文章编号: 1001-0920(2017)06-1075-09 DOI: 10.13195/j.kzyjc.2016.0374

大规模数据集引力同步聚类

乔 颖†, 王士同, 杭文龙

(江南大学 数字媒体学院，江苏 无锡 214122)

摘 要: 受Kuramoto模型启发,构造一种新的万有引力同步模型,用以解决现有同步聚类算法时间复杂度高的问题,并提出大规模数据集的引力同步聚类算法(LSCGS). 首先,使用快速压缩集密度估计(RSDE)算法对大规模数据集进行压缩;然后,通过万有引力同步聚类算法对压缩数据集进行聚类,使用Davies-Bouldin指标自动寻优到最佳聚类数;最后,利用提出的剩余样本聚类(RSC)算法对除压缩集以外的剩余数据进行聚类,可以有效地区分孤立类以及噪声点. 通过在大规模人造数据集、UCI真实数据集和图像数据上的实验,验证LSCGS算法的有效性,与传统同步聚类算法相比,聚类的运算成本得到大幅度的降低.

关键词: 大规模数据；快速压缩集密度估计；万有引力；同步聚类

中图分类号: TP273 **文献标志码:** A

Clustering by gravitational synchronization on large scale dataset

QIAO Ying†, WANG Shi-tong, HANG Wen-long

(School of Digital Media，Jiangnan University，Wuxi 214122，China)

Abstract: Different from the existing synchronization clustering algorithm(Sync) which is recently proposed based on Kuramoto model in physics, and referring to gravitational law, a novel clustering algorithm, called large sample clustering by gravitational synchronization(LSCGS) is proposed for large datasets. Firstly, a large scale dataset is condensed into its reduced dataset by using the reduced set density estimator method. Then, the obtained reduced dataset is clustered by using the proposed gravitational synchronization clustering model with Davies-Bouldin clustering criterion to find out the most suitable clustering results. Finally, the remaining samples in the large dataset are clustered. The proposed method can detect clusters in data of arbitrary shapes, sizes and numbers without any data distribution assumptions. Extensive experiments on the large synthetic dataset, UCI real-world datasets and image segmentations indicate that LSCGS can effectively detect the clusters of the arbitrary shape, and the proposed method achieves high clustering accuracy with lower execution time.

Keywords: large scale dataset；fast reduced set density estimator；gravity；synchronization clustering

0 引 言

在模式识别领域,聚类分析作为一种重要的无监督的数据分析工具,已经在图像分割、数据挖掘、计算机视觉、生物信息学和其他相关领域中有了广泛的应用. 传统的聚类算法可以分为划分方法、层次方法、基于密度方法和基于模型方法等. 近几十年里,越来越多的研究提到了聚类方法,例如K-means[1]、K-medoids[2]、FCM[3]、Expectation Maximization[4]、CLIQUE[5]、BIRCH[6]、DBSCAN[7]和Affinity Propagation[8]. 这些算法有着各自的优缺点,各自不同的模型,但他们的目的都是为了找出数据的“自然分组”,即找出相似的元素,并将这些元素放在一起. 简而言之,聚类分析是将一个数据集中的相似元素归纳到一个集合中的无监督学习过程.

同步现象在自然界中广泛存在,在物理学界已经有了有关同步过程的动力学模型[9]的研究成果,例如Kuramoto模型[10]. 同步化是一组数据的初始频率不相等,但随时间的推移,自发地以相同的频率运动. Böhm等[11]将这种同步化的思想运用到算法中,提出了一种新的聚类算法Sync. 初始时刻,每个振子按照自己的固有频率运动,相互之间没有联系. 随着时间的改变,一些邻近的振子相互影响发生锁相,最终形成多个在局部以相同频率运动的簇. Sync算法在聚类过程中不会受到数据集形状、大小和密度的影

收稿日期: 2016-03-30；**修回日期:** 2016-06-17.

基金项目: 国家自然科学基金项目(61272210, 61170122)；江苏省自然科学基金项目(BK20130155).

作者简介: 乔颖(1992−),女, 博士生, 从事人工智能、模式识别、数据挖掘的研究；王士同(1964−), 男, 教授, 博士生导师, 从事人工智能、模式识别、数据挖掘等研究.

†通讯作者. E-mail: 654410050@qq.com

响,且自动寻优最佳聚类数,在实际运用中具有很大空间. 但是,为了获得最佳的局部聚类结果,Sync算法将附近样本点的每一维分量看作是一个相位振子迭代运算,使它们达到完全的局部同步,时间复杂度高,因此在对大规模数据集进行聚类时具有相当大的局限性.

随着信息量的爆炸式增长,大规模数据[12-13]已经普遍存在于各个研究领域. 由文献[14]给出的示例可以看出,大规模数据的数据量一般是指介于常规数据量和大数据的数据量之间,其规模往往从几千到几万不等. 对于此类大规模数据的学习,传统的数据挖掘算法已经不再适用,主要体现在较高的时间复杂度以及超出常规硬件能力范围的空间复杂度两方面. 因此,开发出具备解决此类大规模数据问题的算法迫在眉睫.

本文针对现有同步聚类无法承受的大规模数据样本,提出一种适用于大规模数据样本引力同步聚类算法(LSCGS). 首先,采用基于核密度的方法[15-16]和中心约束最小球技术[14,17-18]将数据集压缩;然后,使用本文新定义的动力学方程,在获得的压缩集上进行同步聚类,设置r_s作为衡量局部同步程度的参量,通过Davies-Bouldin[19]指标对最近邻的ε值进行自适应过程,自动寻优聚类数;最后,对剩下的数据进行聚类,得到最终的结果. 本文使用新的动力学方程对样本的同步运动进行描述,从万有引力定律这一新的角度去构造动力学模型. 本文受到自然界万有引力定律和粒子之间的相互吸引的启发,在同步聚类算法中,根据牛顿引力定律和牛顿第二定律,构建万有引力动力学模型,即粒子之间彼此相互吸引而向一起聚集,随着时间的推移而形成局部同步簇的过程.

1 理论基础

1.1 Kuramoto模型

Kuramoto模型[10]是物理学界已经探索过的一种典型的同步过程模型,在任何本征频率下运动的相位振子,可以通过相位耦合实现同步. 相位速度的动力学方程为

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t}=\omega_i+\frac{S}{N}\sum_{j=1}^{N}\sin(\theta_i-\theta_j). \tag{1}$$

其中:N表示振子的个数,θ_i表示第i个振子的时变相位,ω_i表示第i个振子的固有频率,S表示耦合度. 当$S=0$时,每个振子以自己的本征频率运动;当$S\neq 0$时,振子之间发生同相位的同步运动.

为了显示振子的同步程度,引入一个全局的序列参量$r(t)$,

$$r(t)\mathrm{e}^{i\psi(t)}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N}\sum_{j=1}^{N}\mathrm{e}^{i\theta_j}. \tag{2}$$

其中:$r(0\leqslant r(t)\leqslant 1)$表示度量振子的一致性,$r(t)$为1时,说明振子达到同步;$\psi(t)$表示振子的平均相位.

以上为以往的同步聚类模型,下面将介绍本文所提出的同步聚类模型——万有引力模型.

1.2 万有引力模型

万有引力和电磁力、弱相互作用力、强相互作用力这3种力构成了自然界的4种基本作用力,万有引力不同于其他3种作用力,它无处不在. 任意两个质点之间都存在相互吸引的万有引力,不受任何介质和物质的化学成分影响,只与两个质点的质量成正比,与质点之间的欧氏距离平方成反比:

$$F=G\frac{m_1m_2}{r^2}. \tag{3}$$

其中:F为两个质点之间的引力,m_1和m_2为两个粒子的质量,G为万有引力常量,r为两个粒子之间的欧氏距离. 根据牛顿第二定律,加速度表达式为

$$\boldsymbol{a}=\frac{F}{m}\cdot\boldsymbol{e_r}, \tag{4}$$

其中$\boldsymbol{e_r}$表示作用力的单位方向. 根据式(3)和(4),将牛顿第二定律与万有引力定律结合得到

$$m\boldsymbol{a}=F=G\frac{m_1m_2}{r^2}\cdot\boldsymbol{e_r}. \tag{5}$$

受牛顿万有引力定律和粒子之间相互吸引和相互作用的思想启发,对万有引力中的公式进行相应的推导变换,构成适合同步聚类的模型,从一个新的角度打开同步的方式.

对于一个有n个粒子数据集$\boldsymbol{X},\boldsymbol{X}_i=(x_i^1,x_i^2,\cdots,x_i^d,\cdots,x_i^m),i=1,2,\cdots,n$. 其中:$m$表示样本的维数,$x_i^d$表示粒子$i$在$d$维数上的位置. 在某一时刻$t$,作用在粒子$i$上来自粒子$j$的万有引力为

$$F_{ij}^d=G\frac{m_i\cdot m_j}{r_{ij}^2}\cdot\frac{x_j^d(t)-x_i^d(t)}{r_{ij}}. \tag{6}$$

其中:r_{ij}表示粒子i与粒子j之间的欧氏距离,$x_j^d(t)-x_i^d(t)$与r_{ij}的比值表示引力的方向,m_1和m_2分别表示粒子i和粒子j的质量. 粒子i在t时刻所受到的合力为

$$F_i^d(t)=\sum_{j=1,j\neq i}^{n}G\frac{m_i\cdot m_j}{r_{ij}^2}\cdot\frac{x_j^d(t)-x_i^d(t)}{r_{ij}}. \tag{7}$$

根据牛顿第二定律,粒子运动的加速度等于作用力与粒子质量的比值,有

$$a_i^d(t)=\frac{F_i^d(t)}{m_i}. \tag{8}$$

由动力学方程可知,粒子运动的速度等于它的初始速度加上加速度乘以时间,粒子的位置变换等于原始位置加上速度乘以时间,所以可以得到如下公式:

$$v_i^d(t+1) = v_i^d(t) + a_i^d(t), \tag{9}$$

$$x_i^d(t+1) = x_i^d(t) + 0.5 \times a_i^d(t). \tag{10}$$

其中: v_i^d 表示粒子 i 在 d 维上的速度大小, x_i^d 表示粒子 i 在 d 维上的位置.

1.3 局部同步聚类

为了将万有引力动力学模型运用到同步聚类中, 文献[20]提到了局部同步的概念. 在现实生活中, 全局同步的现象很少有, 但是部分粒子之间相互影响产生局部同步的现象还是比较常见的, 即所有粒子之间根据内在的结构分成不同的部分, 在各自的小部分中做着相同加速度的同步运动. 因此, 提出样本 ε 最近邻的定义如下:

定理1　样本 $\boldsymbol{x}$ 的 ε 最近邻. 定义 $nb_\varepsilon(\boldsymbol{x})$ 表示样本的最近邻, $\boldsymbol{x}$ 为数据集 $\boldsymbol{C}$ 中的一个样本, n 为数据集 $\boldsymbol{C}$ 中的样本个数, 则

$$nb_\varepsilon(\boldsymbol{x}) = \{\boldsymbol{y} \in \boldsymbol{C} | \mathrm{dist}(\boldsymbol{y}, \boldsymbol{x}) \leqslant \varepsilon\}, \tag{11}$$

其中 $\mathrm{dist}(\boldsymbol{y}, \boldsymbol{x})$ 表示样本点中的距离函数.

利用样本 $\boldsymbol{x}$ 的 ε 最近邻研究样本的局部同步现象, 结合式(7)、(8)和(11)可以变换为

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \frac{1}{2} \sum_{\boldsymbol{y}(t) \in nb_\varepsilon} G \frac{m_{\boldsymbol{x}} \cdot m_{\boldsymbol{y}}}{r_{\boldsymbol{x}_i \boldsymbol{y}_i}^2} \cdot \frac{\boldsymbol{y}_i(t) - \boldsymbol{x}_i(t)}{r_{\boldsymbol{x}_i \boldsymbol{y}_i}} \cdot \frac{1}{m_{\boldsymbol{x}}}. \tag{12}$$

$\boldsymbol{x}_i$ 为样本 $\boldsymbol{x}$ 的第 i 个分量, 因为每个样本的质量大小相等, 对聚类不产生影响, 为了方便计算都设为1, 则样本每个分量的万有引力动力学方程为

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \frac{1}{2} \sum_{\boldsymbol{y}(t) \in nb_\varepsilon} G \frac{\boldsymbol{y}_i(t) - \boldsymbol{x}_i(t)}{r_{\boldsymbol{x}_i \boldsymbol{y}_i}^3}. \tag{13}$$

因此, 第 i 个样本 $\boldsymbol{x}_i$ 在 $t = 0$ 时, $\boldsymbol{x}_i(0) = (x_i^1(0), x_i^2(0), \cdots, x_i^m(0))$ 表示样本 i 在 t 时刻的初始位置, 经过 $t = 1, 2, \cdots, T$ 更新后, 得到样本新的位置为 $\boldsymbol{x}_i(t+1)$.

根据全局序列参量公式(2)定义局部序列参量 r_s, 判断局部同步过程是否完成. 根据文献[20], 给出局部序列参量的定义形式如下:

$$r_s = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{N b_\varepsilon(\boldsymbol{x}_i)} \sum_{\boldsymbol{y} \in nb_\varepsilon(\boldsymbol{x}_i)} \mathrm{e}^{-\|\boldsymbol{y} - \boldsymbol{x}_i\|}. \tag{14}$$

随着时间的推移, 越来越多的数据加入同步时, r_s 值在不断地增大, 当 r_s 的值接近或等于1时, 说明此时当前 ε 下的振子已经接近或达到同步, 此 ε 下的同步聚类过程可判定为结束. 即可理解为形成多个类别小分块, 每个分块以相同的加速度做着同步运动.

图1给出了一个二维数据集做同步聚类的运动过程: 假设 A、B、C、D、O 为一个类群中的点, P_1 和 P_2 为噪声点, 分别对 A 点和 O 点进行受力分析.

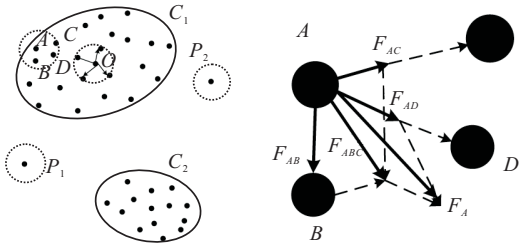


图1　引力同步聚类模型

A 点位于类边缘的位置, B、C、D 点为距离 A 点的 ε 邻近样本. 根据受力分析, 点 A 受到一个向内的力, 产生一个向内的加速度, 向类中心运动. A 点的受力分析见图1(b), F_{AB} 为 A 与 B 之间的引力, A 与 B 相距最近引力也最大, F_{ABC} 为 A 点受到 B 、C 吸引所产生的合力, F_{ABC} 与 F_{AD} 作用产生合力为 F_A, A 点沿这个方向向中心移动. O 点在近类中心的位置, 被其 ε 邻近样本包裹在中间, 从受力分析可以看出, O 点基本处于受力平衡的状态, 不会运动. 可以看出, 处于类边缘的点会向类中间移动, 这样的运动会不断发生, 同一类中的样本会越来越接近, 最终实现局部同步运动, 像 P_1 和 P_2 这样的噪声点会渐渐被孤立. 最终, 数据集被分成几类与噪声点. 综上, 局部同步过程是将样本的每一个分量看作一个粒子, 邻近的粒子之间相互作用, 产生作用力和加速度, 然后向类中心移动, 最终以相同的加速度同步运动, 局部聚类完成. 过程如下:

Step 1: 在起始位置 ($t = 0$) 时, 每个样本点还未与周围点发生作用, 此时定义样本集每个样本为一类, 每个样本点按照自己的速度和加速度运动;

Step 2: 随着时间推移, 在某一时刻 t, 每一个样本与它 ε 邻近样本发生作用, 根据式(13)计算出下一时刻样本位置, 不断重复执行, 直到数据集分为以同一加速度和速度运动的多个集合, 即形成局部的类;

Step 3: 序列参量 r_s 通过式(14)计算, 若 r_s 接近或等于1, 即粒子的同步过程可判定为完成, 聚类算法终止.

2 大规模数据引力同步聚类

根据前面的内容, 引力同步聚类模型同样需要对样本中的每个分量的每一维进行迭代计算, 这样便使得算法的时间复杂度很高, 为 $O(T \times n^2)$, n 为数据集中样本的数量, T 为算法迭代的次数. 为了让本文提出的引力同步聚类运用到大规模样本的数据集中, 首先通过基于概率密度估计和最小球包含技术的压缩方法对原始数据进行压缩, 获得压缩子集, 再在压缩子集上通过Davies-Bouldin[19]聚类指标进行 ε 参数自

适应过程,获得最优聚类.

2.1 RSDE

Girolami和他的同伴提出压缩集密度估计器RSDE[15-17],可以得到一个保持原始分布的压缩集,它是通过求解二次规划问题得到核密度估计器的稀疏权系表达形式. 设大规模数据集$\boldsymbol{C}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}\in R^d$,RSDE可以看作是求解QP问题:

$$\begin{aligned}&\arg\max_{\gamma}\ 2\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{p}-\gamma^{\mathrm{T}}\tilde{\boldsymbol{K}}\boldsymbol{\gamma};\\&\text{s.t. }\gamma^{\mathrm{T}}\mathbf{1}=1,\ \gamma_i\geqslant 0,\ \forall i.\end{aligned}\tag{15}$$

其中:$\tilde{\boldsymbol{K}}$是一个$n\times n$的矩阵,$\boldsymbol{p}$是一个$n\times 1$的向量,$\tilde{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\int_{R^d}K_h(\boldsymbol{x},\boldsymbol{x}_i)K_h(\boldsymbol{x},\boldsymbol{x}_j)\mathrm{d}\boldsymbol{x}$,$K_h$是一个核函数,使用Parzen[21-22]窗法每一维有数据集中的每一个样本点近似等于$\tilde{p}(\boldsymbol{x}_i)=\dfrac{1}{n}\sum\limits_{j=1}^{n}K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)$,$\mathbf{1}$是$n\times 1$的单位向量,$\gamma$是$n\times 1$的权系数向量.

求解QP问题,可以得到和密度估计的稀疏权系表现形式为

$$\hat{p}(\boldsymbol{x},h,\gamma)=\sum_{m=1}^{n_r}\gamma_m K_h(\boldsymbol{x},\boldsymbol{x}_m),\tag{16}$$

其中$\boldsymbol{x}_m\in\boldsymbol{C}_r$,$\boldsymbol{C}_r=\{\boldsymbol{x}_i|\alpha_i>0,i=1,2,\cdots,n\}$是通过RSDE得到的压缩集,表示压缩集中数据集的大小.

可以看出,RSDE虽然能够获得相对准确的压缩子集,但是压缩时间复杂度高,文献[23-24]中给出了对最小包含球(MEB)[13-14]与RSDE间的等价关系的具体解答,通过使用基于MEB的核心集算法对RSDE进行快速求解,获得压缩集,使其时间复杂度接近线性关系.

2.2 过CC-MEB快速求解压缩集

对于处理大规模数据,Tsang等[23-24]提出了一种中心约束最小球包含问题. 设数据集$\boldsymbol{C}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}\in R^d$,$\boldsymbol{C}$的最小包含球表示为MEB($\boldsymbol{C}$),即包含$\boldsymbol{C}$数据集中所有样本点的最小的包含球,$\boldsymbol{c}$为MEB($\boldsymbol{C}$)所对应的球中心向量,$\boldsymbol{R}$为最小球所对应的半径大小,则MEB在空间中的优化问题可表示为

$$\begin{aligned}&\min_{R,\boldsymbol{c}}\ R^2;\\&\text{s.t. }(\boldsymbol{x}_i-\boldsymbol{c})^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{c})\leqslant R^2.\end{aligned}\tag{17}$$

在CC-MEB问题中,对于样本点特征函数$\varphi(\boldsymbol{x}_i)$,引入一个属性项$\delta_i\in\boldsymbol{R}$,从而构造一个新的特征空间$\begin{bmatrix}\varphi(\mathrm{x}_i)\\ \delta_i\end{bmatrix}$,新特征空间的球心点为$\begin{bmatrix}\boldsymbol{c}\\ 0\end{bmatrix}$,因此MEB在新的特征空间中的优化问题可以表示为

$$\begin{aligned}&\min_{R,\boldsymbol{c}}\ R^2;\\&\text{s.t. }(\varphi(\boldsymbol{x}_i)-\boldsymbol{c})^{\mathrm{T}}(\varphi(\boldsymbol{x}_i)-\boldsymbol{c})+\delta_i^2\leqslant R^2.\end{aligned}\tag{18}$$

其对偶式可以表示为

$$\begin{aligned}&\max_{\alpha}\ \alpha^{\mathrm{T}}(\mathrm{diag}(\boldsymbol{K})+\Delta)-\alpha^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha};\\&\text{s.t. }\alpha^{\mathrm{T}}\mathbf{1}=1,\ \alpha_i\geqslant 0,\ \forall i.\end{aligned}\tag{19}$$

其中:$\Delta=[\delta_1^2,\delta_2^2,\cdots,\delta_n^2]^{\mathrm{T}}\geqslant\mathbf{0}$,$\boldsymbol{K}_{n\times n}=[k(\boldsymbol{x}_i,\boldsymbol{x}_j)]=[\varphi(\boldsymbol{x}_i)^{\mathrm{T}}\varphi(\boldsymbol{x}_j)]$. 因为$\alpha^{\mathrm{T}}\mathbf{1}=1$,所以对于任意的$\eta\in\boldsymbol{R}$,可以将式(19)化作

$$\begin{aligned}&\arg\max_{\alpha}\ \alpha^{\mathrm{T}}(\mathrm{diag}(\boldsymbol{K})+\Delta-\eta\mathbf{1})-\alpha^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha};\\&\text{s.t. }\alpha^{\mathrm{T}}\mathbf{1}=1,\ \alpha_i\geqslant 0,\ \forall i.\end{aligned}\tag{20}$$

对比式(15),令$\Delta=-\mathrm{diag}(\tilde{\boldsymbol{K}})+2\boldsymbol{p}+\eta\mathbf{1},\eta\geqslant 0$,则$\Delta\geqslant 0$,式(15)等价为

$$\begin{aligned}&\operatorname{argmax}_{\gamma}\ \gamma^{\mathrm{T}}(\Delta+\mathrm{diag}(\tilde{\boldsymbol{K}})-\eta\mathbf{1})-\gamma^{\mathrm{T}}\tilde{\boldsymbol{K}}\boldsymbol{\gamma};\\&\text{s.t. }\gamma^{\mathrm{T}}\mathbf{1}=1,\ \gamma_i\geqslant 0,\ \forall i.\end{aligned}\tag{21}$$

当$\boldsymbol{K}=\tilde{\boldsymbol{K}},\alpha=\gamma$时,式(20)与(21)相等,可以将RSDE问题看作是一个CCMEB问题,可通过基于近似MEB的快速核心技术求解[14,18,23-24].

这里把快速解RSDE的方法命名为FRSDE(fast reduced set density estimator),伪代码如下:

算法1 $[\boldsymbol{D}]=\mathrm{FRSDE}(C,\eta,\sigma,\tau)$.

输入:原始数据集C,式(20)中CCMEB的η,高斯核的窗宽σ^2,设置$\tau=10^{-6}$. η为一个很大的常数,可以通过式$\Delta=-\mathrm{diag}(\tilde{\boldsymbol{K}})+2\boldsymbol{p}+\eta\mathbf{1}$得到.

输出:压缩后的数据集C_r.

初始化$Q_0,\boldsymbol{c}_0,R_0$. Q_0为初始核心集,$\boldsymbol{c}_0$和R_0分别为初始最小球$B(\boldsymbol{c}_0,(1+\tau)R_0)$的球心和半径.

$t=1$;

//设置迭代步数初值//

While (如果训练集中的样本x在球体$B(c_t,(1+\tau)R_t)$外)

在扩展空间中找到距离c_t最远的样本点x;

设置$Q_{t+1}=Q_t\bigcup\{x\}$;

获取新的CCMEB,即获得CCMEB(Q_{t+1});

设置$c_{t+1}=c_{\mathrm{MEB}}(Q_{t+1})$, $R_{t+1}=R_{\mathrm{MEB}}(Q_{t+1})$;

End

$C_r=\{x|\gamma_i>0,i=1,2,\cdots,n_r\}$

//根据式(21)快速求得RSDE压缩集//

Return C_r

这里τ的取值越小,获得的压缩集越精确,所需要的时间越长;τ的取值越大,获得的压缩集精确度越小,但所需的时间减少. 为了保证压缩集的精确度和压缩时间的合理性,文献[20]给出$\tau=10^{-6}$.

2.3 在压缩集上的引力同步聚类过程

上面一节中,获得了保持原始数据集样本结构的压缩集,引力同步的时间复杂度取决于压缩集的大小,降低了时间复杂度. 在压缩集中,首先为ε设定

初始值,对于给定的压缩集C_r,取C_r中的样本与它的k邻近样本点距离的平均值为ε的初值. 在迭代过程中,需要不断增大ε的值,直到局部同步,$\Delta\varepsilon$为样本点与$k+1$个邻近点距离的平均值减去样本点与k个邻近点距离的平均值,设为步长. 利用Davies-Bouldin[19]聚类指标(DB)去自动寻优相应的ε值.

Davies-Bouldin(DB)表示的是同一类中样本的紧密程度与不同类中样本的分散程度的一个函数. DB的定义为

$$\mathrm{DB}=\frac{1}{m}\sum_{i=1}^{m}R_i. \tag{22}$$

其中: $R_i=\max\limits_{j=1,2,\cdots,m,j\neq i}R_{i,j}, i=1,2,\cdots,m$, m为聚类的个数, R_{ij}为聚类之间的相似性指标. $R_{ij}=(s_i+s_j)/g_{ij}$, g_{ij}为聚类之间的差异性,定义为

$$g_{ij}=\|\boldsymbol{w}_i-\boldsymbol{w}_j\|_q;$$

s_i为聚类C_i的散度测量,定义为

$$s_i=\Big(\frac{1}{n_i}\sum_{\boldsymbol{v}\in C_i}||\boldsymbol{v}-\boldsymbol{w}_i||^r\Big)^{1/r},$$

其中n_i表示C_i中向量的个数. DB指标表示的是聚类结果中这一类与其他类之间相似性的平均值,相似性越小即聚类同一类紧密且不同类分散性好的聚类,所以用DB指标选择最小的值对应的ε,获得最终的同步聚类结果. 这里把上述同步聚类的方法命名为引力同步聚类(GSC),伪代码如下:

算法2　$\boldsymbol{M}=\mathrm{GSC}(C_i)$.

输入:压缩集C_i;

输出:聚类结果$\boldsymbol{M}$.

初始化ε,取样本集C_i中每个样本与其k个邻近样本距离的平均值 $\Delta\varepsilon=$样本点与$k+1$个邻近样本距离的平均值$-$与k个邻近样本距离的平均值;

//求出增大的步长//

$l=0$,全局同步变量$=0$;

While (全局同步变量$=0$)

//$[M^l]=\mathrm{GravitationClustering}(C_r,\varepsilon^l)$//

输入:压缩后的C_i,增加了步长后的ε^l;

输出:ε^l在M^l下得到的聚类结果.

$r_s=0$

While ($r_s\leqslant 1-10^{-3}$)

for (每个样本p属于C_i)

求每个样本的ε近邻

根据式(13),计算下一时刻的样本点的位置;

//计算样本点的局部参量//

$$r_p=\frac{1}{|n_r b_\varepsilon(p)|}\sum_{r\in n_r b_\varepsilon(p)}\mathrm{e}^{-||r-p||^2}$$

$r_s=r_s+r_p$

//累加总序列参量//

End for

End While

Obtain M^l.clustersize

//获得聚类的个数//

Obtain M^l.cluster

//获得最终聚类的结果//

Obtain M^l

2.4　对$C-C_r$的数据集聚类

在压缩集C_r上使用引力同步聚类以后,本文使用算法3 (RSC)对剩余的$C-C_r$的大规模数据集进行聚类,RSC可以完成剩余样本的聚类,并且确定孤立的类和噪声点. 对于一些很小的孤立团体,在被压缩后,很可能压缩集中并不包含这些点,但这些很小的团体可能包含着很重要的信息,在实际应用中,这些点并不可以忽视. 为了确定这些点,RSC把这些小团体设为独立类,并划分出干扰异常的噪声点. 算法过程是:首先把原始数据集逐一对比分配到每个数据集M_{c}^i中去,对于不属于任意数据集M_{c}^i的数据单独分为一类并记为M_{s}^i,再计算每个M_{s}^i之间最小距离与DB寻优中获得的最小值所对应的ε比较,小于ε的化为一类M_{is}^i,即为孤立类,剩余的M_{s}^i为噪声点,时间复杂度为$O(n^2)$. RSC算法的伪代码如下:

算法3　$[M_c,M_s]=\mathrm{RSC}(\boldsymbol{C},\boldsymbol{M},\varepsilon_r)$.

输入:原始数据集$\boldsymbol{C}$,在压缩集上同步聚类的聚类结果$\boldsymbol{M}$, $M=M^1\bigcup M^2\bigcup\cdots\bigcup M^m$, m为在压缩集上的聚类数, M^i为在压缩集C_r上的第i个聚类, ε_r为最小DB值所对应的ε值;

输出:聚类结果M_c,噪声点M_s,孤立类M_is.

$U=C-C_r$;//剩余的需要完成聚类的数据集//

$\varepsilon^0=\mathrm{knn}(k,U)$;//初始化$\varepsilon$值//

$\Delta\varepsilon=\mathrm{knn}(k+1,C_r)-\mathrm{knn}(k,U)$

//计算出ε的步长//

为原本样本加类标签

$l=0$

While ($\varepsilon^l\leqslant\varepsilon_r$)

$M^1(l)=M^1,M^2,\cdots,M^m(l)=M^m,U(l)=U$

Repeat

求每个样本x的ε^l最近邻U'

for ($k=1:m$)

$U'=x\in C_r,y\in C-C_r|\mathrm{dist}(y,x)\leqslant\varepsilon^l\}$

其中$\mathrm{dist}(y,x)=\sqrt{\sum\limits_{i=1}^{n-n_r}\sum\limits_{j=1}^{n_r}(y(i)-x(j))^2}$

计算出每个U'与M^k的交集个数,并将此样本点放入交集个数最大的M^k中;$U=U-x$

End for

//将原始数据集分配到每一类M^k中,直到U等于0,或这不能再分为一类//

如果$U \neq 0$,将每个样本设为一个类

$p = \text{size}(U, 2)$;

$p_{\text{temp}} = 0$

While $(p \neq p_{\text{temp}})$

$p_{\text{temp}} = p$

$\text{dist}(M_s^i, M_s^j) = \min\limits_{l,k \leqslant p, l \neq k} (M_s^l, M_s^k)$

//找到任意两类之间的最近距离//

If $(\text{dist}(M_s^i, M_s^j) \leqslant \varepsilon^l)$

合并M_s^i和M_s^j为一类记为M_{is}

$p = p - 1$;

End

$l = l + 1; \varepsilon^l = \varepsilon^l + \Delta\varepsilon$

End

将M_s^i中样本数为1的作为噪声点

Return M_c, M_s, M_{is}

2.5 LSCGS时间复杂度分析

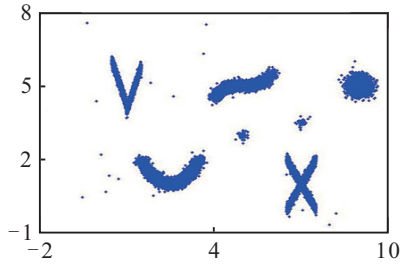
LSCGS算法的时间复杂度由FRSDE算法的时间复杂度、GSC算法的时间复杂度、ε自适应寻优的时间复杂度以及RSC算法的时间复杂度组成. 其中: FRSDE为数据压缩算法,GSC为引力同步聚类算法,RSC为对剩余数据集聚类的算法. FRSDE的时间复杂度接近线性,为$O(n)$,n为原始数据集样本的大小;GSC的时间复杂度是对压缩集样本中每一维数据进行迭代,时间复杂度为$O(T \times {n_t}^2)$,其中n_t为FRSDE算法产生的样本集C_r的数据集大小,T为迭代次数,一般取$5 \leqslant T \leqslant 20$. RSC的时间复杂度一般少于$O(n^2)$. 综上,LSCGS算法的总时间复杂度是$O(n + L \times T \times {n_t}^2 + n^2)$,$L$为$\varepsilon$的搜索空间,$n_t$远远小于$n$. 现有的Sync算法是对数据集中每一维数据进行迭代运算,算法时间复杂度为$O(T \cdot n^2)$,T为迭代次数,一般取$(5 \leqslant T \leqslant 20)$.

3 实 验

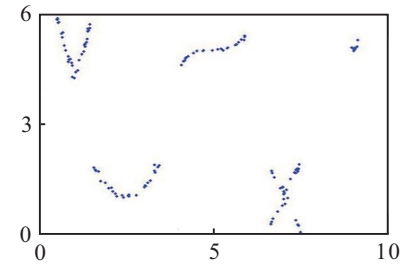
本文使用不同规模的人造数据集、图像分割数据集及部分真实数据集对LSCGS在大规模数据集上的聚类效果进行判断,采用NMI[25]和RandIndex[26]两种指标对算法进行评估,指标越高,说明聚类效果越好. 实验所采用的环境为: Intel Corei-2130 3.4 GHz CPU; 4.0 GB RAM, Windows7; Matlab 7.10.0. 在实验ε的初始化中,相邻样本个数k值都设为3,高斯函数中的$2\sigma^2$设定为样本数据集的2范数的平方,DB指标里的r、q设为2.

3.1 人造不同规模数据集

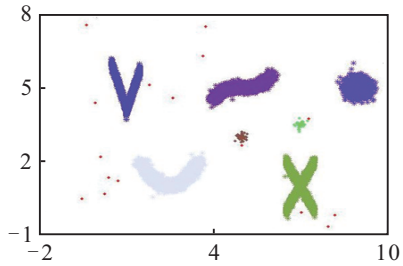
使用LSCGS算法分别对构造的9组不同规模、大小为2维的人造数据集进行聚类,这里ε设定为0.6. 数据集由7种不同大小的人造数据集和一些人造噪声点构成. 为了验证LSCGS对异常点和孤立点的划分,在人造数据集中加了20个噪声点样本,图2(a)为32 100的样本集,图2(b)为样本压缩集,压缩集大小为125,图2(c)为聚类结果. 表1给出的是在9组不同规模样本的人造数据集上,每组运行5次获得的平均值,“-”表示在Sync算法中时间无法容忍.



(a) 32 100样本集



(b) 压缩后的样本集



(c) 聚类结果

图 2 引力同步聚类过程

表 1 不同数据集中的算法性能指标

数据集大小	LSCGS/(t/s)				Sync/(t/s)
	FRSDE	GSC同步聚类	RSC	总时间	
822	4.09	104.57	0.57	109.23	3 987.9
1 624	7.19	165.84	1.81	174.84	24 798.1
3 228	15.12	291.23	7.53	313.88	–
8 040	26.17	267.81	38.45	332.43	–
16 060	32.15	349.95	159.52	541.62	–
32 100	64.33	466.00	425.45	955.79	–
64 180	120.35	580.05	2 668.42	3 368.82	–
80 220	150.94	563.80	2 562.13	3 276.87	–
112 280	175.55	609.33	8 108.61	8 893.59	–

为了检验算法可以清楚划分出孤立类和噪声点的优点,分别与其他经典算法(K-means[1], FCM[3],

SC[27]算法)进行对比实验,实验所用的数据集是样本个数16 060样本集,聚类效果见图3.

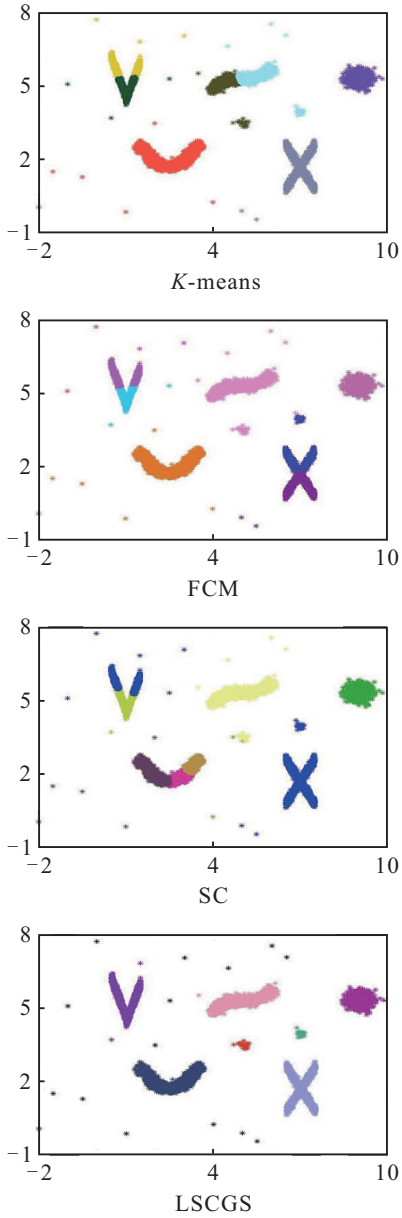


图 3　不同算法对数据集的聚类效果

由上面的实验结果可以得到以下结论:

1) 从图2(b)可以看出,压缩后的数据集基本保持了原始数据集的结构特征,对于噪声点和孤立点,压缩时很可能是被忽略的,所以算法3对剩余数据集的聚类很好地还原了原始数据集中的孤立类以及噪声点的标记.

2) 对比图2(a)和图2(b)的数据集,可以发现压缩集的样本远远少于原始数据集,这使得引力同步聚类可以顺利地进行下去,大大减少了聚类的时间,降低了时间复杂度. 对比表1可以看出,当样本数大于1 600时,Sync算法的时间已经高达了10^5以上,对于大规模样本数据集根本无法容忍,可见LSCGS的执行效率明显优于Sync.

3) 从图3可以看出: K-means算法和FCM算法有4个聚类未被正确识别,其中两个聚类中的一部分分别被分成了另外两类,无法识别孤立类与噪声点; SC算法有一个类被分成了3类,一类被分成了两类,不能成功地分出小团体的孤立类和噪声点. 由此可以看出,传统聚类算法难以处理噪声点与孤立类,且依赖于人工经验预先设下类别数. 而本文算法可以正确识别出7个类,对于孤立类也清楚地识别,20个噪声点中仅有5个点被分到相邻的聚类中(图中黑色点为噪声点).

3.2　图像分割实验

本文选取了4幅图像(分辨率均为320像素×320像素)进行图像分割实验,分别是Horse, Elephant,Sky, Grassland(图像下载于 http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench). 由于图片较大,为方便展示,文章中对示意图进行了缩放,如图4所示.

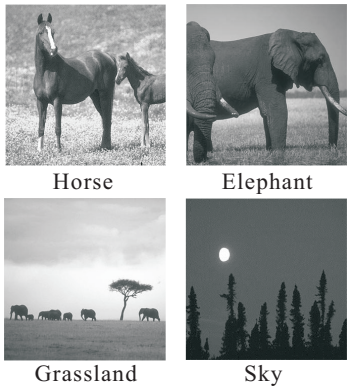


图 4　图像分割实验的图像

为了说明LSCGS的有效性,分别与K-means算法和FCM算法的聚类结果进行对比. 由于Sync的时间开销太大,无法在原图上进行实验,4幅图的类别数分别为2、2、3、3.

图5和图6分别为本文算法和FCM算法的图像分割效果, K-means与FCM算法的分割效果相近,限于篇幅, K-means的分割效果不再给出. 为了可以定量地分析图像分割效果,这里采用Rand Index指标进行评估. 采用该指标进行评估之前首先对像素点划分类标,方法见文献[28]. LSCGS分别在4幅图片上运行5次,求得平均Rand Index指标记录见表2.

表 2　3类算法在4幅图片上的Rand Index

数据集	LSCGS	FCM	K-means
Horse	0.914 3	0.919 1	0.919 5
Elephant	0.811 0	0.899 1	0.891 1
Grassland	0.995 5	0.732 7	0.733 0
Sky	0.916 1	0.734 7	0.996 9

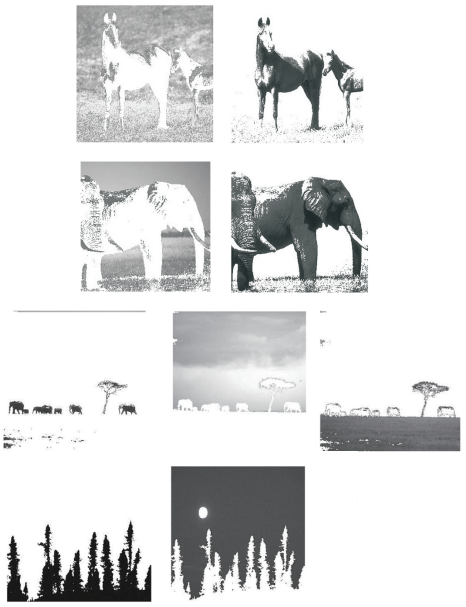


图5 LSCGS算法在图像上的分割效果

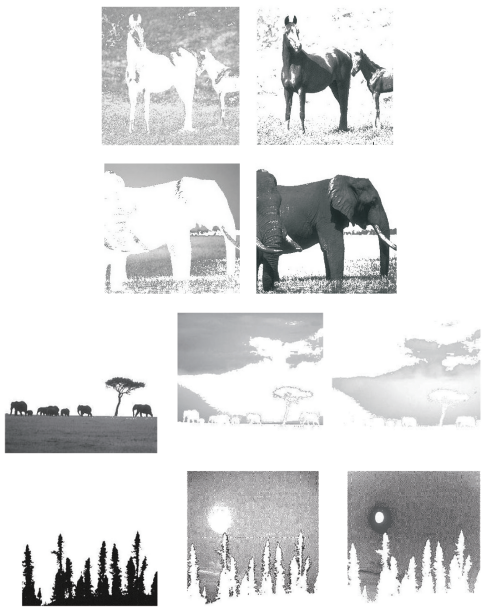


图6 FCM算法在图像上的分割效果

从实验结果可以看出: 在Horse和Elephant中, 3种算法的指标相近, 都能有效地分割出目标物体; 在Sky中, LSCGS明显优于FCM算法, 与K-means算法聚类指标相近; 在Grassland中, LSCGS明显优于其他两种算法. 但FCM和K-means在执行前需要预设类别数, 而LSCGS算法可以自动寻优到类别数, 更具有实用性.

3.3 真实数据集

为了验证本文算法在大规模数据集上的性能, 使用3个大规模UCI数据集Landsat、 Segment和Brainweb. 为充分验证本文所提算法针对大规模数据集时间复杂度下降的优势, 对Landsat和Segment进行扩充, 对每个样本数据增加偏移分量. Landsat增加均值为0, 方差为1的正态分布的偏移分量, 使数据扩充为44 350; Segment增加均值为0, 方差为1的正态分布的偏移分量, 使数据扩充为23 100. 3类算法在各个数据上分别运行10次, 求得的聚类结果和平均NMI、 RI值如表3所示. 在FCM算法和K-means算法执行之前需要设定聚类数, 3个样本的类别数分别为6、 7、 2. 观察表3可以得到: 本文所提出的算法在3个数据集上都可以得到较为准确的指标, 与经典算法相比, 指标略优于FCM和K-means算法. 算法LSCGS有聚类数大于数据本身的情况, 可能是一些噪声数据构成的小集合, 被算法误认为一类.

表3 3类算法在各个数据上的指标

算法		Landsat	Segment	Brainweb
LSCGS	NMI	0.572 8	0.507 1	0.719 4
	RandIndex	0.495 6	0.408 8	0.816 6
	类别数	7	9	2
FCM	NMI	0.418 8	0.367 9	0.719 3
	RandIndex	0.173 6	0.391 7	0.815 7
K-means	NMI	0.543 8	0.393 6	0.718 5
	RandIndex	0.442 5	0.404 3	0.815 0

4 结 论

本文根据万有引力定律和牛顿第二定律与自然界的同步现象, 提出了大规模数据集引力同步聚类算法LSCGS. LSCGS算法不会受到数据集形状、大小和密度的影响, 且自动寻优最佳聚类数, 优于传统K-means、FCM、SC等算法对人工经验的依赖程度, 同时又能够保证算法的准确度.

本文所提算法的主要优点有: 1) 相对于传统的Sync算法, LSCGS可以处理大规模数据集, 对于Sync根本没办法处理的图片样本, 本文算法可以准确分割出所需要的类; 2) 本文算法从一个新的角度, 即万有引力的角度重新定义了同步的过程, 使同步聚类算法的模型更加多样化, 给同步聚类算法一个全新的解释; 3) LSCGS算法对噪声点和孤立点的处理和划分明显好于K-means、FCM、SC等算法, 可以更加精确地划分出所需要的所有类.

然而, LSCGS算法所提出的模型依然需要迭代每个样本的每个分量, 在处理高维数据时, 需要较多的时间, 如何克服这一困难, 是之后需要探讨的课题.

参考文献(References)

[1] Jain A K. Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2015, 31(8): 651-666.

[2] Schnitzer D, Flexer A. The unbalancing effect of hubs on K-medoids clustering in high-dimensional spaces[C]. 2015 Int Joint Conf on Neural Networks(IJCNN).

Killarney: IEEE, 2015: 1-8.

[3] 杨飞, 朱志祥. 基于特征和空间信息的核模糊C-均值聚类算法[J]. 电子科技, 2016(2): 16-19.
(Yang F, Zhu Z X. Kernelized fuzzy C-means clustering algorithm based on features and spatial information[J]. Electronic Science and Technology, 2016(2): 16-19.)

[4] Lattanzi M, Del Giudice G, Rappuoli R. Variational bayesian expectation maximization for radar map estimation[J]. IEEE Trans on Signal Processing, 2016, 64(6): 1391-1404.

[5] Torcini A, Luccioli S, Bonifazi P, et al. Clique of functional hubs orchestrates population bursts in developmentally regulated neural networks[C]. APS Meeting Abstracts, 2014, 10(9): 1679-1689.

[6] 齐兴斌, 赵丽, 李雪梅. 基于BIRCH聚类加速的彩色图像增强算法[J]. 计算机测量与控制, 2016, 24(4): 137-140.
(Qi X B, Zhao L, Li X M. A color image enhancement algorithm based on BIRCH cluster acceleration[J]. Computer Measurement and Control, 2016, 24(4): 137-140.)

[7] Cordova I, Moh T S. DBSCAN on resilient distributed datasets[C]. 2015 Int Conf on High Performance Computing and Simulation(HPCS). Amsterdam: IEEE, 2015.

[8] Brusco M J, Steinley D. Affinity propagation and uncapacitated facility location problems[J]. J of Classification, 2015, 32(3): 1-38.

[9] 黄霞, 徐灿, 孙玉庭, 等. 耦合振子系统的多稳态同步分析[J]. 物理学报, 2015, 64(17): 53-63.
(Huang X, Xu C, Sun Y T, et al. Multiple synchronous states in a ring of coupled phase oscillators[J]. Acta Physica Sinica, 2015, 64(17): 53-63.)

[10] Moreno Y, Pacheco A F. Synchronization of kuramo to oscillators in scale—Free networks[J]. Euro Physics Letters, 2004, 68(4): 603-609.

[11] Bhm C, Plant C, Shao J, et al. Clustering by synchronization[C]. Proc of Acm Sigkdd Conf on Knowledge Discovery and Data Mining. Washington: 2010: 583-592.

[12] Liu C, Hey R, Wang W. K-AP clustering algorithm for large scale dataset[C]. First Int Workshop on Complexity and Data Mining. Nanjing: IEEE, 2011: 87-89.

[13] Nguyen K, Le T, Lai V, et al. Least square support vector machine for large-scale dataset[C]. Int Joint Conf on Neural Networks. Killarney: IEEE, 2015: 1-8.

[14] Shi Y Z, Wang S T, Wang J, et al. Fast classification for nonstationary large scale data sets using minimal enclosing ball[J]. control and Decision, 2013, 28(7): 1065-1072.

[15] Botev Z I, Kroese D P. Kernel density estimation via diffusion[J]. Annals of Statistics, 2010, 38(5): 2916-2957.

[16] Wang S, Wang J, Chung F L. Kernel density estimation, kernel methods, and fast learning in large data sets[J]. IEEE Trans on Cybernetics, 2014, 44(1): 1-20.

[17] Wang J, Deng Z, Wang S, et al. Training generalized feedforword kernelized neural networks on very large datasets for regression using minimal-enclosing-ball approximation[J]. Proc of ELM-2014, 2015(1): 203-214.

[18] 钱鹏江, 王士同, 邓赵红, 等. 基于最小包含球的大数据集快速谱聚类算法[J]. 电子学报, 2010, 38(9): 2035-2041.
(Qian P J, Wang S T, Deng Z H, et al. Fast spectral clustering for large data sets using minimal enclosing ball[J]. Acta Electronica Sinica, 2010, 38(9): 2035-2041.)

[19] Coelho G P, Barbante C C, Boccato L, et al. Automatic feature selection for BCI: An analysis using the davies-bouldin index and extreme learning machines[C]. 2012 Int Joint Conf on Neural Networks(IJCNN). Brisbane: IEEE, 2012: 1-8.

[20] Ying W, Chung F L, Wang S. Scaling up synchronization-inspired partitioning clustering[J]. IEEE Trans on Knowledge and Data Engineering, 2014, 26(8): 2045-2057.

[21] 钱鹏江, 王士同, 邓赵红. 基于稀疏Parzen窗密度估计的快速自适应相似度聚类方法[J]. 自动化学报, 2011, 37(2): 179-187.
(Qian P J, Wang S T, Deng Z H. Fast adaptive similarity-based clustering using sparse parzen window density estimation[J]. Acta Automatica Sinica, 2011, 37(2): 179-187.)

[22] 应文豪. 基于Parzen Window估计的分类与聚类方法及应用研究[D]. 无锡: 江南大学数字媒体学院, 2013.
(Ying W H. Contribution to Classification and Clustering Methods based on parzen window density estimation[D]. Wuxi: School of Digital Media, Jiangnan University, 2013.)

[23] Tsang I W, Kwok J T, Cheung P M. Core vector machines: Fast SVM training on very large data sets[J]. J of Machine Learning Research, 2005, 6(2): 363-392.

[24] Tsang I W, Kwok J T, Zurada J M. Generalized core vector machines.[J]. IEEE Trans on Neural Networks, 2006, 17(5): 1126-1140.

[25] Shang F, Jiao L C, Shi J, et al. Fast density-weighted low-rank approximation spectral clustering[J]. Data Mining and Knowledge Discovery, 2011, 23(2): 345-378.

[26] Jiang Y Z, Deng Z H, Wang J, et al. Transfer generalized fuzzy c-means clustering algorithm with improved fuzzy partitions by leveraging knowledge[J]. Moshi Shibie Yu Rengong Zhineng/pattern Recognition and Artificial Intelligence, 2013, 26(10): 975-984.

[27] Ding L, Gonzalez-Longatt F M, Wall P, et al. Two-step spectral clustering controlled islanding algorithm[J]. IEEE Trans on Power Systems, 2013, 28(1): 75-84.

[28] 应文豪, 许敏, 王士同, 等. 在大规模数据集上进行快速自适应同步聚类[J]. 计算机研究与发展, 2014, 51(4): 707-720.
(Ying W H, Xu M, Wang S T, et al. Fast adaptive clustering by synchronization on large scale datasets[J]. J of Computer Research and Development, 2014, 51(4): 707-720.)

（责任编辑：齐 霁）