANG Ming, GUO Lina

(*School of Computer Science and Technology, Nanjing Normal University, Nanjing* 210046, *China*)

Abstract: The amount of gene expression data scale is increasing sharply with the rapid development of bioinformatics technology, which poses a serious challenge for traditional clustering algorithms. Density-based hierarchical clustering (DHC) can solve the problem of the nested class of gene expression data and has good robustness, but for handling huge amounts of data. Therefore, a density-based hierarchical clustering algorithm on MapReduce(DisDHC) was proposed. It partitioned data sets into smaller blocks, clustered each block using DHC in parallel, gathered the result for re-clustering, and produced all density centers of each cluster. The experiments on GAL dataset, Cell cycle dataset, and Serum dataset show that DisDHC reduces clustering time and achieves high performance.

Key words: MapReduce; density-based hierarchical clustering; gene expression data

收稿日期:2014-03-21;**修回日期**:2014-06-15

基金项目:国家自然科学基金(61272222,61003116),江苏省自然科学基金重点重大专项(BK2011005),江苏省自然科学基金(BK2011782),江苏省普通高校研究生科研创新计划项目(CXLX12_0415)资助.

作者简介:涂金金,男,1988年生,硕士生. 研究方向:机器学习、模式识别. E-mail: tujinjin1988@sina.com

通讯作者:杨明,男,博士/教授. E-mail: myang@njnu.edu.cn

0 引言

基因数据量巨大,功能繁多,因此聚类分析成为目前处理基因表达数据的有效技术之一[1].它具有分析未知基因潜在功能和补全基因功能注释的作用.

目前,针对基因表达数据的聚类也有了一些研究成果.如 Eisen 等[2]提出了一种层次聚类算法:循环扫描数据集,找出最相似的两个基因并将其合并成一个新的基因,最终生成一棵层次聚类树.Jiang 等[3]提出了一种基于密度的层次聚类算法.该算法给出了一种新的密度定义,并且利用这种定义计算出每个基因间的吸引程度,构建吸引树,最终对生成的吸引树进行划分以达到聚类的目标.这种算法能够较好地解决基因表达数据嵌套类问题,并且具有良好的鲁棒性.这些算法存在一个普遍的问题:对于海量数据,处理效率不高.

一些研究者提出将传统经典的聚类算法应用到基因表达数据上,如 k-means[4]、普聚类[5]、近邻传播[6]、CAST[7]以及自组织映射(SOMs)[8]等.并且这些算法已经存在并行方式,如 Zhao 等[9]提出的并行 k-means 聚类算法,Chen 等[10]提出的并行谱聚类以及 Lu 等[11]提出的基于 MapReduce[12]的近邻传播聚类算法.这些算法相对于传统算法在效率上有极大地提高,然而,基因表达数据有其自身的特点:多噪声和结构复杂,存在嵌套类,因此,将上述并行算法运用到基因表达数据上并不能取得良好的效果.

本文针对基因表达数据自身的特点以及聚类的效率问题,结合 DHC[3]算法,提出了一种基于 MapReduce 的密度层次聚类算法(DisDHC).实验结果表明,相较已有算法,DisDHC 算法能在保持 DHC 聚类效果的同时,极地大缩短聚类时间.

1 相关工作

1.1 MapReduce 简介

MapReduce 是一种比较流行的并行框架,以"分而治之"的思想来处理大规模数据.这一思想由 map 和 reduce 两个函数来实现.map 负责将任务分解成多个子任务,以键值对形式存储记录;reduce 负责把多个中间结果汇总在一起,进 一步处理获得最终结果.在上述过程中, MapReduce 框架处理了各种复杂问 题,如工作调度、容错处理、网络通信等,具体处理过程如图 1[13]所示.

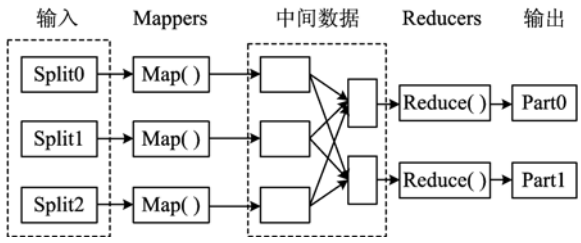

图 1 MapReduce 数据处理流程图

Fig. 1 Schematic diagram of Data processing on MapReduce

1.2 基因聚类数据简介

基因表达数据通常是经过一些实验测量技术获得的,主要有四种技术: cDNA (complementary DNA) microarray 技术,即 cDNA 微阵列技术[14]、基因芯片技术[15]、SAGE,即序列分析基因表达技术[16]和实时 PCR 技术[17].

图 2 展示了酵母细胞周期数据集基因表达数据的片段.该数据集最初是由 Cho 等[18]经实验测得的,后来 Yeung 等[19]对其进行精心挑选最终获得了酵母细胞周期数据集.图中每一行(除了第一行)代表一条基因,每一列代表水平(不同的发展阶段、组织等).因此,其中某一个数据的意思就是某一条基因在某个指定水平中的表达强度.

Main	Gp	c1	c2	c3	c4	……
YDL179w	1	5.22036	5.03044	4.89784	5.24175	……
YLR079w	1	5.86363	5.68698	5.87212	5.73010	……
YER111c	1	4.46591	4.74493	5.20949	4.64439	……
YBR200w	1	5.55683	5.69036	5.18178	5.03695	……
……						

图 2 酵母细胞周期数据集示例

Fig. 2 The diagram of the yeast cell cycle dataset

1.3 DHC 聚类算法

DHC 聚类算法主要是通过定义每一个基因的密度,构建吸引树,并以一定方式对其进行划分生成密度树,获得最终的聚类.该算法中基因与基因之间的距离定义如下:

如果 $S(O_i,O_j)>0$,则 $d(O_i,O_j)=1/S(O_i,O_j)$;反之,$d(O_i,O_j)=+\infty$.其中,$S(O_i,O_j)$的具体描述如下:

$$S(O_i,O_j)=\frac{\sum_{l=1}^{d}(O_{il}-\overline{O_i})(O_{jl}-\overline{O_j})}{\sqrt{\sum_{l=1}^{d}(O_{il}-\overline{O_i})^2\sum_{l=1}^{d}(O_{jl}-\overline{O_j})^2}} \quad (1)$$

$$\overline{O_i} = \frac{\sum_{l=1}^{d} O_{il}}{d} \tag{2}$$

$$\overline{O_j} = \frac{\sum_{l=1}^{d} O_{jl}}{d} \tag{3}$$

算法中，假设给定一个半径 r，定义每一个 d 维基因的超球体积为

$$\mathrm{Vol}(S_r^{O_i}) = r^d \tag{4}$$

将超球分割为一系列的超壳，使得每一个超壳的体积为一个单位，具体情况见图 3[3]，则第 k 个超壳的半径为式(5).

$$r_k = k^{\frac{1}{d}} \tag{5}$$

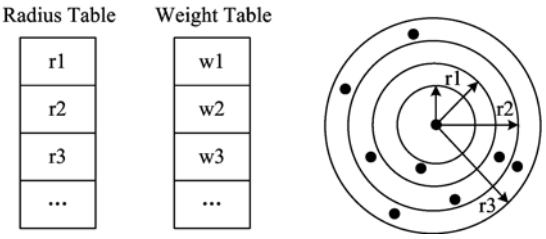

图 3　一条基因的超球和超壳示意图

Fig. 3　Hypersphere and hyper-shell of a gene

每一个基因的密度定义为每一个超壳中落入的基因数目乘以相应权重的和，即

$$\left.\begin{aligned} &\mathrm{density}(O) = \sum_{k=1}^{\infty} \mathrm{Weight}(\mathrm{Shell}_k^O)\, \mathrm{N}_k \\ &\text{s. t.} \quad \mathrm{Weight}(\mathrm{Shell}_k^{O_i}) = \frac{1}{\log\, k}, \\ &\qquad \mathrm{Shell}_k^{O_i} = \mathrm{Shell}_{k+1}^{O_i} - \mathrm{Shell}_k^{O_i} \\ &\qquad\quad = \{O_j \mid r_k < d(O_i, O_j) \leqslant r_{k+1}\}, \\ &\qquad \mathrm{N}_k = \| \mathrm{Shell}_k^{O_i} \| \end{aligned}\right\} \tag{6}$$

根据密度的定义，构建吸引树，其中，每一个基因的吸引者定义为：

$$\left.\begin{aligned} &\mathrm{Attractor}(O) = \underset{O_j \in A(O)}{\operatorname{argmax}}\, \mathrm{Attraction}(O_j, O) \\ &\text{s. t.} \\ &\mathrm{Attraction}(O_i, O_j) = \frac{\mathrm{density}(O_i).\, \mathrm{density}(O_j)}{d(O_i, O_j)^{k-1}} \end{aligned}\right\} \tag{7}$$

其中，O_j 表示密度大于基因 O 的所有基因.

遍历吸引树，对于任意一条边 O_iO_j，如果它们之间的相似度 $S(O_i, O_j) < T$，并且落在基因 O_j 在 T 邻域内基因数量不小于 MinPts(T，MinPts 是指定的参数)，则对 O_iO_j 进行划分. 根据这一规则，循环划分吸引树，直至形成密度树，获得最终的聚类结果.

2　基于 MapReduce 的密度层次聚类算法

2.1　基于数据稀疏化的并行密度层次聚类算法

由于基因数据量大，功能繁多，因此本文基于效率的考虑，首先提出了数据稀疏化的并行密度层次聚类算法；然后运用 MapReduce 框架实现算法. 该算法获得良好效果基于以下的假设[11]：

假设 2.1　数据集是稠密的且能够应用聚类算法聚类.

假设 2.2　在数据规模很大时，如果可以获得充分多的密度中心点(即聚类代表点)，并且他们可以代表原数据的分布，则利用 DHC 算法对这些密度中心点进行聚类，获得的新的密度中心点仍然可以代表原数据的分布.

根据以上的假设，基于数据稀疏化的密度层次聚类算法具体描述如下：

算法 2.1　稀疏化的密度层次聚类算法

输入：基因表达数据集-G，数据集划分的块数-N；

输出：聚类结果(聚类号，基因)-(L, g).

步骤如下：

Step 1　将 G 随机划分为 N 个子集，即

$$G = \bigcup_{i=1}^{N} S_i,\ S_i \cap S_j = \Phi;$$

Step 2　对所有的划分 S_i，$i = 1, 2, \cdots, N$，并行的进行 DHC 聚类，得到密度中心点集合 $M = \bigcup_{i=1}^{N} \mathrm{DHC}(S_i)$，这里的 $\mathrm{DHC}(S_i)$ 代表以 S_i 为输入，DHC 聚类得到的密度中心点. 这一过程中记录每一个基因属于具体密度中心点的标号，记为 index；

Step 3　以 M 为输入，再次进行 DHC 聚类，得到最终整个数据集的密度中心点 LM；

Step 4　将已标记密度中心点的数据集作为输入，与 LM 中的聚类标记进行比较，将基因划入具有相同 index 的密度中心，作为一类，最终结果以(聚类号，基因)-(L, g)形式输出.

2.2　基于 MapReduce 的密度层次聚类算法 DisDHC

在数据稀疏化的密度层次聚类算法基础上，我们运用 MapReduce 框架将其实现. DisDHC 主要包含 3 个过程：第一个过程 map 负责将数据集随机分为 N 块，reduce 负责在每个子块上进行 DHC 聚类，形成候选密度中心点，并且生成带标记的数据集. 第

二个过程主要是对候选密度中心点进行二次DHC聚类,形成最终的整体密度中心点.第三个过程中,map负责对带标记的数据集分割,并且对比整体密度中心点中的标记,将基因划分进相应的聚类,reduce负责对各子块的聚类进行汇总,形成最终聚类.

上述三个过程的具体算法描述如下:

算法 2.2 DisDHC map1

输入:数据集G文件路径、划分块数N;

输出:键值对<key,value>、key为块号、value为基因.

主要步骤如下:

Step 1 有随机数Math.random()函数生成块号number∈[1,N];

Step 2 key=number,value=g,g为一条基因,以键值对<key,value>形式将结果输出至reduce中.

算法 2.3 DisDHC reduce1

输入:<key, valueList>, key代表数据块编号,ValueList代表具有相同key的基因列表;

输出:候选密度中心点、已添加标号的数据集.

主要步骤如下:

Step 1 利用DHC算法对ValueList进行聚类;

Step 2 抽取每个聚类中子树的根节点作为密度中心点,结合本节点的ID以及其所在的类号组合在一起作为一条记录输出,即候选密度中心点,记为candidate;并且在每一条基因g的前面添加ID和聚类号,组合成新的基因$\overline{g}$;

Step 3 数据候选密度中心点和已添加标号基因数据集至HDFS上.

算法 2.4 DisDHC map2

输入:各个节点候选密度中心点;

输出:整体选密度中心点集-candidates.

主要步骤如下:

Step 1 读取候选密度中心点,设置DHC算法中T,MinPts两个参数,使得DHC聚类的个数和上一过程中接近或相同;

Step 2 用DHC进行聚类,添加最终聚类标号index至每一条密度中心,即以键值对<index, candidate>形式输出至分布式缓存,以利于下一步快速读取.

算法 2.5 DisDHC map3

输入:带标记的基因数据集-G($\overline{g}$);

输出:键值对<index, g>, index-类号, g-基因.

主要步骤如下:

Step 1 从分布式缓存中获得candidates;

Step 2 将每一条基因$\overline{g}$添加的信息与candidates中每一个密度中心点candidate前添加的信息进行比较,如果相同,则将candidate相应的类号index做为基因$\overline{g}$的类号,并且删除$\overline{g}$的前部信息得到初始基因g,以键值对<index, g>的形式输出至reduce;

3 实验及分析

3.1 实验设置及数据集描述

本文实验所使用的云计算平台共有21个节点,每个节点处理器为Intel(R) Xeon(R) CPU E5620,主频为2.40 GHz,操作系统为64位Debian Linux操作系统.Hadoop平台版本为hadoop-0.20.2.

本文实验使用了3个典型的基因表达数据集[20]:酵母数据集(GAL)、酵母细胞周期数据集(CellCycle)和人血清数据集(Serum).

为了测试DisDHC算法的执行效率,我们将上述数据集分别复制规模为1 000, 2 000, 4 000, 6 000, 8 000, 10 000, 100 000, 1 000 000合成数据集.

3.2 聚类评价准则

本文采用Rand指数做为实验的评价准则.Rand是一种衡量聚类结果一致性程度的指标.Rand指数的值越大,则代表DisDHC聚类与标准聚类一致性程度越高.

假设样本集$X=\{x_1,x_2,\cdots,x_n\}$,聚类结果划分为$R=\{R_1,R_2,\cdots,R_N\}$,标准聚类划分为$S=\{S_1,S_2,\cdots,S_M\}$.Rand指数定义如下:

$$R=\frac{a+d}{a+b+c+d} \tag{8}$$

其中,

$$a=\sum_{i,j}C_{n_{ij}}^2,\ b=\sum_i C_{n_i}^2-\sum_{i,j}C_{n_{ij}}^2,$$

$$c=\sum_j C_{n_j}^2-\sum_{i,j}C_{n_{ij}}^2,\ d=C_n^2-a-b-c;$$

n_i和n_j分别表示S_i和R_j中个体数,n_{ij}表示同属于S_i和R_j的个体数.

3.3 实验结果分析

第一组实验观察了在不同规模的人血清数据集上,DHC算法和DisDHC算法聚类的运行时间.从

表 1 可以看出，DisDHC 算法极大地缩短了聚类时间，并且在数据集规模超100 000之后仍然可以获得有效结果，而 DHC 算法则无法运行.

表 1　不同规模 Serum 数据集上 DHC 和 DisDHC 聚类时间比较

Tab. 1　Clustering time comparison of DHC and DisDHC algorithms on Serum datasets with different size

Serum 规模/条	DHC 聚类时间/s	DisDHC 聚类时间/s	Map 数量/个
1 000	8.49	6.25	2
2 000	31.88	25.33	2
4 000	109.36	27.78	4
6 000	240.49	35.53	6
8 000	431.02	39.64	8
10 000	675.72	42.57	10
100 000	—	131.48	40
1 000 000	—	1 761.28	80

第二组实验观察了在数据规模为 10 000 的 GAL 和 CellCycle 数据集上，DHC 和 DisDHC 算法的聚类时间以及 Rand 指数. 从表 2 可以看出，在保持 Rand 指数基本不下降的情况，DisDHC 能够缩短聚类时间.

表 2　规模 10 000 的 GAL 和 CellCycle 数据集上 DHC 和 DisDHC 聚类时间以及 Rand 指数对比

Tab. 2　Clustering time and Rand index comparison of DHC and DisDHC on GAL and CellCycle dataset at size of 10 000

数据集	DHC 时间/s	DisDHC 时间/s	DHC Rand	DisDHC Rand	聚类个数
GAL	1 158.67	109.31	0.942	0.938	4
CellCycle	582.77	98.45	0.787	0.803	5

第三组实验观察了在数据规模 10 000 的 Serum、GAL 和 CellCycle 数据集上 DisDHC 算法较 DHC 算法的加速比以及随着集群规模增大 DisDHC 算法执行效率的变化. 如图 4 和图 5 所示，随着集群规模的增大，加速比接近于直线趋势增长，执行时间接近于直线趋势下降.

第四组实验考察了两个参数：MinPts 和 T，对 DisDHC 聚类算法的影响. 本组实验分别在数据规模10 000的 GAL 数据集和 CellCycle 数据集上进行，图 6 和图 7 分别展示了 Rand 指数随参数 MinPts 和 T 变化而受到的影响，表 3 和表 4 分别给出了聚类个数随参数 MinPts 和 T 变化而受到的影响. 结合表 3 和图 6 可以看出，参数 T 在区间 $[0.81,0.85]$ 变化，MinPts 取值为 9 的情况下，

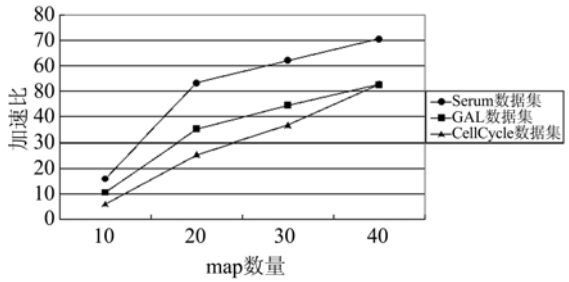

图 4　在不同集群规模下 DisDHC 算法在 Serum、GAL 和 CellCycle 数据集上的加速比变化趋势

Fig. 4　Speedup trend of DisDHC on Serum, GAL and CellCycle dataset in different cluster scale

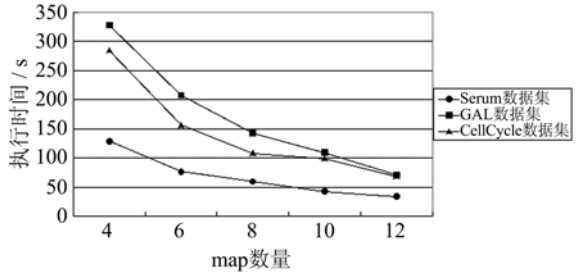

图 5　在不同集群规模下 DisDHC 算法在 Serum、GAL 和 CellCycle 数据集上的执行时间变化趋势

Fig. 5　Running time of DisDHC on Serum, GAL and CellCycle dataset in different cluster scale

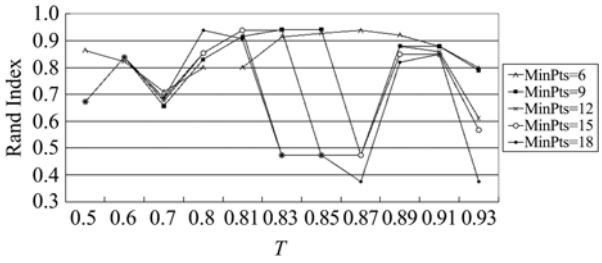

图 6　不同 MinPts 取值下对规模 10 000 GAL 数据集进行 DisDHC 聚类 Rand 指数的变化

Fig. 6　Rand Index of DisDHC with different MinPts on GAL dataset at size of 10 000

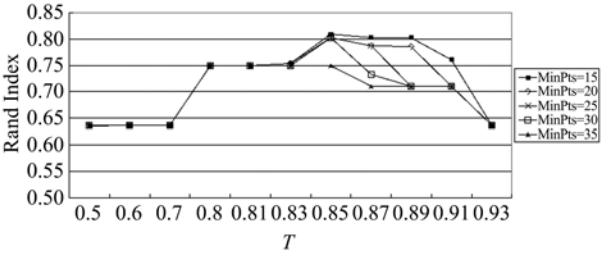

图 7　不同 MinPts 取值下对规模 10 000 CellCycle 数据集进行 DisDHC 聚类 Rand 指数的变化

Fig. 7　Rand Index of DisDHC with different MinPts on CellCycle dataset at size of 10 000

表 3 在规模10 000 GAL 数据集上不同 MinPts 和 T 取值对 DisDHC 聚类个数的影响

Tab. 3 Cluster's number of DisDHC with different MinPts and T parameters on GAL dataset at size of 10 000

MinPts	T				
	6	9	12	15	18
0.5	8	6	6	6	6
0.6	10	7	7	6	6
0.7	10	5	4	4	4
0.8	8	6	5	5	4
0.81	8	5	4	4	3
0.83	6	4	4	2	2
0.85	6	5	2	2	2
0.87	5	2	2	2	1
0.89	5	3	3	2	2
0.91	5	4	3	2	2
0.93	7	5	4	2	1

表 4 在规模 10000CellCycle 数据集上不同 MinPts 和 T 取值对 DisDHC 聚类个数的影响

Tab. 4 Cluster's number of DisDHC with different MinPts and T parameters on CellCycle dataset at size of 10 000

MinPts	T				
	15	20	25	30	35
0.5	2	2	2	2	2
0.6	2	2	2	2	2
0.7	2	2	2	2	2
0.8	4	4	4	4	4
0.81	4	4	4	4	4
0.83	5	5	4	4	4
0.85	7	6	6	6	4
0.87	7	6	5	4	3
0.89	5	3	3	2	3
0.91	5	4	3	2	2
0.93	2	2	2	2	2

DisDHC 在数据规模10 000的 GAL 数据集上 Rand 指数能够获得较好的结果，并且聚类的个数与 GAL 的外部标准类个数接近或者相同. 而在数据规模 10 000的 CellCycle 数据集上，从表 4 和图 7 可以看出，参数 T 在区间[0.83，0.89]变化，MinPts 取值为 20 时，Rand 指数较高并且和聚类个数接近真实类个数.

第五组实验观察了 DisDHC 聚类算法在三个实验数据集上聚类效果. 我们在每个数据集上选出一个聚类结果，进行展示. 具体如图 8～10 所示. 其中圆圈表示基因，中间的数字代表该基因实际的类别. 根节点代表的基因在对应的聚类中具有最大的

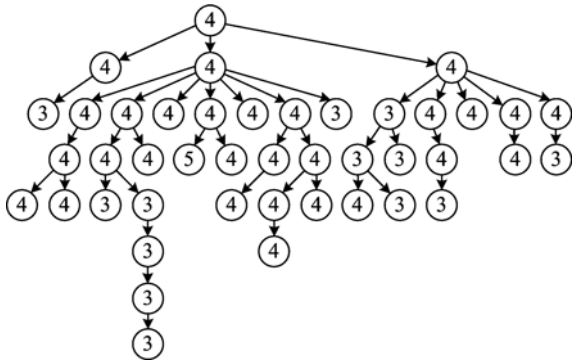

图 8 在 CellCycle 数据集上第 4 个聚类效果图

Fig. 8 The diagram of the forth cluster in the CellCycle data set

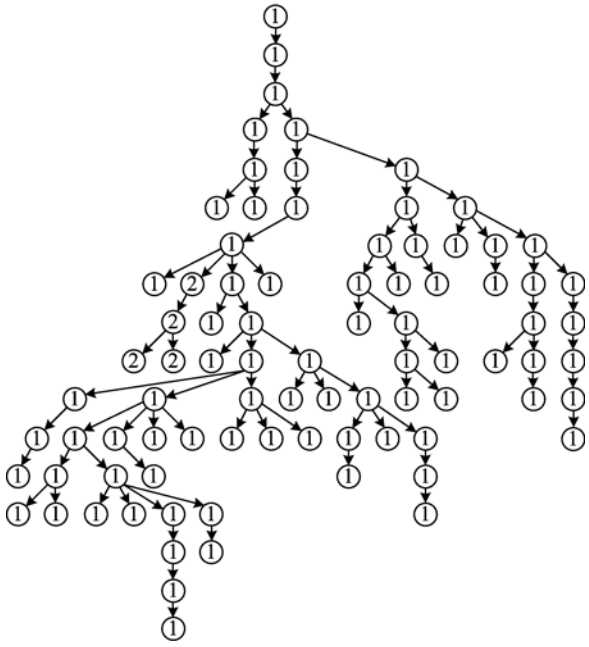

图 9 在 GAL 数据集上第 1 个聚类效果图

Fig. 9 The diagram of the first cluster in the GAL data set

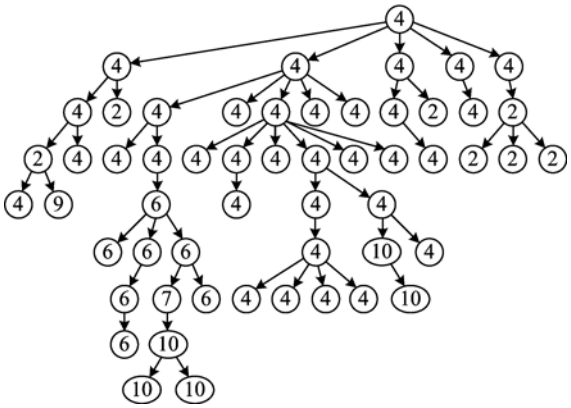

图 10 在 Serum 数据集上第 4 个聚类效果图

Fig. 10 The diagram of the forth cluster in the Serum data set

密度，即该聚类的密度中心.第二层节点代表的基因直接被根节点基因吸引，其他的基因被根节点基因间接吸引，叶子节点是该聚类的边界.

4　结论

DisDHC 聚类对数据集进行稀疏化，获得候选密度中心点，然后再次对候选点使用 DHC 聚类，产生整体密度中心点.本文使用流行的 MapReduce 框架来实现 DisDHC 聚类算法，结果表明，它能够在保持聚类效果基本不变的情况下，极大地减少聚类时间.

该算法涉及两个参数：MinPts 和 T，对这两个参数的选取可能在一定程度上影响算法的聚类效果，因此需要进一步研究对参数的选取问题.

参考文献(References)

[1] Gilbert D R, Schroeder M, van Helden J. Interactive visualization and exploration of relationships between biological objects[J]. Trends in Biotechnology, 2000, 18(12): 487-494.

[2] Eisen M B, Brown P O. DNA arrays for analysis of gene expression[J]. Methods Enzymology, 1999, 303: 179-205.

[3] Jiang D X, Pei J, Zhang A D. DHC: A density-based hierarchical clustering method for time series gene expression data[C] // Proceedings of the 3rd IEEE Symposium on Bioinformatics and Bioengineering. Washington, USA: IEEE Press, 2003: 393-400.

[4] Jain A K, Murty M N, Flynn P J. Data clustering: A review[J]. ACM Computing Surveys, 1999, 31(3): 264-323.

[5] Bach F R, Jordan M I. Learning spectral clustering [R]. Proceedings of Neural Information Processing Systems, No. UCB/CSD-03-1249, Berkeley, USA, 2003.

[6] Frey B J, Dueck D. Clustering by passing messages between data points[J]. science, 2007, 315(5814): 972-976.

[7] Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns [J]. Journal of Computational Biology, 1999, 6(3-4): 281-297.

[8] Tamayo P, Slonim D, Mesirov J, et al. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation [J]. Proceedings of the National Academy of Sciences, 1999, 96(6): 2 907-2 912.

[9] Zhao W Z, Ma H F, He Q. Parallel k-means clustering based on MapReduce[C]// Proceedings of the First International Conference on Cloud Computing. Beijing, China: Springer, 2009: 674-679.

[10] Chen W Y, Song Y Q, Bai H J, et al. Parallel spectral clustering in distributed systems[J]. Pattern Analysis and Machine Intelligence, 2011, 33(3): 568-586.

[11] 鲁伟明，杜晨阳，魏宝刚，等. 基于 MapReduce 的分布式近邻传播聚类算法[J]. 计算机研究与发展，2012，49(8)：1 762-1 772.

[12] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[13] 刘鹏. 实战 Hadoop：开启通向云计算的捷径[M]. 北京：电子工业出版社，2011.

[14] Schena M, Shalon D, Davis R W, et al. Quantitative monitoring of gene expression patterns with a complementary DNA microarray[J]. Science, 1995, 270(5235): 467-470.

[15] Lockhart D J, Dong H, Byrne M C, et al. Expression monitoring by hybridization to high density oligonucleotide arrays [J]. Nature Biotechnology, 1996, 14(13): 1 675-1 680.

[16] Jiang D X, Pei J, Zhang A D. Gpx: Interactive mining of gene expression data[C]// Proceedings of the 30th International Conference on Very Large Databases. Toronto, Canada: ACM Press, 2004: 1 249-1 252.

[17] Bustin S A. Absolute quantification of mRNA using real-time reverse transcription polymerase chain reaction assays [J]. Journal of Molecular Endocrinology, 2000, 25(2): 169-193.

[18] Cho R J, Campbell M J, Winzeler E A, et al. A genome-wide transcriptional analysis of the mitotic cell cycle[J]. Molecular Cell, 1998, 2(1): 65-73.

[19] Yeung K Y, Haynor D R, Ruzzo W L. Validating clustering for gene expression data[J]. Bioinformatics, 2001, 17(4): 309-318.

[20] 杨春梅. 基因表达数据聚类分析算法研究和应用[D]. 天津大学，2006.