第46卷第3期 中国科学技术大学学报 Vol. 46, No. 3
2016年3月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Mar. 2016

文章编号:0253-2778(2016)3-0173-07

一种新的软聚类投票法及其并行化实现

张静静,杨 燕*,王红军,韩晓涛,邓 强

(西南交通大学信息科学与技术学院,成都 611756)

摘要:聚类集成作为数据挖掘的重要应用工具,得到了广泛的认可和研究.本文在投票法的基础上提出一种新的软聚类投票(VMSC)算法.算法首先求取平均隶属度矩阵,然后进行迭代优化.该算法能够消除噪声点影响,具有很好的稳定性.Spark云计算平台能够高效处理大数据.为了提出的算法处理大数据,在Spark云计算平台上实现并行的VMSC算法.VMSC算法实验用12组UCI数据集进行验证,并与sCSPA、sMCLA、sHGBF及SVCE等软聚类算法进行对比.结果表明,VMSC算法对软聚类算法具有较好的集成效果.在Spark云计算平台上对VMSC算法并行实现.实验表明,该算法具有较理想的并行效果,能够有效处理大数据.

关键词:软聚类集成;投票;云计算;大数据

中图分类号:TP301 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.03.001

引用格式:ZHANG Jingjing, YANG Yan, WANG Hongjun, et al. A novel voting method and parallel implementation for soft clustering[J]. Journal of University of Science and Technology of China, 2016, 46(3):173-179.

张静静,杨 燕,王红军,等. 一种新的软聚类投票法及其并行化实现[J]. 中国科学技术大学学报, 2016,46(3):173-179.

A novel voting method and parallel implementation for soft clustering

ZHANG Jingjing, YANG Yan*, WANG Hongjun, HAN Xiaotao, DENG Qiang

(*School of Information Science and Technology, Southwest Jiaotong University, Chengdu* 611756)

Abstract: As an important tool of Data Mining, clustering ensemble has been widely recognized and studied. This paper proposes a novel voting method for Soft Clustering(VMSC). The ensemble process consists of two steps: calculating the average degree of membership matrix as the input of the second step, and iterative optimization. This method deals well with eliminating the influences of noise and has good stability. The cloud computing platform of Spark handles big data efficiently. The VMSC algorithm was parallelizod to make it suitable for big data on Spark Cloud Computing platform. In the VMSC experiments, 12 UCI datasets were used to test it, and its results were compared with 4 other soft clustering ensemble algorithms: sCSPA, sMCLA, sHGBF and SVCE. The experiments indicate that the VMSC algorithm has a better integration effect. And the parallel experiments show that its parallel implementation manages big data efficiently.

Key words: soft clustering ensemble; voting; cloud computing; big data

收稿日期:2015-08-27;**修回日期**:2015-09-29

基金项目:国家自然科学基金项目(Nos. 61134002,61170111,61572407)资助.

作者简介:张静静,女,1989年生,硕士. 研究方向:数据挖掘、云计算. E-mail: youyouzhangjing@yeah.net.

通讯作者:杨燕,博士/教授. E-mail: yyang@swjtu.edu.cn.

0 引言

聚类作为数据挖掘的一种重要工具,随着数据挖掘研究的发展,其重要性得到人们的肯定.目前,聚类已经被广泛的应用于模式识别、机器学习、图像处理、信息检索、数据挖掘、时空数据库应用和序列和异类数据分析等[1].另外,聚类分析对生物学、心理学、地理学、地质学、考古学以及市场营销等的研究也有重要作用[2-3].所谓聚类,就是数据对象按照簇内相似度最大、簇间相似度最小的原则进行划分.虽然近几十年来涌现了很多聚类算法,但是尚没有哪种单一的聚类算法能够发现任意形状和结构的簇[4].聚类集成技术是解决这一问题的一种有效方法,聚类集成是将某一数据集的多个聚类结果综合,寻找出一个优于单个聚类算法的结果.

聚类集成的概念由 Strehl 等首次提出,并提出了三种基于图分割思想的聚类集成算法:CSPA、HGPA 和 MCLA[5].Dimitriadou 等在 2002 年提出了一种模糊聚类组合的方法[6],该方法是求出所有的聚类隶属度的方差中最小的值作为集成的隶属度.Fern 等在 2004 年提出了一种将聚类集成问题转化为二分图划分的问题的聚类集成方法[7].Topchy 等提出了一种基于有限混合多维分布概率模型的共识函数[8].Fred 等提出了一种基于证据积累(evidence accumulation, EAC)的聚类集成算法[9],该算法首先根据证据积累得出数据对象间的相似度矩阵,然后用凝聚层次聚类算法得出最终的聚类结果.Zhou 等提出了基于标签对齐的四种聚类集成方法[10].Nguyen 等提出了迭代投票共识函数(iterative voting consensus, IVC),并且实现了 IPVC 和 IPC 两个版本的算法[11].Tumer 等提出了一种自适应投票聚类集成算法[12],并证明了这种方法不仅适用于无噪声环境,在噪声环境下也能非常有效.Wang 等提出了一种基于投票模型的软投票聚类集成算法[13],该算法基于联合概率计算的方法,对软聚类进行集成.Ren 首次把对象权重运用到集成过程中[14],由共协矩阵来确定权重,把权重与共识函数融合.2014 年,Chakeri 提出将集成过程转化为权重顶点权重边图的求解过程[15],求图的最大团,即集成结果;Dumonceaux 提出了一种基于海廷代数表示的集成算法,从代数的角度为聚类集成建模求解[16];Su 等提出了一个模糊聚类集成算法[17],定义了 FCO、Flink、FCTS 模糊聚类对象间的相似度量,建立模糊共协矩阵,用分层聚类算法得到集成结果.Hao 等在 2015 年提出了 WETU 算法[18],用来定义数据集聚类结果中簇的相似度.Zhong 等提出了一种两层次提取共协矩阵的方法[19],该方法不仅从数据对象被分到同一类上考虑相似性,同时也考虑了聚类结果簇之间所包含的关系.

近年来,社会生活的信息化使得数据以前所未有的速度增长.为了在这种大数据环境下高效地挖掘出潜在的有价值信息,就需要有可靠的大数据处理平台.云计算便是一种处理大数据的分布式计算环境.第一代并行计算平台 MapReduce[20] 的出现,使得用户能够使用由 PC 组成的集群快速有效地对海量数据进行统计分析,但是 MapReduce 最擅长的是离线海量数据的分析,处理结果的获取往往要延迟几分钟至几个小时,这种缺陷在很多场景下是不可接受的.美国加州大学伯克利分校的 AMPLab 的 Spark 云计算平台,与 Hadoop 不同的是它立足于内存计算,不仅性能超出 Hadoop 百倍,还支持 Interactive Query、流计算和图计算等,拥有 Hadoop 所无法比拟优势.

本文基于投票模型,提出一种针对软聚类的投票算法,该算法根据软聚类算法生成的隶属度矩阵,计算平均的隶属度;然后借用激励函数进行迭代优化;设计该算法的并行实现模式并在 Spark 平台上实现;最后用实验对其进行验证,研究其在并行模式下的效率.

1 相关原理

聚类按照划分的不同分为硬聚类和软聚类两种.硬聚类的结果是一个表示数据对象标签的向量,某个数据属于一类的概率是 0 或 1;软聚类的结果是个隶属度矩阵,表示的是数据对象属于一类的概率,该概率属于[0,1].本文介绍是一种针对软聚类的集成算法.

1.1 多数投票法

多数投票法[21]是一种简单直观的聚类集成算法.在聚类集成过程中,多数投票法将数据对象划分到多数聚类结果所选中的簇.设数据集 X,该数据集分为 c 簇,对该数据集有 m 个基聚类结果,用 $d_{ij}\in\{0,1\}(i=1,\cdots,m;j=1,\cdots c)$ 表示在第 i 个聚类结果中某数据是否被分到第 j 类;若被分到第 j 类,则 $d_{ij}=1$,否则 $d_{ij}=0$.多数投票法决定某一数据被分到第 k 类公式如下:

$$\sum_{i=1}^{m} d_{ik} = \max_{j=1}^{c} \sum_{j=1}^{c} d_{ij} \tag{1}$$

1.2 Spark 云计算平台

Spark 是一个基于内存计算的分布式计算框架. 它包含两个重要的抽象概念:弹性分布式数据集(resilient distributed datasets, RDD)[22] 和共享变量.

RDD 是 Spark 的核心概念,是一个容错、并行的数据结构,是一个只读的分区记录集合. 一个 RDD 通常可以包含多个分区,分区即数据集片段. 有两种方法可以产生 RDDs:一种是并行驱动程序中的数据集;另一种是导入外部存储系统的数据集,外部存储系统可以是共享文件系统、HDFS、HBase 或者其他可被 Hadoop 所用的数据形式. RDDs 支持两种类型的操作:转化(transformation)和行动(action). 转化操作是由已存在的 RDD 生成新的 RDD,通常情况下该过程不会立即执行运算,该操作只是记录转化模式,只有到行动步骤的时候才会分配到节点上执行,如 Map 、flatMap、filter 等都是转化操作. Action 通常情况下是经过计算后返回到驱动程序一个值,如 Reduce、Collect 等操作.

共享变量即在程序执行过程中需要被拷贝到每个节点的变量,这些变量通常情况下不会由节点传播到驱动程序. Spark 提供两种类型的共享变量:广播变量和累加器.

2 VMSC 算法及并行化

2.1 VMSC 算法思想

本文提出一种针对软聚类的投票法. 假设数据集 X 包含 N 个数据对象,被分为 k 类,软聚类得到的结果是一个 $N\times k$ 的隶属度矩阵,软聚类投票得到集成隶属度矩阵计算函数定义为

$$u_{ij} = \frac{1}{m}\sum_{l=1}^{m} u_{ij}^{l} \tag{2}$$

式中, u_{ij} 表示第 i 个数据属于第 j 类的隶属度,m 为基聚类结果的数目, u_{ij}^{l} 表示在第 l 个基聚类结果中数据对象 i 属于第 j 类的隶属度.

对于软聚类结果,将所求平均值作为最终的隶属度有一个缺点:模糊度小的基聚类结果会对最终结果影响较大. 针对这一缺点,本文用激励函数进行调整,采用的函数模型为

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{3}$$

为了使得调整粒度可伸缩,用式(3)变形作为调整函数模型,变形函数如下:

$$f(x) = \frac{1}{1+\mathrm{e}^{-\sigma x}} - \frac{1}{2} \tag{4}$$

式中,σ 是一个大于 0 的实数. σ 取不同值时对应的曲线图如图 1 所示.

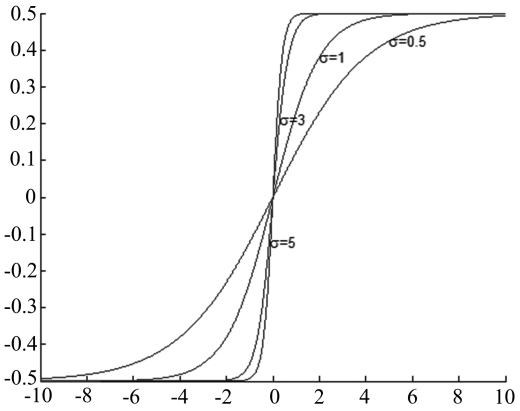

图 1　σ 不同取值下的激励函数曲线图

Fig. 1　The Sigmod function curves with different σ values

对公式(4)和 $\frac{f(x)}{x}$ 求导:

$$f'(x) = \frac{\mathrm{e}^{-x}}{(1+\mathrm{e}^{-x})^2} > 0 \tag{5}$$

$$\left(\frac{f(x)}{x}\right)' = \frac{(x-1)\mathrm{e}^{-x}-1}{x^2\,(1+\mathrm{e}^{-x})^2},\ (x\neq 0) \tag{6}$$

由式(5)和(6)可知,式(4)有如下性质:

性质 2.1　$f(x)$为奇函数且在定义域范围内单调递增;

性质 2.2　$\frac{f(x)}{x}$是一个偶函数;

性质 2.3　$\frac{f(x)}{x}$在 $x>0$ 上单调递减,在 $x<0$ 上单调递增.

由性质 2.2 和性质 2.3 可知,式(4)中,x 的取值越接近 0,函数值的变化率越大. 假设 u_{ij} 表示上一次生成的数据对象 i 属于第 j 类的隶属度值, u_{ij}^{new} 表示新一次迭代生成的隶属度值,聚类集成阶段的优化公式定义如下:

$$u_{ij}^{\mathrm{new}} = u_{ij} + \sum_{l=1}^{m} f(u_{ij}^{l} - u_{ij}) \tag{7}$$

式(7)循环迭代,便可得到满足条件的隶属度值. 由于一个离群点的意见对应着多个反对点的意见,迭代优化过程会使隶属度值向多数意见靠拢,消除离群点的影响,增强稳定性. 例如,取 $m=3$,假设 u_{ij}^{1}和 u_{ij}^{2}大于均值,u_{ij}^{3}小于均值,则有

$$f(u_{ij}^1-u_{ij})+f(u_{ij}^2-u_{ij})>f(u_{ij}-u_{ij}^3) \quad (8)$$

由公式(7)可知，$(u_{ij}^1-u_{ij})+(u_{ij}^2-u_{ij})=u_{ij}-u_{ij}^3$，$u_{ij}^1-u_{ij}<u_{ij}-u_{ij}^3$ 且 $u_{ij}^2-u_{ij}<u_{ij}-u_{ij}^3$，由性质2.2和性质2.3可推出式(8)成立. 迭代生成的 u_{ij}^{new} 会向 u_{ij}^1 和 u_{ij}^2 靠拢.

VMSC算法流程如下：

算法 2.1 VMSC

输入：软聚类结果

输出：隶属度矩阵U和标签L；

Step 1 根据基聚类结果求取隶属度平均值；

Step 2 平均隶属度值作为初始 u_{ij}，用公式(7)计算 u_{ij}^{new}；

Step 3 判断是否满足停止条件，若满足则继续下一步骤，否则用 u_{ij}^{new} 作为新的均值，跳转到Step2；

Step 4 根据满足条件的隶属度矩阵，提取标签，输出隶属07度矩阵U和标签L.

2.4 算法并行化

并行化的实现采用Spark的数据并行思想，将数据分布到多个节点上同时进行运算. VMSC算法的输入是多个软聚类隶属度合并后的矩阵. 矩阵的一行代表数据集中的某个数据对象在多个聚类结果中属于各类的隶属度. 数据被读入Spark创建成RDD数据集，被划分到多个节点上. 集群根据算法的公式执行Map操作生成新的RDD，新RDD是输入的隶属度矩阵和生成的聚类中心的组合，该过程的RDD结构变化如下：

$$(\text{Value1})\Rightarrow(\text{Value1},\text{Value2}).$$

其中，Value 1为输入某对象的组合隶属度矩阵，Value 2为生成的结果隶属度矩阵. 根据算法的迭代优化过程执行一系列的Map操作，对结果进行优化，直到得出满意的结果. 并行化实现的具体流程如图2所示.

输入数据以文件形式输入，第一步读取文件中的数据. 第二步是根据读取的数据创建包含输入数据的弹性分布式数据集RDD. 第三步是实现VMSC算法的Step2，即对应的一系列的Transformation操作，该过程根据原有的RDD生成包含输入数据和均值隶属度的RDD，然后再在新的RDD上计算新的均值隶属度矩阵，并更新RDD. 第四步即Action操作，对应Step3的获取判定值操作. 该过程根据连续两次的隶属度矩阵结果，计算两者的差值. 第五步用第四步得到的值和设定的阈值比较，如果小于阈值或者迭代次数达到最大迭代次数，则结束算法；若不满足，则跳转到第二步继续执行，直到满足停止条件.

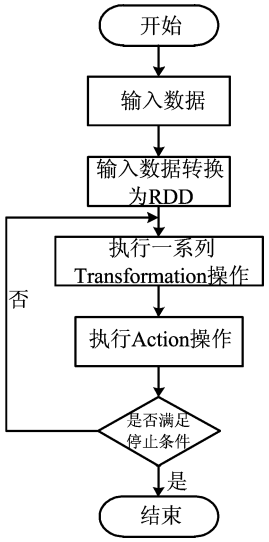

图2 VMSC并行化实现流程图

Fig. 2 The flow chart of the parallel VMSC

3 实验结果与分析

3.1 VMSC算法实验

3.1.1 实验数据集

实验选用12组来自UCI的数据集，其中msplice是moive数据集的拼接. 数据集的详细信息描述如表1所示. 表1描述了数据名称、数据对象个数、数据维数和数据分类数等信息.

表1 数据集详细信息描述

Tab. 1 The detailed information on Datasets

数据集	数据个数	属性	类别
blood	748	4	2
landsat	2 000	36	6
diabetes	768	8	2
Phishing	2 456	30	2
satimage	6 435	36	6
synthetic	600	60	6
Dermatology	366	34	6
pima	768	8	2
msplice	3 175	240	3
vehicle	846	18	4
german	1 000	24	2
parkinsons	195	22	2

3.1.2 评价标准和实验结果

采用的聚类为FCM[23]、GMM聚类[24-25]和KFCM聚类；标签对齐采用文献[26]提出的标签对

齐方法;对比算法采用文献[13]提出的 SVCE 算法和文献[27]提出的 sCSPA、sMCLA 和 sHBGF 三种算法.

实验的评价采用 RI(rand index)[28]和准确率(Accuracy)等评价标准.设 SS 为两个点属于同一类且在聚类结果中被划分为同一类的对象组数,SD 为两个点属于同一类但在聚类结果中未被划分到同一类的对象组数,DS 为属于不同类的两个点在聚类结果中被划分到同一类的对象组数,DD 为属于不同类的两个点在聚类结果张被划分到不同类的对象组数,则 RI 评价指标的定义如下:

$$\mathrm{RI}=\frac{\mathrm{SS}+\mathrm{DD}}{\mathrm{SS}+\mathrm{SD}+\mathrm{DS}+\mathrm{DD}} \tag{9}$$

Accuracy 定义如下:

$$A=\frac{t}{N} \tag{10}$$

式中,t 为被正确分类的数据对象个数,N 为数据集数据对象总数.表 2 和表 3 分别为 12 组数据集在 $\sigma=1$,运行 20 次的平均 RI 和 Accuracy 值.

表 2　5 种算法上的平均 RI 值

Tab. 2　The average RI values of the 5 algorithms

数据集	sCSPA	sHBGF	sMCLA	SVCE	VMSC
blood	0.502 6	0.505 6	0.517 0	0.562 1	**0.563 3**
landsat	0.829 8	0.842 8	0.814 1	0.841 5	**0.844 2**
diabetes	0.553 8	0.519 3	0.549 9	0.561 4	**0.565 9**
Phishing	0.767 2	0.767 8	0.775 7	0.783 1	**0.784 2**
satimage	0.812 1	0.807 6	0.817 9	0.856 7	**0.857 4**
synthetic	0.536 0	**0.612 1**	0.566 3	0.557 5	0.552 8
Dermatology	0.816 0	0.829 0	0.808 0	0.851 0	**0.855 4**
pima	0.548 8	0.550 9	0.551 9	0.552 2	**0.554 2**
msplice	**0.589 6**	0.557 2	0.425 8	0.384 9	0.465 1
vehicle	0.649 6	0.650 0	0.647 9	0.636 5	**0.653 5**
german	0.503 0	0.503 2	0.542 4	0.545 7	**0.548 5**
parkinsons	0.515 2	0.514 4	0.551 0	0.559 2	**0.565 8**

表中黑体的为效果最好的数据,由表 2 看出,在 12 组数据集上,在 RI 评价标准上 VMSC 算法有 10 个取得最好.RI 评价指标越好,表示同类被分在同一簇,不同类被分到不同簇所占的比重越大.实验结果表明,同类的能够被分在一起,不同类的能够有效区分开;在 Accuracy 评价标准上也有 6 个取得最优值,准确率越大说明被正确分类的对象个数越多.实验结果表明,VMSC 算法能够有效地分类.通过以上对比可知,VMSC 算法能够有效地对软聚类集成,且能获得较好的集成结果,集成结果更倾向于将同一类划分到一起,不同类划分开.图 3 自左向右分别是表 2,表 3 对应的图,通过图我们能更直观地看到结果.

表 3　5 种算法上的平均 Accuracy 值

Tab. 3　The average Accuracy values of the 5 algorithms

数据集	sCSPA	sHBGF	sMCLA	SVCE	VMSC
blood	0.536 1	0.554 7	0.433 6	0.677 0	**0.678 6**
landsat	0.630 1	**0.659 9**	0.536 2	0.612 6	0.641 8
diabetes	0.664 8	0.599 7	0.658 9	0.676 0	**0.682 3**
Phishing	0.862 6	0.854 3	0.871 3	0.873 8	**0.874 7**
satimage	0.536 7	0.576 5	0.598 0	**0.697 6**	0.694 0
synthetic	0.634 4	**0.736 8**	0.682 2	0.669 6	0.662 7
Dermatology	0.603 0	0.552 5	0.512 9	**0.687 1**	0.624 8
pima	0.657 2	0.659 4	0.661 5	0.662 4	**0.665 4**
msplice	0.516 0	0.429 4	0.315 5	**0.519 1**	0.420 7
vehicle	0.377 3	**0.381 3**	0.369 7	0.338 4	0.376 0
german	0.541 5	0.538 9	0.397 8	0.651 4	**0.656 2**
parkinsons	0.593 8	0.591 8	0.646 2	0.672 3	**0.682 6**

3.2　并行化实验

3.2.1　数据集

并行实验的数据集采用 UCI 的 covtype、SUSY 和 HIGGS 数据集,数据集的详细信息描述如表 4 所示.表 4 描述了数据集的数据对象个数、维数和分类数等详细信息.

表 4　并行化实验数据集详细信息

Tab. 4　The detailed information on Datasets that used in parallel experiments

数据集	数据个数	维数	分类数
covtype	581 012	54	7
SUSY	5 000 000	18	2
HIGGS	11 000 000	28	2

3.2.2　评价标准和实验结果

并行实验的性能测试,采用加速比(SpeedUp)、数据伸缩率(SizeUp)和扩展率(ScaleUp)三个指标.其中 SpeedUp 是指同一个任务在单个节点的运行时间和多个节点上的运行时间的比值.SizeUp 是指在节点不变的情况下,数据集不断增加的运行时

间的比值. ScaleUp 是指当节点与数据量同比增长的情况下的运行时间比率.

3 个数据子集的实验加速比和数据伸缩率和扩展率对比分别如图 4 所示. 图 4 中的左图为节点数和加速比的关系图,图中当节点增加时加速比呈线性增长,虽然图中 SUSY 的加速比有个别点低于 covtype 数据集,但总体上数据规模越大,加速比越高,说明算法具有良好的并行性,且更适用于大规模数据. 从图 4 中 SizeUP 结果图可以看出,三个数据集在节点数不变,数据集增大的情况下基本呈线性增长,说明该算法具有良好的稳定性. 图 4 中的 ScaleUp 结果图中 covtype 数据集的 ScaleUP 值偏离 1.0 最远,HIGGS 数据集虽然有浮动,但是离1.0 最近,说明算法具有良好的扩展率,且数据量越大,扩展性越好. 综合图 4 来看,数据集越大,其并行性越好,数据集越小,其并行性能相对差,说明算法具有良好的并行性,且更适用于大规模数据.

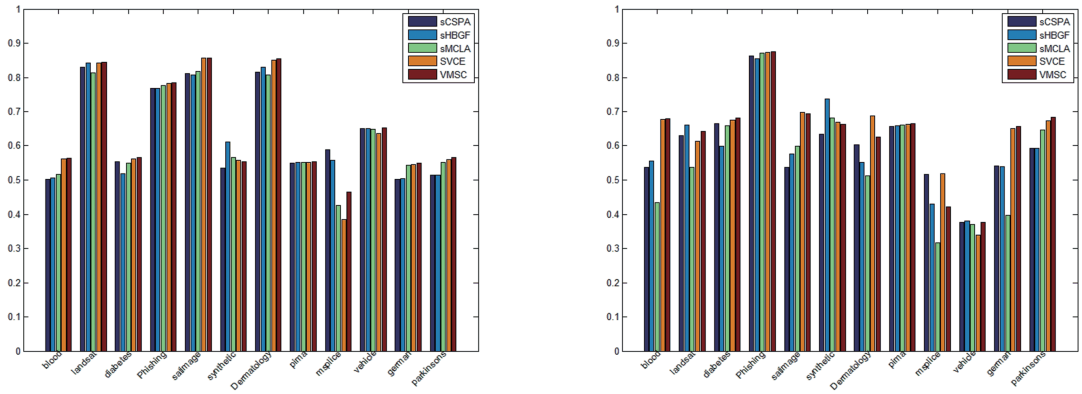

图 3 5 种算法的 RI 和 Accuracy 评价指标图

Fig. 3 The average values of RI and Accuracy index of the 5 algorithms

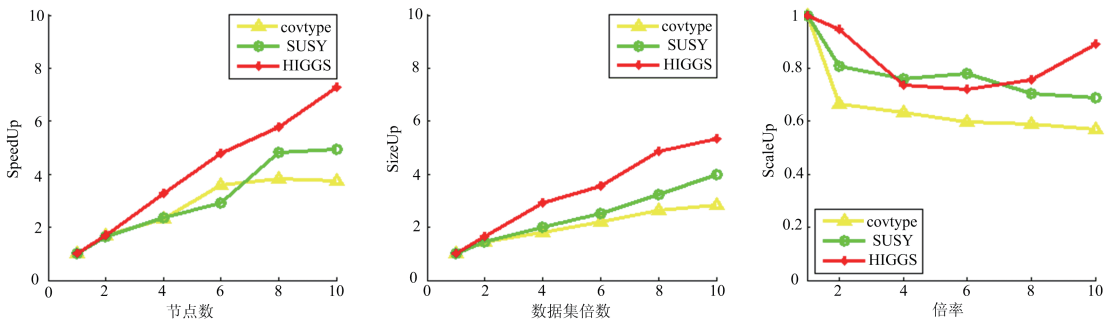

图 4 VMSC 算法在 3 个数据集上的 SpeedUP、SizeUp 和 ScaleUP 值

Fig. 4 The VMSC's SpeedUp, SizeUp and ScaleUp values of the 3 Datasets

4 结论

本文提出了一种新的软投票聚类集成算法,并在 Spark 平台上实现其并行化. 该算法首先计算隶属度矩阵的均值,该均值作为迭代优化过程的输入,优化集成结果. 优化过程能够使最终隶属度矩阵向大多数意见靠拢,降低噪声点的影响,增强稳定性. 实验表明,VMSC 算法具有较好的集成效果. 并行化实验表明,该算法的并行实现能够有效地处理大数据,且随着数据规模的增大,算法性能越好. 在今后的工作中将考虑优化该算法,发掘其他软聚类集成算法,加入半监督信息,对算法的并行化做更多的研究.

参考文献(References)

[1] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48-61. SUN Jigui, LIU Jie, ZHAO Lianyu. Clustering algorithms research[J]. Journal of Software, 2008, 19(1): 48-61.

[2] CADES I V, SMYTH P, MANNILA H. Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction [C]// Proceedings of the 7th ACM SIGKDD. San Francisco, USA: ACM Press, 2001: 37-46.

[3] JAIN A K, MURTY M N, FLYNN P J. Data clustering: A review[J]. ACM Computing Surveys, 1999, 31(3): 264-323.

[4] WOLPERT D H, MACREADY W G. No free lunch theorems for search[R]. Santa Fe: Santa Fe Institute, Technical Report: SFI-TR-95-02-010, 1996.

[5] STREHL A, GHOSH J. Cluster ensembles--a knowledge reuse framework for combining multiple partitions [J]. The Journal of Machine Learning Research, 2003, 3(1): 583-617.

[6] DIMITRIADOU E, WEINGESSEL A, HORNIK K. A combination scheme for fuzzy clustering [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2002, 16(7): 901-912.

[7] FERN X Z, BRODLEY C E. Solving cluster ensemble problems by bipartite graph partitioning[C]//Proceedings of the 21st International Conference on Machine Learning. Banff , Canada: ACM Press, 2004: 36.

[8] TOPCHY A, JAIN A K, PUNCH W. A mixture model for clustering ensembles[C]//Proceedings of the SIAM International Conference on Data Mining. Baltimore, USA: SIAM Press, 2004: 379.

[9] FRED A L N, JAIN A K. Combining multiple clusterings using evidence accumulation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(6): 835-850.

[10] ZHOU Z H, TANG W. Clusterer ensemble [J]. Knowledge-Based Systems, 2006, 19(1):77-83.

[11] NGUYEN N, CARUANA R. Consensus clusterings [C]//Proceedings of the 7th International Conference on Data Mining. Omaha, USA: IEEE Press, 2007: 607-612.

[12] TUMER K, AGOGINO A K. Ensemble clustering with voting active clusters [J]. Pattern Recognition Letters, 2008, 29(14):1947-1953.

[13] WANG H, YANG Y, WANG H, et al. Soft-Voting Clustering Ensemble [A]// Lecture Notes in Computer Science, 2013, 7872: 307-318.

[14] REN Y, DOMENICONI C, ZHANG G, et al. Weighted-object ensemble clustering[C]// Proceeding of the 13th International Conference on Data Mining. IEEE Press, 2013: 627-636.

[15] CHAKERI A, HALL L O. Dominant sets as a framework for cluster ensembles: An evolutionary game theory approach [C]//Proceeding of 22nd International Conference on Pattern Recognition. Stockholm, Sweden: IEEE Press, 2014: 3457-3462.

[16] DUMONCEAUX F, RASCHIA G, GELGON M. An algebraic approach to ensemble clustering [C]// Proceeding of 22nd International Conference on Pattern Recognition Stockholm, Sweden: IEEE Press, 2014: 1301-1306.

[17] SU P, SHANG C, SHEN Q. Link-based pairwise similarity matrix approach for fuzzy c-means clustering ensemble [C]//Proceeding of 22nd International Conference on Fuzzy Systems. IEEE Press, 2014: 1538-1544.

[18] HAO Z F, WANG L J, CAI R C, et al. An improved clustering ensemble method based link analysis[J]. World Wide Web, 2015, 18(2): 185-195.

[19] ZHONG C, YUE X, ZHANG Z, et al. A clustering ensemble: Two-level-refined co-association matrix with path-based transformation [J]. Pattern Recognition, 2015, 48(8): 2699-2709.

[20] DEAN J, GHEM A S. MapReduce: Simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[21] STEPENOSKY N, GREEN D, KOUNIOS J, et al. Majority vote and decision template based ensemble classifiers trained on event related potentials for early diagnosis of Alzheimer's disease[C]//Proceeding of the International Conference on Acoustics, Speech and Signal Processing. Toulouse, France: IEEE Press, 2006: 1935-1941.

[22] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing [C]// Proceeding NSDI'12 Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. San Jose, USA: IEEE Press, 2012, 70(2): 141-146.

[23] DUNN J C. A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters [J]. Journal of Cybernetics, 1974, 3(3): 32-57.

[24] SHENTAL N, BAR-HILLEL A, WEINSHALL D. Computing Gaussian Mixture Models with EM Using Side-Information [A]// Advances in Neural Information Processing Systems, International Conference on Machine Learning. MIT Press, 2003.

[25] KUSHARY D. The EM algorithm and extensions[J]. Biometrics, 20088, 15(1): 154-156.

[26] TOPCHY A P, LAW M H C, JAIN A K, et al. Analysis of consensus partition in cluster ensemble [C]//Proceeding of the 4th IEEE International Conference on Data Mining. Brighton, UK: ACM Press, 2004: 225-232.

[27] PUNERA K, GHOSH J. Consensus-based ensembles of soft clusterings[J]. Applied Artificial Intelligence, 2008, 22(7-8): 780-810.

[28] RAND W M. Objective criteria for the evaluation of clustering methods [J]. Journal of the American Statistical Association, 1971, 66(336):846-850.