一种基于 MapReduce 的半监督近邻传播算法*

冯兴杰[a,b]，王文超[a†]
（中国民航大学 a. 计算机科学与技术学院；b. 信息网络中心，天津 300300）

摘　要： 近邻传播（affinity propagation，AP）算法是一种具有较高准确度的聚类算法，但是其具有较高的时间复杂度，且无法有效聚类结构松散数据。针对这两个问题，提出了一种基于 MapReduce 的半监督近邻传播算法（MR-SAP）。首先利用 MapReduce 编程框架，在各个数据节点上运行 AP 算法，得到局部的聚类中心，以及代表每一个局部聚类中心成为全局聚类中心可能性的决策系数；然后综合局部聚类中心进行全局的 AP 聚类，其中初始参考度的选取依据输入的决策系数；最后通过引入 IGP 聚类评价指标比较聚类效果，引导算法向结果最优方向运行。实验结果表明该算法在处理不同大小、不同类型数据集时均具有良好的效率和扩展性，且具有较高的聚类精度。

关键词： 近邻传播；聚类；半监督；IGP（类内比例）；MapReduce
中图分类号： TP301.6　　**文献标志码：** A　　**文章编号：** 1001-3695（2018）07-2011-04
doi：10.3969/j.issn.1001-3695.2018.07.020

Semi-supervised affinity propagation algorithm based on MapReduce

Feng Xingjie[a,b], Wang Wenchao[a†]
（*a. School of Computer Science & Technology, b. Information Network Center, Civil Aviation University of China, Tianjin* 300300, *China*）

Abstract: Affinity propagation algorithm is a high accuracy clustering algorithm, but it has high time complexity, and can not effectively cluster loosely structured data. In order to solve these two problems, this paper proposed a semi-supervised affinity propagation algorithm based on MapReduce (MR-SAP). Firstly, it used the MapReduce programming framework to run the AP algorithm in each data node, obtained the clustering centers locally, and also obtained decision coefficient which represented each local clustering center for the possibility of global clustering center. Then it combined local AP clustering center for running global AP, selected the initial preference based on the decision coefficient of input. Finaly through the comparison of clustering results based on IGP clustering evaluation index, it made the algorithm run in the best direction. Experiments show that MR-SAP has good efficiency and scalability in dealing with different sizes and different types of data sets, and has high clustering accuracy.

Key words: affinity propagation (AP); clustering; semi-supervised; IGP (in-group proportion); MapReduce

近几年，各行各业所产生的数据量呈现出一种爆炸性增长的趋势，为了应对海量数据的处理需求，大数据技术迅速崛起，成为科技界和企业界甚至世界各国关注的热点[1]。其中 Hadoop 作为 Apache 软件基金会下的一个开源分布式计算平台，它的出现解决了大数据并行计算、存储、管理[2]等关键问题，用户可以在不了解分布式底层细节的情况下开发分布式程序[3]。HDFS 和 MapReduce 是 Hadoop 的两大核心技术：HDFS 是一个分布式文件系统，为分布式程序的开发提供了可靠的存储架构；MapReduce 是一个分布式编程框架，简化了分布式程序的开发过程。基于此，Hadoop 生态系统成为目前最为成熟且流行的大数据问题解决方案。聚类是一种有效的数据挖掘方法，迄今为止，研究人员已经提出了很多聚类算法，大体上可以将这些算法分为基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法、基于模型的方法[4]。但是进入大数据时代，传统的聚类算法无法在有限的时间内解决海量数据的聚类问题，因此有必要结合大数据技术对传统的聚类算法进行改造，使之适用于大数据环境下的聚类需求。

近邻传播算法（AP）是 Frey 等人在 2007 年提出的一种新型的聚类算法，该算法因其具有较高的聚类精度，从提出以来一直受到广泛的关注。但是 AP 算法一方面具有较高的时间复杂度，使得它无法处理海量数据，同时对于结构复杂且比较松散的数据集，无法有效聚类[5]。因此本文提出了一种基于 MapReduce 的半监督近邻传播算法。算法首先对数据进行拆分，进行局部的 AP 聚类，得到局部的聚类中心，引入决策系数的定义，它表征的是每一个局部聚类中心相比于同一个计算节点上的其他局部聚类中心成为全局聚类中心的可能性，这个决策系数随着局部聚类中心一起输出；其次对局部聚类中心进行全局 AP 聚类，初始参考度的值由所有输入的局部聚类中心的相似度以及输入的决策系数决定，而不再简单地设置为所有相似度的中位数；最后引入 IGP 聚类评价指标对每次迭代之后的聚类结果进行评估，通过调整参考度引导算法向结果最优方向运行。

1　相关技术及方法

1.1　MapReduce 编程框架

MapReduce 是 Hadoop 生态系统的计算核心，它是一款分布式的编程框架，它通过将数据分割、并行处理等问题进行底层封装，保证用户在开发基于 MapReduce 的应用程序时只需要考虑程序的逻辑实现问题，简化了开发难度[6,7]。

框架的核心是 map 任务与 reduce 任务，所处理的数据以〈key，value〉键值对的形式存在。MapReduce 首先对存储在 HDFS 上的输入数据进行分片，分配给承担 map 任务的计算节点进行处理，默认地会对输出结果进行排序（sort 过程），将排序之后的中间结果存储在本地磁盘之上；然后承担 reduce 任务的数据节点根据计算需求从相应节点的本地磁盘复制中间结果到本节点上，对数据进行合并，运行 reduce 任务，计算的最终结果输出到 HDFS 上。可以看出，在大多数情况下，用户只需要将逻辑实现集成到 map 与 reduce 任务上即可完成基本的 MapReduce 的开发，简单易行。MapReduce 的基本实现流程如图 1 所示。

收稿日期： 2017-04-16；**修回日期：** 2017-05-25　　**基金项目：** 国家自然科学青年基金资助项目（61301245，61201414）
作者简介： 冯兴杰（1969-），男，河北邢台人，教授，硕导，主要研究方向为数据库及数据仓库、智能信息处理理论与技术；王文超（1991-），男（通信作者），河北沧州人，硕士研究生，主要研究方向为数据挖掘，大数据技术（chaozicauc@126.com）.

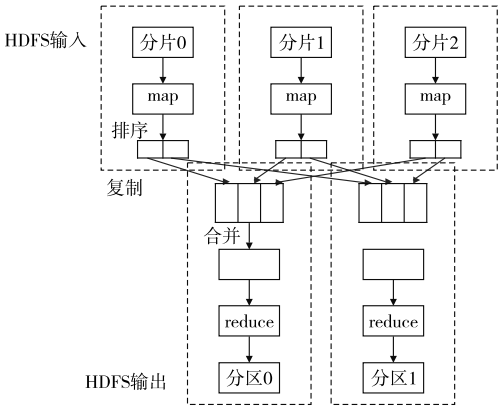

图1 MapReduce实现流程

1.2 近邻传播算法

近邻传播算法是一种新型的基于代表点(exemplar)的聚类算法,算法初始将所有的数据对象看成潜在的类代表点,通过不断迭代更新数据对象之间的关系确定一组最具有代表性的数据对象作为最终的类代表点[8,9]。算法主要通过维护相似度矩阵、吸引度矩阵和归属度矩阵三个矩阵实现。首先对于相似度矩阵,相似度是表征数据对象相似程度的变量,AP算法的相似度定义如式(1)所示。

$$s(i,k) = -\| x_i - x_k \|^2 \tag{1}$$

其中:i、k 表示数据对象,相似度定义为数据对象之间的距离差的平方的相反数。

吸引度 $r(i,k)$ 中,i 表示数据对象,k 表示作为候选代表点的数据对象。吸引度反映的是候选代表点 k 适合作为 i 的聚类中心的证据积累,如图2所示,其中 k' 是与 k 竞争的候选代表点。

吸引度的定义如式(2)所示。

$$r(i,k) = s(i,k) - \max_{k'\text{s.t.}\,k'\neq k}\{a(i,k') + s(i,k')\} \tag{2}$$

其中:$a(i,k)$ 代表归属度,归属度反映的是数据对象 i 选择候选代表点 k 作为其聚类中心的证据积累,如图3所示。

归属度的定义如式(3)所示。

$$a(i,k) = \min\{0, r(k,k) + \sum_{i'\text{s.t.}\,i'\notin\{i,k\}} \max\{0, r(i',k)\}\} \tag{3}$$

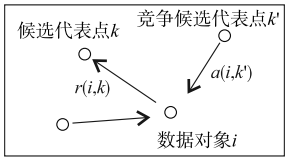

图2 吸引度生成过程

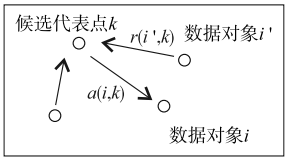

图3 归属度生成过程

在AP算法中存在一个重要的定义即参考度 p。p 定义为 $s(k,k)$,即相似度矩阵的对角线元素,它并不是自动生成而是由人为设定的,p 的选择与最终的聚类结果具有极大的相关性。p 值过大,造成最终产生的类簇数偏多;p 值过小,结果的类簇数偏少,选择的 p 值是否合适影响着算法的精度。在基本的AP算法中,一般选择所有相似度的中位数作为输入的 p 值,这样产生的聚类结果数居于中等。

数据对象能否成为最终的代表点取决于自吸引度 $r(k,k)$、自归属度 $a(k,k)$。根据吸引度与归属度定义,自吸引度定义如式(4)所示,自归属度定义如式(5)所示,且初始所有的归属度包括自归属度均设为0。

$$r(k,k) = p - \max_{k'\text{s.t.}\,k'\neq k}\{a(k,k') + s(k,k')\} \tag{4}$$

$$a(k,k) = \sum_{i'\text{s.t.},i'\neq k} \max\{0, r(i',k)\} \tag{5}$$

在算法运行过程中可能会出现聚类结果摇摆不定的现象,称此时发生了振荡。为了避免振荡的发生,引入振荡变量 λ,此后连续两次迭代中,吸引度与归属的关系如式(6)(7)所示,变量 m 表示第 m 次迭代。

$$r_m(i,k) = (1-\lambda)\times\{s(i,k) - \max_{k'\text{s.t.}\,k'\neq k}\{a(i,k') + s(i,k')\}\} + \lambda \times r_{m-1}(i,k) \tag{6}$$

$$a_m(i,k) = (1-\lambda)\times\min\{0, r(k,k) + \sum_{i'\text{s.t.}\,i'\notin\{i,k\}} \max\{0, r(i',k)\}\} + \lambda \times a_{m-1}(i,k) \tag{7}$$

基于以上公式的定义,近邻传播算法的过程如下:

a)根据输入数据计算相似度矩阵、参考度,这里参考度取值一般取所有相似度的中位数。

b)设置最大迭代次数以及振荡变量,振荡变量 λ 一般取值为0.5,计算吸引度与归属度值,构造吸引度矩阵和归属度矩阵并进行矩阵的迭代更新。对于数据对象 k 是否为聚类中心由 $r(k,k) + a(k,k)$ 是否大于0确定,大于0时则认为它是一个聚类中心。

c)当迭代次数超过最大迭代次数或者迭代多次聚类结果均未发生变化,则输出聚类中心以及每一个数据对象所属的类别,算法结束。

1.3 聚类有效性评价指标

IGP(in-group proportion)是2007年由斯坦福大学的Kapp等人提出的一种高效的聚类有效性评价指标。IGP表示一个比例,即对于一个属于某一类别的数据对象,距离它最近的另一个数据对象也属于同一类别,满足以上要求的数据对象占总体的比例即为IGP[10]。IGP的定义如式(8)所示。

$$\text{IGP}(u,X) = \frac{\#\{j \mid \text{class}_X(j) = \text{class}_X(j^N) = u\}}{\#\{j \mid \text{class}_X(j) = u\}} \tag{8}$$

其中:u 表示类别,X 表示属于相应类别的数据对象,j 是数据对象,j^N 表示距离 j 最近的数据对象,$\text{class}_X(j) = \text{class}_X(j^N) = u$ 表示二者属于同一类别,#表示满足情况的数目。所有聚类的平均IGP指标越大表示聚类的质量越好;此外它不适用于聚类数为1的情况。

2 基于MapReduce的半监督近邻传播算法(MR-SAP)

2.1 MR-SAP算法基础

定义1 决策系数(dcf)。设对于数据集 D,经过AP算法的处理得到 n 个聚类中心的集合为 $\{e_1, e_2, \cdots, e_n\}$,其中 $e_i (i \in [1,n])$ 的 $r(e_i,e_i) + a(e_i,e_i)$ 的值最大(这样的 e_i 可能有多个),则对任意聚类中心 $e_j (j \neq i)$ 的决策系数 $\text{dcf}(e_j) = \{r(e_j,e_j) + a(e_j,e_j)\} / \{r(e_i,e_i) + a(e_i,e_i)\}$ $(0 < \text{dcf} \leqslant 1)$,而具有最大值的聚类中心 e_i 的决策系数为1。

证明 根据AP算法定义,数据点 k 是否能够作为聚类中心取决于 $r(k,k) + a(k,k)$ 是否大于0,而这个值的大小表示的就是相比于其他聚类中心,它作为聚类中心的优越性。因此考察所有产生的聚类中心,取最大值作为基准,其他值与该最大值的比值定义为决策系数,表征这种优越性。

王开军等人[11]曾提出使用Silhouette聚类评价指标确定AP算法的最佳聚类数,然而Kapp等人提出IGP后,通过实验证明,IGP指标对于聚类结果的评估要优于Silhouette,且周世兵等人[12]通过对几种聚类评价指标的对比实验证明了IGP指标相比于其他包括Calinski-Harabasz、Davies-Bouldin、weighted inter-intra、Krzanowski-Lai、Hartigan在内的聚类评价指标,更适合于对AP算法进行评估,确定最佳聚类的数目。因此基于以上理论采用IGP指标对近邻传播算法进行优化,提出了基于IGP调整的半监督AP算法伪代码,如算法1所示。

算法1 基于IGP调整的半监督AP算法(IGP-SAP)

输入:相似度矩阵 S,数据量 n,参考度 p,振荡变量 λ,最大迭代次数 maxite,最大平衡迭代次数 maxhite。

初始化:所有数据的归属度 $a=0$,记录平衡迭代次数 hite=0,表征聚类结果最优的变量 IGP=0。

```
for iter = 1 -> maxite do
    更新吸引度矩阵(式(2)和(6));
    更新归属度矩阵(式(3)和(7));
    for i = 1 -> n do
        if a(i,i) + r(i,i) > 0 //式(4)和(5)
          i 是聚类中心;
        end if
    end for
    将所有非聚类中心的数据对象划分到各个聚类中心;
    计算所有非聚类中心数据对象的平均IGP值(式(8)),设为newIGP;
    if newIGP > IGP;
```

```
    IGP = newIGP;  //寻找到当前最佳聚类数
    hite = 0;      //平衡迭代次数置 0
    p = p + p/10;  //调整参考度
  else hite + +
  end if
  if hite≥maxhite   //满足最大平衡迭代次数
    跳出循环;
    output 各个聚类中心以及其他数据对象所属类别;
  end if
  迭代次数大于最大迭代次数,循环结束;
end for
output 各个聚类中心以及其他数据对象所属类别
```

2.2 MR-SAP 算法

基于以上算法基础,应用大数据编程框架 MapReduce 提出了基于 MapReduce 的半监督近邻传播算法。算法由三个 MapReduce 过程实现,其中 map1 实现对输入数据的划分,降低相似度矩阵、吸引度矩阵、归属度矩阵计算的时间复杂度;reduce1 应用基本的 AP 算法实现对划分数据的初步聚类,得到局部的聚类中心和每个局部聚类中心的决策系数;map2 应用基于 IGP 调整的 AP 算法实现对局部聚类中心的聚类,得到全局聚类中心,并计算所有分类的 IGP 值,调整参考度 p 值,直到得到具有最大 IGP 值的全局聚类中心;map3 负责将数据划分到不同聚类中心;reduce3 负责将具有相同类编号的数据汇总输出。算法 map1 的流程如算法 2 所示。

算法 2　map1 算法

```
输入:数据集 D(以〈key,value〉形式)。
for i = 1 - >D. length do
    //利用随机数生成器对数进行划分
    key′ = new random(). nextInt() × √(D. length);
    context. write(key′,value);
end for
```

将划分之后的数据交由 reduce 算法处理运行 AP 算法,算法 reduce1 的流程如算法 3 所示。

算法 3　reduce1 算法

```
输入: key′相同的数据集合〈key′,value_list〉,最大迭代次数 maxite,振荡变量 λ。
初始化:λ = 0.5,所有数据的归属度 a = 0。
for value in value_list do
    for value′in value_list do
        计算相似度,构建相似度矩阵 S(式(1));
    end for
end for
选择相似度的中位数作为参考度 p 值;
for i = 1 - > maxite do
  更新吸引度矩阵(式(2)(6));
  更新归属度矩阵(式(3)(7));
  for i = 1 - > value_list. length do
      if a(i,i) + r(i,i) > 0 //式(4)(5)
        第 i 个数据是局部聚类中心;
        将 a(i,i) + r(i,i)的值存储到数组 dcf 中;
        将第 i 个数据存储到数组 ap_value 中;
      end if
  end for
  寻找数组 dcf 中的最大值,根据定义 1 计算数组各个元素的决策系数,更新数组 dcf;
end for
for i = 1 - > dcf. length do
  context. write(dcf[i],ap_value[i])
end for
```

将得到的决策系数和局部聚类中心交由 map2 算法处理,运行基于 IGP 调整的半监督 AP 算法,算法 map2 的流程如算法 4 所示。

算法 4　map2 算法

```
输入:局部聚类中心以及各自的决策系数(以〈dcf,value〉形式)。
计算局部聚类中心的相似度,构造相似度矩阵 S(同算法 3 相似度计算);
筛选出所有相似度的中位数最为基础的参考度 p 值,根据决策系数,更新每个局部聚类中心的 p 值;
更新规则为 p = p × dcf;
同步更新相似度矩阵 S(即更改对角线元素);
将更新之后的相似度矩阵 S 作为 IGP-SAP 算法输入(其他输入参数与 IGP-SAP 相同),更新求解全局聚类代表;
for i = 1 - > 全局聚类代表数目 do
    context. write(i,全局聚类代表); //i 作为全局聚类代表的编号
end for
```

将全局聚类代表存储到分布式缓存中,交由 map3 算法处理,将原始的数据划分到不同的类中。算法 map3 的流程如算法 5 所示。

算法 5　map3 算法

```
输入:原始数据集 D(同算法 1)。
从缓存中读取全局聚类结果,包括类编号;
for i = 1 - > D. length do
    for j = 1 - > 全局聚类代表数目 do
      计算相似度,相似度最大的全局聚类代表作为数据的所属类别;
      context. write(类编号,数据);
    end for
end for
```

Map3 将结果交由 reduce3 汇总输出。由于 MapReduce 会将经过 map 函数处理之后具有相同 key 值的数据汇总输出,所以在 map3 算法中用类编号作为 key 值输出,reduce3 直接输出 map3 传来的结果。

图 4 是 MR-SAP 算法的整体执行流程。

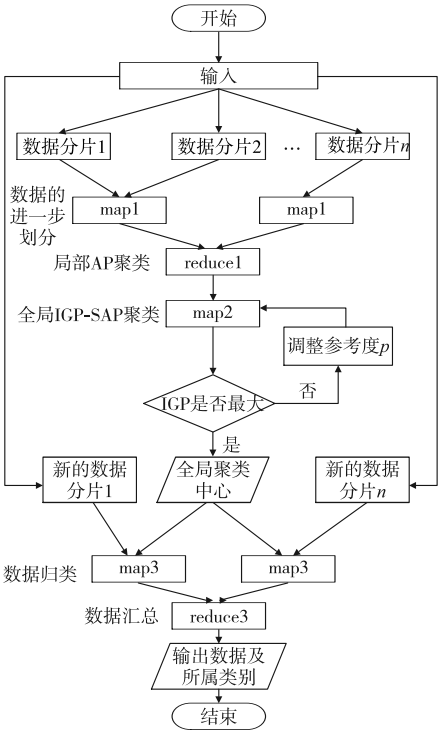

图4　MR-SAP执行流程

3　实验结果及分析

为了验证算法关注的两个问题,实验从算法加速比、扩展性、效率、聚类簇数、结果准确率五个方面进行设计。

3.1　实验环境

分布式实验环境由九个节点构成,master 为主节点,slave1 ~ slave8 为计算节点,各节点配置如表 1 所示。

表 1　实验环境配置

节点名称	内存	硬盘	CPU 型号
master	4 GB	1 TB	i7 3770
slave1	1 GB	250 GB	奔腾 e2140
slave2	1 GB	250 GB	奔腾 e2140
slave3	1 GB	250 GB	奔腾 e2140
slave4	1 GB	250 GB	奔腾 e2140
slave5	1 GB	250 GB	奔腾 e2140
slave6	1 GB	250 GB	奔腾 e2140
slave7	1 GB	250 GB	奔腾 e2140
slave8	1 GB	250 GB	奔腾 e2140

此外各节点的操作系统选择 Ubuntu Linux 14.04.2 LTS,

JDK 版本为1.7.0,Hadoop 版本为2.4.0,节点之间使用百兆交换机连接。

3.2 实验

为了验证算法具有处理大数据的能力,实验数据采用人工生成数据,构造了大小为1 GB、2 GB、4 GB、8 GB,属性维度为10、类簇数为48的四个不同大小的松散结构数据集。

加速比是指同一任务在单处理器和多处理器系统中运行消耗时间的比率,用来衡量并行化性能的一个有效指标。因此为了考察算法的并行化性能,应用以上数据进行算法的加速比性能实验。实验对四个数据集分别在2、4、6、8个节点上运行算法,实验结果如图5所示。为了使对比明显,加入了线性加速比结果。从图5中可以看出,对于同一数据集,随着计算节点数的增加,加速比逐渐增加,并且增加趋势接近线性,但是与线性加速比有一定差距,且节点数越大,其差距越大。主要原因在于,节点数的增加使得节点之间的通信、数据传递所消耗的时间也逐渐增加。然后对比不同数据集的加速比,数据量越大,加速比越接近线性加速比,原因是大数据集的数据计算所消耗的时间占总耗时的比例相比于小的数据集要更大。

为了进一步验证算法的扩展性,对四个数据集分别在2、4、6、8个数据计算节点上运行算法,考察算法的运行时间,实验结果如图6所示。由图6可知,随着计算节点的增加,各组数据的运行时间均呈现下降趋势,且趋势接近线性,说明算法具有良好的可扩展性,而未呈线性的原因同样是由于节点之间的通信。

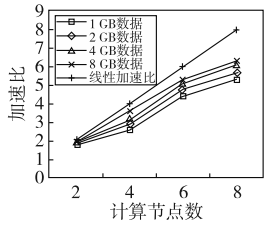

图5 MR-SAP加速比性能对比

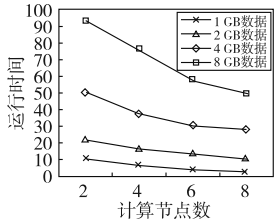

图6 MR-SAP扩展性性能对比

为了验证算法效率,第三组实验引入 DisAP 算法[13]、MR-Kmeans 算法作为 MR-SAP 的对比算法进行实验,数据选择以上数据集中大小为2 GB的数据集。仍然在2、4、6、8个节点上分别运行三种算法处理该数据集,实验结果如图7所示。由图7可以看出,三种算法随着计算节点数的下降,运行时间都呈现出接近线性的下降趋势,因而都具有比较良好的可扩展性。但是在运行效率上,MR-SAP 算法优于其他两种算法,分析原因可知,实验数据是松散数据集,MR-Kmeans 算法和 DisAP 算法无法有效处理松散数据集,为了寻求比较优良的聚类结果,要迭代更多次数、耗时更多,因此可以证明,MR-SAP 具有良好的效率。

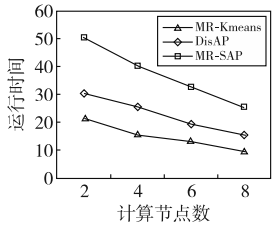

图7 算法效率对比

为了验证算法对于不同类型数据集,是否都能得到准确的类簇数,选择聚类数作为评价指标。选择的数据集包括五个数据集合,其中一个是前文人工生成的2 GB数据集,其他四个选择公共数据集,如表2所示。

表2 公共实验数据集信息

数据集	实例数	属性维度	类簇数	结构
wine	178	3	3	松散
iris	150	4	3	紧密
ionosphere	351	34	2	松散
glass	214	9	6	紧密

由于公共数据集数据量较小,实验选择两个计算节点,聚类结果如表3所示。

表3 聚类簇数结果

数据集	算法		
	MR-Kmeans	DisAP	MR-SAP
wine	13	11	3
iris	3	3	3
ionosphere	22	25	2
glass	5	6	6
2 GB 人工数据集	78	56	48

由表3可以看出,对于不同类型的数据集,MR-SAP 都给出了正确的聚类数。DisAP 算法对于结构比较紧密的数据集基本给出了正确的聚类数,而对于结构复杂的松散数据集却无法给出正确的聚类数;MR-Kmeans 算法具有最差的聚类准确率,五个数据集中,只有 iris 数据集给出了正确的聚类数。基于以上分析可以证明,MR-SAP 对于不同结构类型的数据均能得到准确的类簇数。

最后,对于给定的数据集考察聚类的准确率,即每一个实例是否被正确地划分到原本所属类别,结果如表4所示。

表4 聚类准确率 /%

数据集	算法		
	MR-Kmeans	DisAP	MR-SAP
wine	75.2	80.6	97.4
iris	97.6	98.8	98.2
ionosphere	63.5	68.4	96.0
glass	89.0	99.6	99.2
2 GB 人工数据集	57.5	67.2	93.5

由表4可以看出,对于松散结构数据,MR-SAP 算法在聚类的准确率上都优于其他两类算法;对于比较紧密的数据集,DisAP 的性能要略优于 MR-SAP,但相差不大,性能基本保持一致。

4 结束语

在大数据时代,近邻传播算法虽然具有较高聚类精度,但其无法处理结构复杂松散的海量数据。本文提出了基于 Map-Reduce 的半监督近邻传播算法,引入 IGP 聚类指标和决策系数的概念解决结构复杂松散问题,并引入 MapReduce 编程框架解决处理海量数据问题。实验结果表明该算法成功解决了以上两个问题,并证明了算法的有效性。

参考文献:

[1] 程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述[J]. 软件学报, 2014,25(9): 1889-1908.
[2] Perera S, Gunarathne T. Hadoop MapReduce 实战手册[M]. 杨卓荦,译. 北京:人民邮电出版社, 2015.
[3] 夏靖波, 韦泽鲲, 付凯, 等. 云计算中 Hadoop 技术研究与应用综述[J]. 计算机科学, 2016, 43(11): 6-11,48.
[4] 伍育红. 聚类算法综述[J]. 计算机科学, 2015, 42(Z6):491-499.
[5] White T. Hadoop 权威指南[M]. 曾大聃, 周傲英, 译. 北京: 清华大学出版社, 2010.
[6] 肖宇, 于剑. 基于近邻传播算法的半监督聚类[J]. 软件学报, 2008, 19(11): 2803-2813.
[7] 陆嘉恒. Hadoop 实战[M].2 版. 北京: 机械工业出版社, 2012.
[8] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.
[9] Givoni I E, Chung C, Frey B J. Hierarchical affinity propagation [C]//Proc of the 27th Conference on Uncertainty in Artificial Intelligence. Arlington, VA: AUAI Press, 2012:238-246.
[10] Kapp A V, Tibshirani R. Are clusters found in one dataset present in another dataset[J]. Biostatistics, 2007, 8(1):9-31.
[11] 王开军, 张军英, 李丹, 等. 自适应仿射传播聚类[J]. 自动化学报, 2007, 33(12): 1242-1246.
[12] 周世兵, 徐振源, 唐旭清. 基于近邻传播算法的最佳聚类数确定方法比较研究[J]. 计算机科学, 2011, 38(2): 225-228.
[13] 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): 1762-1772.