动态阈值粗糙C均值算法

王 丹 吴孟达
(国防科技大学理学院数学与系统科学系 长沙 410073)

摘 要 粗糙C均值算法中3个参数 w_l,w_u,ε 的选择是算法应用的关键问题。针对粗糙C均值算法中反映类间叠加程度的参数 ε 的设定,提出一种动态自适应调整阈值 ε 的粗糙C均值算法,该算法根据"类-类"间距离与"对象-类"间距离,对每一个待聚类对象动态设定阈值 ε。两组人工数据和图像数据的实验表明,该算法具有较好的适应性和聚类效果。

关键词 C均值聚类,粗糙集,粗糙C均值聚类
中图法分类号 TP311 **文献标识码** A

Dynamic Threshold Rough C-Means Algorithm

WANG Dan WU Meng-da
(Department of Mathematics and Systems Science, College of Science, National University of Defense Technology, Changsha 410073, China)

Abstract Selection of parameters w_l, w_u, ε plays an important role in rough C-Means algorithm. In this paper, a dynamic threshold rough C-Means algorithm was proposed to self-adaptive adjusting threshold ε that reflects the superposition between classes. This algorithm computes a threshold for every object on the basis of class interval and the distance between class and object. The better effect can be testified by two synthetic data and image data experiments.

Keywords C-Means clustering, Rough sets, Rough C-Means clustering

1 引言

粗糙集理论[1]是1982年由波兰数学家Z. Pawlak提出的一种处理含糊和不精确问题的数学工具。粗糙集理论不需要预先给定某些特征和属性的数量描述,直接从给定问题的客观描述集合出发,通过不可分辨关系和不可分辨类确定给定问题的近似域,粗糙集的主要思想是把事物分成肯定的、否定的和不能肯定的3个集合,分别用下近似、负域和边界表示,从而找出该问题的内在规律。近年来,粗糙集在数据库、人工智能、图像处理、模式识别、决策分析、知识发现、专家系统等领域取得了较快的发展,与模糊数学等理论一起成为处理不确定信息的重要工具。

众所周知,对象的刻画往往通过对象的特征提取,以一组特征向量来描述对象,而聚类是将这样的 N 个对象划分为 c 类,划分的标准是使得相同的类内的对象相似性达到最大,不同类之间的相似性达到最小,而对象相似性一般通过特征向量的距离来刻画,所以聚类的目标也就是使得类内距离最小,类间距离达到最大。经典C均值算法是以类内距离达到最小为目标,通过不断调整类心来进行聚类,在经典C均值算法中类心取的是质心,分类原则是划分每个对象到离其最近的一个类。1981年Bezdek[2]改进了经典C均值算法,提出模糊C-均值(Fuzzy C-Means)聚类算法(FCM),此算法以最小类内平方误差和为聚类准则,与C均值聚类算法不同之处在于不将样本分成分明子集,而是计算每个样本属于各模糊子集(聚类)的隶属度,通过目标函数极小化来获得最优的聚类。该算法提出后在图像的分割、压缩、识别等领域得到了广泛的应用。FCM算法以模糊隶属度函数来刻画对象归属的模糊性,实际上是一种精确的模糊性刻画方法,而实际中,往往很难做到这种精确性,比如某个对象属于一个类的程度为0.49,属于另一个类的程度为0.51,按照模糊C均值算法,这时候此对象将划分为第二个类,而在现实中,这样的情况往往是很难分清的。产生这种现象的根本原因是模糊隶属度刻画模糊性实际上仍然是一种精确的刻画方法。2002年,Lingras等[3]人提出了粗糙C均值算法,其基本思想是承认聚类边界的存在,通过把粗糙集理论中上、下近似引进到C均值算法中,在一个类的下近似中的对象肯定是属于这个类的,而位于边界(上近似与下近似的差集)的对象,在粗糙C均值算法中认为由于信息的缺乏而不能明确判断,也就是说,在粗糙C均值算法中承认了类之间存在重叠。在模糊C均值算法中,模糊隶属度函数的引入实际上也承认了一定程度的重叠,但在处理这部分重叠时,模糊C均值是用更加精确的方式来刻画模糊性。Lingras粗糙C均值算法已经成功应用到很多实际案例中。在此基础上的相关研究工作也开展了很多。比如Mitra[5]通过遗传算法来改进粗糙C均值算法的参数选择问题,Malyszko[6]通过粗糙熵的引入来优化粗糙C均值算法的参数,Peters[4]对算法进行了更加细致的设计。Mitra[7]提出了粗糙模糊C均值算法,进一步扩展了粗糙C均值算法。王丹等[9]在2007年将粗糙集上、下近似的概念引入模糊C均值算法,也提出一种粗糙模糊C均值算法改进了模糊C均值算法B郑超、苗夺谦等[10]通过引入每个对象周围的密度分布构

到稿日期:2010-04-28 返修日期:2010-08-05 本文受国家自然科学基金(60872152)资助。
王 丹(1981-),男,博士生,讲师,主要研究方向为不确定信息处理、图像处理;吴孟达(1956-),男,教授,主要研究方向为不确定信息处理。

造了一种新的粗糙C均值算法B邵锐等[11]将粗糙C均值算法应用到了图像分割中。

本文第2节介绍经典C均值算法、模糊C均值算法和粗糙C均值算法；第3节提出一种动态阈值粗糙C均值算法；第4节基于粗糙C均值聚类改进了聚类有效性指标DB指数；第5节以人工数据和图像分割中的聚类为例对比了几种方法的性能。

2 聚类算法

2.1 经典C均值算法(HCM)

经典C均值算法的基本思想是选定 c 个类和选取 c 个初始聚类中心，按最小距离原则将各个对象分配到 c 个类中的某一个，之后通过不断调整各对象的类别，最终使得各对象到其归属类别中心的距离平方和最小。算法步骤如下：

(1)任选 c 个初始聚类中心 $v_i, i=1,2,\cdots,c$。

(2)将每个待分类对象 $x_j, j=1,2,\cdots,N$ 按照最小距离原则划分到 c 个类中的某一个类，即：如果 $d_{ij}=\min\limits_{1\leqslant k\leqslant c} d(x_j, v_k), j=1,2,\cdots,N$，则 $x_j\in U_i$，其中，$d(x_j,v_k)=\|x_j-v_k\|$，U_i 为对应类心 v_i 的类。

(3)重新计算类心

$$v_k=\frac{\sum\limits_{x_j\in U_k} x_j}{n_k} \tag{1}$$

式中，n_k 为 U_k 类中元素的个数，即 $n_k=\mathrm{card}(U_k)$。

(4)重复步骤(2)－(4)直到收敛，收敛的条件可以取对象不再产生变化，也可以取距离和最小。

2.2 模糊C均值算法(FCM)

令 $X=\{x_1,x_2,\cdots,x_n\}\subset R^p$ 为待分类对象，模糊C-均值算法的目标函数为：

$$J_m(U,V)=\sum_{j=1}^{N}\sum_{i=1}^{c}u_{ij}^m d_{ij}^2$$

式中，$V=\{v_1,v_2,\cdots,v_c\}$，v_i 为第 i 类的聚类中心，权重 $m\in(1,\infty)$，$d_{ij}^2=\|x_j-v_i\|^2$。

约束条件为：

$$(1)u_{ij}\in[0,1];(2)0<\sum_{j=1}^{n}u_{ij}<N;(3)\sum_{i=1}^{c}u_{ij}=1$$

运用拉格朗日乘数法，可以得到 v_i, u_{ij} 的迭代计算公式：

$$v_i=\sum_{j=1}^{N}u_{ij}^m x_j/\sum_{j=1}^{N}u_{ij}^m, i=1,2,\cdots,c$$

$$u_{ij}=\frac{1}{\sum\limits_{k=1}^{c}\left(\frac{d_{ij}^2}{d_{kj}^2}\right)^{\frac{1}{m-1}}} \tag{2}$$

模糊C均值算法的步骤跟经典C均值算法类似，只是需要更新对象隶属程度 u_{ij}。

2.3 粗糙C均值算法(RCM)

在粗糙C均值算法中，上、下近似集被引入到聚类中，聚类中的每个对象遵循以下原则：

(1)每个对象最多属于一个类的下近似。

(2)如果一个对象属于某一个类的下近似，一定属于这个类的上近似。

(3)如果一个对象不属于任一个类的下近似，那么一定属于至少两个类的上近似。

从上面原则可以看出粗糙C均值算法允许存在一定的聚类重叠。当然上面的3个条件并不是独立的。若再从类的角度来看这个问题，对于聚类后的每一个类，同样存在3种情况：(1)只有下近似元素，(2)只有上近似元素，(3)既有下近似元素也有上近似元素。根据这些原则，粗糙C均值算法的类心计算可调整为：

$$v_i=\begin{cases} w_l\dfrac{\sum\limits_{x_j\in\underline{BU}_i}x_j}{|\underline{BU}_i|}+w_u\dfrac{\sum\limits_{x_j\in\overline{BU}_i-\underline{BU}_i}x_j}{|\overline{BU}_i-\underline{BU}_i|}, & \underline{BU}_i\neq\phi,\overline{BU}_i-\underline{BU}_i\neq\phi \\ \dfrac{\sum\limits_{x_j\in\underline{BU}_i}x_j}{|\underline{BU}_i|}, & \underline{BU}_i\neq\phi,\overline{BU}_i-\underline{BU}_i=\phi \\ \dfrac{\sum\limits_{x_j\in\overline{BU}_i-\underline{BU}_i}x_j}{|\overline{BU}_i-\underline{BU}_i|}, & \underline{BU}_i=\phi,\overline{BU}_i-\underline{BU}_i\neq\phi \end{cases} \tag{3}$$

式中，$\underline{BU}_i, \overline{BU}_i$ 分别表示类 U_i 的下近似和上近似，$|\cdot|$表示集合的基数，w_l, w_u 分别为下近似、上近似所占的权重，且 $w_l+w_u=1$，一般情况下 $0.5\leqslant w_l<1$，即在决定类的类心时下近似起的作用应该更大一些。粗糙C均值算法的步骤如下：

(1)选定 c 个初始聚类中心 $v_i, i=1,2,\cdots,c$。

(2)决策每一个对象 x_j 所属的分类，决策规则如下：

a. 计算 $d_{kj}, k=1,2,\cdots,c$，且 $d_{ij}=\min\limits_{1\leqslant k\leqslant c}d_{kj}$；

b. 若存在类 U_l，使得 d_{lj} 满足：$|d_{lj}-d_{ij}|<\varepsilon$，则 $x_j\in\overline{BU}_i$ 且 $x_j\in\overline{BU}_l$，否则 $x_j\in\underline{BU}_i$；

(3)按照式(3)更新类心。

(4)重复步骤(2)－(3)，直到收敛，这里收敛的条件可以取类心不再发生变化或取某个目标达到最优，比如构造类内距离与类间距离的某个函数。

从式(3)可以看出，若 $\underline{BU}_i=\overline{BU}_i$，粗糙C均值算法则转化为经典C均值算法。在粗糙C均值算法聚类过程中，有3个参数 w_l, w_u, ε 要事先给出来，若考虑 $w_l+w_u=1$，则可认为只有两个参数，w_l, w_u 参数在聚类过程中决定不同类型的对象的重要程度，ε 参数决定聚类过程中类与类之间重叠的程度，ε 取得越大，边界就越大，类与类之间交叠也得越多。粗糙C均值算法也是对模糊C均值算法的补充，它处理边界对象不再像模糊C均值算法一样精确地决定对象的归属，而是对于隶属程度相差不大的对象，承认其边界性或者称为不可分辨性。认为需要其它进一步的信息才能区分。

3 动态阈值粗糙C均值算法

Peters[4]指出在粗糙C均值算法中，3个参数 w_l, w_u, ε 的选择是目前没有得到很好解决的问题。Mitra在文献[5]中指出参数的选择是粗糙C均值算法最大的挑战。Mitra通过构造适应度函数用遗传算法来选择最优的 w_l, w_u，Maly-szko[6]通过粗糙熵的引入来优化 w_l, w_u 的选择，构造 w_l, w_u 的自适应选择是目前主要的研究方向。从某种程度上来说，w_l, w_u 决定了聚类的类心精度和收敛速度，而 ε 决定了聚类的准确度，所以对 ε 选择的研究是有必要的。但关于阈值 ε 的选择目前的研究工作较少，一般对 ε 的选择是根据边界大小的先验信息给定一个值，缺乏有效的自适应选择。本文提出一种变阈值粗糙C均值算法(DRCM)，主要讨论阈值 ε 的选择问题。

在粗糙C均值算法中，判断对象 x_j 分类的决策规则是基于此对象与不同类心之间的距离，而且在第2节中使用线性距离 $|d_{lj}-d_{ij}|<\varepsilon$ 来界定对象的归属，Peters[4]指出了用此距离是不合适的，如图1所示。

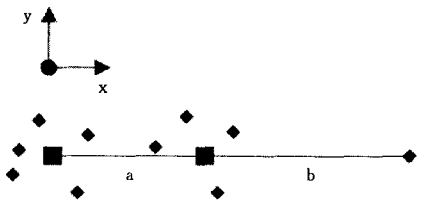


图 1　线性距离反例(方形为类心,菱形为对象)

在图 1 中,假设最右边对象为 A,左边类心代表类 1,中间类心代表类 2,若两个类的类心距离 $d_{1A}-d_{2A}=a<\varepsilon$,则对象 A 将会判定给两个对象的上近似,而似乎对象 A 判定给类 1 的上近似并不合适,所以 Peters 建议此界定标准改为 $\frac{d_{lj}}{d_{ij}}<\varepsilon$ 更加合适一些。

上述的讨论实际上还揭示了这样一个潜在的事实,一个对象是否属于某两个类的上近似不只与此对象与两个类心的距离有关,同时还跟类心之间的距离有关,这在当前的研究文献中并没有提到这一点。图 1 中的反例成立的条件是 $a<\varepsilon$,也就是说 1 类和 2 类是靠得很近的,若 ε 足够小,也就是说本身两个类的可区分性很小,则判断对象属于两个类的上近似也是理所当然的。那么根据这个思想,对于一个对象的归属可以得到这样的结论:若两个类间距相比对象离类的距离越小,也就是说对象离两个类都很远,那么这个对象离两个类的距离即使相差大一点,仍然可以认为这个对象属于这两个类的上近似,而对于类间距相比对象离类的距离越大,也就是说两个类离得越远,这时候对象属于两个类的上近似的要求就应该更加严格一些,即对象距两个类的距离差应该更小一些,这个结论实际上定义了以下准则:

(1)相比对象离类的距离,两个类离得越近,其边界越大;

(2)相比对象离类的距离,两个类离得越远,其边界就越小。

上述两个准则给出了阈值 ε 变化的规则。

阈值 ε 调整准则:

沿用上面定义的符号,记 $dist=\min\{d_{lj},d_{ij}\}$,定义类内距离与类间距离比 $r=\frac{dist}{d(v_l,v_i)}$,这里 $d(v_l,v_i)$ 表示两个类心之间的距离,ε 取值服从下述函数:

$$s(r;a,b)=\begin{cases}\varepsilon_{\min}, & r\leqslant a\\ \varepsilon_{\min}+2\left(\frac{r-a}{b-a}\right)^2(\varepsilon_{\max}-\varepsilon_{\min}), & a<r\leqslant\frac{a+b}{2}\\ \varepsilon_{\max}-2\left(\frac{r-b}{b-a}\right)^2(\varepsilon_{\max}-\varepsilon_{\min}), & \frac{a+b}{2}<r<b\\ \varepsilon_{\max}, & r\geqslant b\end{cases}\tag{4}$$

式中,$\varepsilon_{\min}$,$\varepsilon_{\max}$ 给出了边界调整的最小、最大限,参数 a,b 对 r 进行度量,如取 $a=0.5$,$b=2$ 意味着当对象类内距离为类间距离的一半和两倍时对 ε 进行刻画,若取 $\varepsilon_{\min}=0.1$,$\varepsilon_{\max}=3$,$s(r;a,b)$ 函数图像如图 2 所示。

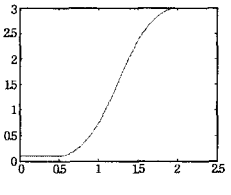


图 2　ε 调整函数

式(4)给出了阈值 ε 的调整规则,当然上述 s 函数的引入似乎相比原来的 ε 还增加了参数,但 s 函数中参数的给出显然要比 ε 更加容易一些。ε 反映的是聚类之间的重叠程度,但实际操作中,在聚类前是很难确定不同类之间的重叠程度的,而 s 函数中只需要给出期望最大重叠程度和最小重叠程度,即可在算法中根据不同类的情况自适应地调整。当然函数中也存在着参数 a,b 对调整函数形状进行的控制,但这些的设定都相对 ε 要容易些一些。更重要的是上述动态阈值粗糙 C 均值算法提供了更大的聚类灵活性,不同的聚类结果均可以通过最大 $\varepsilon_{\max}$、最小的 $\varepsilon_{\min}$ 的调整来获得,进一步,甚至可以定义一定的目标,通过这两个参数的调整来使得聚类达到某个目标。

变阈值粗糙 C 均值算法流程调整如下:

(1)选定 c 个初始聚类中心 v_i,$i=1,2,\cdots,c$;

(2)决策每一个对象 x_j 所属的分类,决策规则如下:

a. 计算 d_{kj},$k=1,2,\cdots,c$,且 $d_{ij}=\min\limits_{1\leqslant k\leqslant c}d_{kj}$;

b. 对于类 U_l,根据式(4)计算 ε_l,若 d_{lj} 满足:$|d_{lj}-d_{ij}|<\varepsilon_l$,则 $x_j\in\overline{BU}_i$ 且 $x_j\in\overline{BU}_l$,否则 $x_j\in\underline{BU}_i$;

(3)按照式(3)更新类心;

(4)重复步骤(2)—(3),直到收敛,这里收敛的条件可以取类心不再发生变化或取某个目标达到最优。

4　聚类效果指标

好的聚类一般具有较小的类内距离、较大的类间距离,也就是说要求同一类的对象具有最大的相似性,不同类的对象具有最大的相异性。Davies-Bouldin[7] 指数是用来刻画聚类效果的一个重要指标,仍沿用上面的符号,DB 指数计算如下:

$$DB=\frac{1}{c}\sum_{k=1}^{c}\max_{k\neq j}\left\{\frac{S(U_k)+S(U_j)}{d(U_k,U_j)}\right\}\tag{5}$$

式中,$S(U_i)=\frac{\sum\limits_{x_j\in U_i}\|x_j-v_i\|}{|U_i|}$ 表示类内平均距离,此值越小越好,$d(U_k,U_j)$ 表示类间距离,即 $d(v_k,v_j)$,此值越大越好。这个指标是针对一般聚类算法提出来的,在粗糙 C 均值算法中,Mitra[7] 引入了上下近似,从而改进 $S(U_i)$ 的计算,如下:

$$S(U_i)=\begin{cases}w_l\frac{\sum\limits_{x_j\in\underline{BU}_i}\|x_j-v_i\|}{|\underline{BU}_i|}+w_u\frac{\sum\limits_{x_j\in\overline{BU}_i-\underline{BU}_i}\|x_j-v_i\|}{|\overline{BU}_i-\underline{BU}_i|}, & \underline{BU}_i\neq\phi,\overline{BU}_i-\underline{BU}_i\neq\phi\\ \frac{\sum\limits_{x_j\in\underline{BU}_i}\|x_j-v_i\|}{|\underline{BU}_i|}, & \underline{BU}_i\neq\phi,\overline{BU}_i-\underline{BU}_i=\phi\\ \frac{\sum\limits_{x_j\in\overline{BU}_i-\underline{BU}_i}\|x_j-v_i\|}{|\overline{BU}_i-\underline{BU}_i|}, & \underline{BU}_i=\phi,\overline{BU}_i-\underline{BU}_i\neq\phi\end{cases}\tag{6}$$

实际上,上下近似的引入不仅更改了类内距离的计算方式,还改进了 DB 指数的计算方式。这是因为在粗糙聚类的概念中,类与类之间的关系方式同样改变了,类与类之间可能存在了边界,所以在本文中,DB 指数改进为:

$$DB=\frac{1}{c}\sum_{k=1}^{c}\max_{k\neq j}\left\{\frac{S(U_k)+S(U_j)-S(U_k\cap U_j)}{d(U_k,U_j)}\right\}\tag{7}$$

式中

$$S(U_k,U_j)=\frac{\sum_{x_p\in \overline{BU}_k\cap \overline{BU}_j}\| x_p-\bar{v}\|^2}{|\overline{BU}_k\cap\overline{BU}_j|}$$

式中，$\bar{v}=\frac{\sum_{x_p\in \overline{BU}_k\cap \overline{BU}_j} x_p}{|\overline{BU}_k\cap\overline{BU}_j|}$。

5 实验

为了对比不同算法的效果，下面采用人工生成的数据和图像数据进行实验。对于人工数据，我们采用两组不同形式的聚类数据，如图 3 所示。

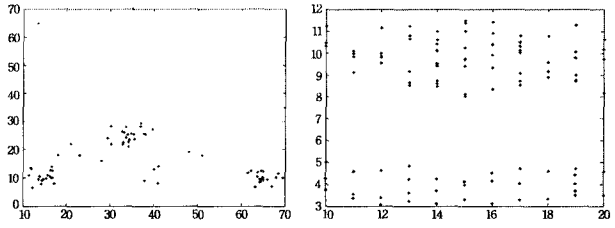
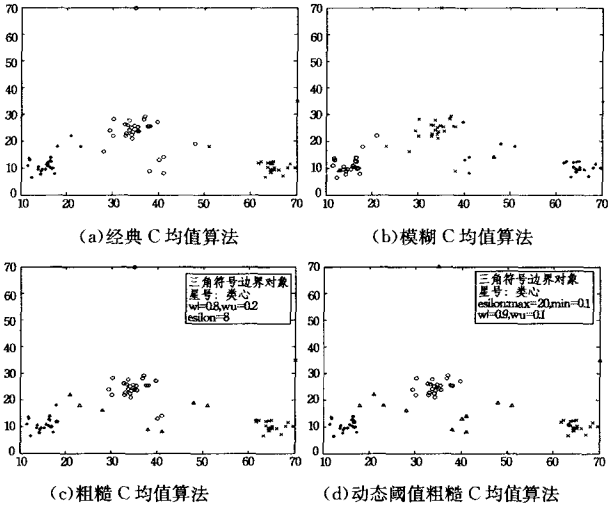


图 3 人工数据(分别对应 data1，data2)

从理论上来说，经典 C 均值算法和粗糙 C 均值算法都是以内类距离最小为目标来聚类，这就决定了这两种算法对于团状或簇状数据结构的数据具有较好的效果，而对于其它形状数据结构的数据聚类往往效果不好。下面针对 data1，data2 两种不同的数据结构来测试不同算法的聚类结果，如图 4、图 5 所示。



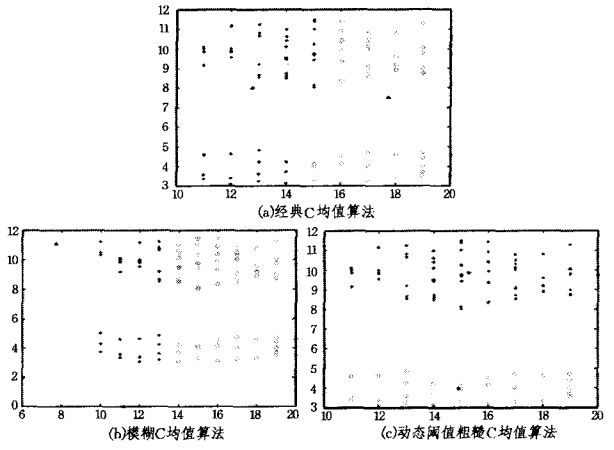
(a)经典 C 均值算法　(b)模糊 C 均值算法

(c)粗糙 C 均值算法　(d)动态阈值粗糙 C 均值算法

图 4 data1 数据聚类结果

data1 数据是通过随机生成均匀分布的 3 类数据，并在不同类之间设置了一些边界噪声数据构成，对于 data1 这类团状数据，4 类算法都基本能正确聚类，但在处理边界数据时显现出了差别，模糊 C 均值算法的结果与初始隶属度矩阵有较大的关系，当给定初始隶属度矩阵不合适时将出现错判。对于图中三角形代表的边界数据点，动态阈值粗糙 C 均值算法能较好地将其判别出来。

上面经典 C 均值算法和模糊 C 均值算法均出现了误判，当然 FCM 算法不能排除是由于初始隶属度矩阵选择不合适造成的，但好的初始隶属度矩阵的获取确实是困难的事情。动态阈值粗糙 C 均值算法对数据进行了正确的聚类。其中，$w_l=0.9$，$w_u=0.1$，$\varepsilon_{max}=2$，$\varepsilon_{min}=0.5$。



(a)经典C均值算法

(b)模糊C均值算法　(c)动态阈值粗糙C均值算法

图 5 data2 数据聚类

从图 6 中可以看出，ε 的取值对粗糙 C 均值聚类正确性的影响较大，w_l，w_u 影响聚类的收敛速度和类心的精度，而聚类的正确程度如由 ε 决定的。从聚类中心和 *DB* 指数来比较几种算法的性能，如表 1 所列。这里初始聚类中心就取 data1 数据生成的中心，即认为是真实类心。

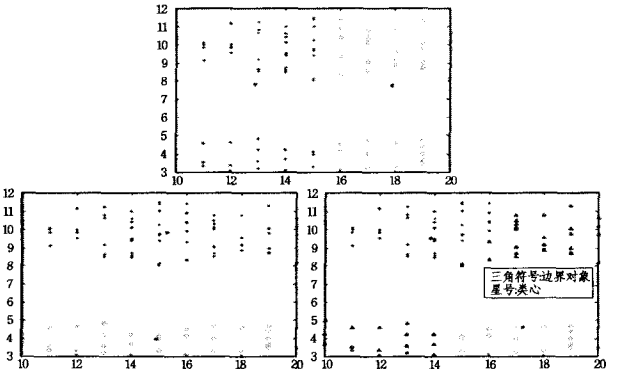
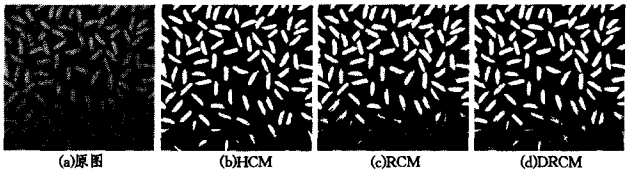


图 6 RCM(从左至右 ε 取值依次为 0.1，1.2，2)

表 1 几种算法聚类比较

	data1	
数据类心	(15,10)(35,25)(65,10)	
算法	类心	DB 指数
HCM	(15.67,11.81)(35.43,24.05)(64.58,11.69)	0.235
FCM	(46.47,0)(7.32,12.14)(5.46,0.86)	15.57
RCM	(18.09,11.74)(35.05,23.85)(61.57,12.08)	0.242
DRCM	(16.76,11.68)(34.69,24.64)(63.23,10.7)	0.208

模糊 C 均值算法由于类心受到隶属度的影响，类心的精度较低。最后再来看看算法在 Rice 图像分割上的应用情况，如图 7 所示。



(a)原图　(b)HCM　(c)RCM　(d)DRCM

图 7 Rice 图聚类效果

从结果可以看到，对于噪声污染的数据，经典 C 均值算法聚类时存在误判，粗糙 C 均值算法由于阈值设置的不合适，容易将噪声放大，而动态阈值的粗糙 C 均值算法能取得较好的效果。

在第 3 节中，ε 的调节是通过 $S(r;a,b)$ 来完成的，在实际

(下转第 242 页)

Knowledge Engineering,2001,11(3):231-258
[8] 李超明,苏开乐.一个基于智能的 MAS 模型及其方法论[J].计算机研究与发展,2007,44(6):980-989
[9] 张新良,石纯一.多 Agent 联盟结构动态生成算法[J].软件学报,2007,18(3):574-581
[10] Jo C,Chen G,Choi J. A new approach to the BDI agent-based modeling[C]// ACM. 2004:1545
[11] 毛新军,屈婷婷,王戟.自适应多 Agent 系统的面向 Agent 软件开发方法学 ODAM[J].计算机研究与发展,2008,45(011):1892-1901
[12] Bordini R, Hübner J. BDI Agent Programming in AgentSpeak Using Jason(Tutorial Paper)[M]. Berlin, German: Springer, 2006
[13] Bellifemine F,Caire G,Poggi A,et al. JADE:A software framework for developing multi-agent applications. Lessons learned [J]. Information and Software Technology, 2008, 50(1/2): 10-21
[14] Rao A. AgentSpeak(L):BDI agents speak out in a logical computable language[C]// Eindhoven. The Netherlands: Springer, 1996:42-55
[15] Bratman M E. Intention, Plans and Practical Reason[M]. Cambridge: Harvard University Press, 1987
[16] 路军,王亚东,王晓龙.BDI Agent 解释器的研究和改进[J].软件学报,2000,11(8):1118-1125
[17] Bordini R, Moreira Á. Proving BDI Properties of Agent-oriented Programming Languages: The asymmetry thesis principles in AgentSpeak(L)[J]. Annals of Mathematics and Artificial Intelligence, 2004, 42(1): 197-226
[18] Blackburn P, De Rijke M, Venema Y. Modal logic[M]. Cambridge Univ Pr, 2001
[19] Plotkin G. A structural approach to operational semantics[R]. Department of Computer Science, Aarhus University, 1981
[20] Milner R. Operational and algebraic semantics of concurrent processes[R]. Edinburgh: Department of Computer Science, University of Edinburgh, 1990
[21] Moore J. Inductive assertions and operational semantics[J]. International Journal on Software Tools for Technology Transfer (STTT), 2006, 8(4): 359-371
[22] Serbanuta T F, Rosu G, Meseguer J. A rewriting logic approach to operational semantics [J]. Information and Computation, 2009, 207(2): 305-340
[23] Milner R, Tofte M, Macqueen D, et al. The definition of standard ML: revised[M]. The MIT Press, 1997
[24] Wikstrom A. Functional programming using standard ML[M]. Prentice Hall International(UK) Ltd., 1987
[25] Moreira A, Bordini R. An operational semantics for a BDI agent-oriented programming language[C]// Toulouse. France, 2002: 45-59

(上接第 221 页)

计算中同样也存在着参数的选择问题,参数 a 指定最小阈值的分界点,一般可取 0.5,而参数 b 决定最大阈值的起始点,控制着 $S(r;a,b)$ 函数的斜率,可以想象,若 ε_{max} 越大,b 越小,边界对象将越多,ε_{max},b 共同控制着边界,对算法提供了一种灵活的调整机制,图 8 是针对 data1,不同 ε_{max},b 值时,边界元素的个数情况。从图 8 中可以看到,随着 ε_{max} 的增大,b 相应增大依然可以达到较好的效果。

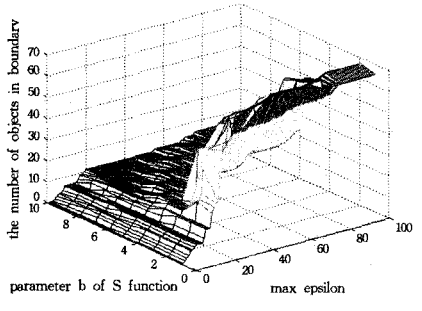


图 8 边界大小 VS 参数

结束语 聚类在数据挖掘、模式识别、图像分割等领域有着重要的应用。聚类算法中,C 均值算法是一类最经典的算法,粗糙 C 均值算法通过引进上下近似,改善了经典 C 均值算法分类的绝对性,承认了类间边界的存在。但粗糙 C 均值算法引入的参数带了算法的使用的困难,参数的设置对粗糙 C 均值算法的应用具有关键性的作用,本文提出了动态调整阈值 ε 的方法,通过最大阈值和最小阈值的设定从一定程度上降低了选择阈值的困难,实验结果也表明算法具有较好的效果,下一步的方向应该集中在如何在一个统一的框架下动态调整算法的所有参数。

参 考 文 献

[1] Pawlak Z. Rough Sets [J]. International Journal of Computer and Information science, 1982(11): 241-356
[2] Bezdek J C. Pattern recognition with fuzzy objective function algorithms[J]. New York: Plenum, 1981
[3] Lingras P, West C. Interval set clustering of Web users with rough k-means[R]. 2002-002. Department of Mathematics and Computer Science, St. Mary's University, Halifax, Canada, 2002
[4] Peters G. Some refinements of Rough K-means clustering[J]. Pattern Recognition, 2006(39): 1481-1491
[5] Mitra S. An evolutionary rough partitive clustering[J]. Pattern Recognition Letters, 2004(25): 1439-1449
[6] Malyszko D, Stepaniuk J. Rough Entropy Based k-Means Clustering[C]// RSFDGrc. 2009: 406-413
[7] Mitra S, Banka H, Pedrycz W. Rough-Fuzzy Collaborative[J]. IEEE transactions on systems, man, and cybernetics-part B: cybernetics, 2006, 4(36): 795-805
[8] 孙吉贵,刘杰,赵连宇.聚类算法研究[J].软件学报,2008(1):48-61
[9] 王丹,吴孟达.粗糙模糊 c 均值算法及其在图像聚类中的应用[J].国防科技大学学报,2007(2):76-80
[10] 郑超,苗夺谦,王睿智.基于密度加权的粗糙 K-均值聚类改进算法[J].计算机科学,2009,36(3):220-222
[11] 邵锐,巫兆聪,钟世明.基于粗糙 K 均值算法在图像分割中的应用[J].测绘信息与工程,2005(5):1-2
[12] Herawan T, Deris M M, Abawajy J H. A rough set approach for selecting clustering attribute [J]. Knowledge-Based Systems, 2010, 23: 220-231
[13] Miao Duo-qian, Duan Qi-guo, Zhang Hong-yun, et al. Rough set based hybrid algorithm for text classification[J]. Expert Systems with Applications, 2009, 36: 9168-9174
[14] Sarkar M. Fuzzy-rough nearest neighbor algorithms in classification[J]. Fuzzy sets and systems, 2007, 158: 2134-2152