模糊 C-球壳聚类算法的研究 [1]

魏立梅 谢维信 *

(西安电子科技大学电子工程学院 202 室 西安 710071)
*(深圳大学校长办公室 深圳 518060)

摘 要 通过对基于同一目标函数的两种模糊 C-球壳 (FCSS) 聚类算法性能的比较与分析，提出一种新算法．该算法收敛速度快，聚类结果准确．

关键词 模糊 C-球壳聚类，目标函数，原型
中图号 TP391.4

1 引 言

模糊聚类已经广泛应用于模式识别和计算机视觉领域 [1,2] ．模糊 C-均值 (FCM) 聚类算法是模糊聚类中应用最广泛的方法 [1] ． FCM 在二十多年的发展过程中，聚类原型由点逐步扩展到线 [3] 、面 [4] 、球壳 [5−8] 和椭球壳 [9,10] ．最近又出现聚类原型为矩形壳和多边形壳的聚类算法 [11] ．下面介绍模糊 C-球壳聚类算法的发展情况。

最早开始壳聚类算法研究的是 Dave ．在文献 [5, 6] 中， Dave 提出了球壳状分布数据的聚类问题．为解决球壳状数据的聚类问题， Dave 采用球壳作为聚类原型，定义目标函数，推导出球壳聚类的迭代算法，并用实验证明算法能有效地聚类球壳状分布的数据．后来，有不少学者致力于这方面的研究，相继出现了新的球壳聚类算法 [7,8] ．

设壳状分布的数据集 $X = \{x_1, x_2, \cdots, x_N\}$, X 中含有 C 个球壳． C 个聚类原型为 (v_i, r_i), $i = 1 \sim C$ ． v_i 和 r_i 分别表示第 i 个球壳的中心和半径． u_{ij} 表示数据 x_j 对第 i 个原型 (v_i, r_i) 的隶属度． $U = \{u_{ij}\}$, $V = \{v_i\}$, $R = \{r_i\}$ ． $U = \{u_{ij}\}$ 满足下式：

$$u_{ij} \in [0,1], \forall i, \forall j \tag{1a}$$

$$\sum_{i=1}^{C} u_{ij} = 1, \forall j \tag{1b}$$

$$0 < \sum_{j=1}^{N} u_{ij} < N, \forall i \tag{1c}$$

模糊 C-球壳聚类算法 (FCSS) 的目标函数如下：

$$J(U, V, R) = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^m d_{ij}^2 \tag{2}$$

在已有的 FCSS 算法中，目标函数中距离 d_{ij} 的定义有两种形式，分别表示为 (3) 式和 (4) 式。

$$d_{ij}^2 = (\|x_j - v_i\| - r_i)^2 \tag{3}$$

$$d_{ij}^2 = (\|x_j - v_i\|^2 - r_i)^2 \tag{4}$$

[1] 1998-12-09 收到， 1999-12-30 定稿

基于 (3) 式的 FCSS 算法以文献 [6] 和文献 [7] 为代表. 文献 [8] 是基于 (4) 式的 FCSS 算法. 本文研究文献 [6] 和文献 [7] 中提出的都基于 (3) 式的 FCSS 算法.

2 基于同一距离测度的 FCSS 算法性能的分析

(2) 式和 (3) 式表示的 FCSS 聚类问题, 是 (1) 式约束下的优化问题. (2) 式中 J 的条件极值可以由拉格朗日乘数法求得. 首先, J 在 (1) 式下的条件极值可以表示成 (5) 式, 其中 λ 为常数.

$$J(U,V)=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^{m}d_{ij}^{2}+\lambda\left(\sum_{i=1}^{C}u_{ij}-1\right) \tag{5}$$

令 (5) 式中函数 J 的三个一阶偏导数为 0 , 得到 (6) 式至 (8) 式 (参见文献 [6]) :

$$I_j=\{i|1\le i\le C,\ d_{ij}=0\}$$

$$\overline{I}_j=\{1,2,\cdots,C\}-I_j$$

$$\text{当 } I_j=\phi,\quad u_{ij}=1\Big/\left\{\sum_{k=1}^{C}\left(\frac{d_{ij}^{2}}{d_{kj}^{2}}\right)^{1/(m-1)}\right\} \tag{6a}$$

$$\text{当 } I_j\neq\phi,\ \forall i\in\overline{I}_j,\ u_{ij}=0,\ \text{且}\sum_{i=1,i\in I_j}^{C}u_{ij}=1 \tag{6b}$$

$$r_i=\sum_{j=1}^{N}u_{ij}^{m}\|x_j-v_i\|\Big/\sum_{j=1}^{N}u_{ij}^{m} \tag{7}$$

$$\sum_{j=1}^{N}u_{ij}^{m}\frac{d_{ij}}{\|x_j-v_i\|}(x_j-v_i)=0 \tag{8}$$

(6) 式给出了隶属度的迭代公式, (7) 式给出了半径的迭代公式. (8) 式是耦合非线性方程.

Dave 采用数值方法求解 (8) 式的方程, 得到类中心的迭代公式如下 [6]:

$$v_i=\sum_{j=1}^{N}u_{ij}^{m}x_j\Big/\sum_{j=1}^{N}u_{ij}^{m} \tag{9}$$

Man 和 Gath 则利用图 1 , 从 (8) 式中推导出二维情况下类中心的精确解 (最优解) 公式如下 [7]:

$$v_i^{*}=\frac{\sum_{j=1}^{N}u_{ij}^{m}\begin{pmatrix}x_{j1}-r_i\cos\theta\\ x_{j2}-r_i\sin\theta\end{pmatrix}}{\sum_{j=1}^{N}u_{ij}^{m}}=\frac{\sum_{j=1}^{N}u_{ij}^{m}(x_j-y_j)}{\sum_{j=1}^{N}u_{ij}^{m}} \tag{10}$$

其中 $x_j=(x_{j1},x_{j2})$, $y_j=(r_i\cos\theta,r_i\sin\theta)$, y_j 是模为 r_i 的径向矢量.

将 (6) 、 (7) 和 (9) 式表示的聚类算法记为 FCSS1(见文献 [6]) ; 将 (6) 、 (7) 和 (10) 式表示的聚类算法记为 FCSS2(见文献 [7]) . FCSS1 和 FCSS2 具有相同的隶属度和半径迭代公式, 不同的类中心迭代公式. 它们性能上的不同只能通过类中心迭代公式表现出来. 下面我们先给出这两种算法的类中心迭代公式的不同的几何意义和代数意义, 然后对这两种方法进行性能的比较与分析.

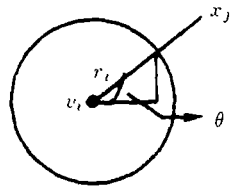


图 1　数据与原型的关系示意图

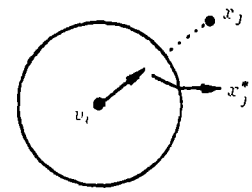


图 2　几何意义示意图

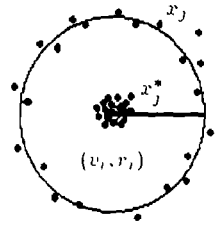


图 3　几何意义示意图

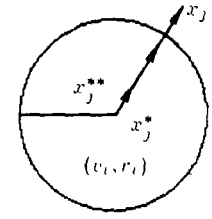


图 4　x_j 、 x_j^* 和 x_j^{**} 的位置关系示意图

注：(9) 式用圆壳附近的数据点 x_j 的加权中点作为 $v_i(t+1)$

(10) 式用圆心附近的 x_j^* 的加权中点作为 $v_i(t+1)$

2.1 类中心迭代公式的几何意义

在 (9) 式中，若对 $\forall j$ ，隶属度 $u_{ij}=1$ 或 $u_{ij}=0$ ，则类中心 v_i 由完全隶属于类原型 (v_i, r_i) 的数据点确定，是所有位于圆 (v_i, r_i) 上的数据点的中点．当圆 (v_i, r_i) 上的数据点是逐对关于类中心 v_i 对称时，圆上的数据点的中点恰好是 v_i ．而当圆上的数据点并不是逐对对称时，它们的中点并不是类中心，只是类中心的近似．因此，(9) 式是按每个数据点 x_j 对 $v_i(t)$ 的隶属度 u_{ij} 对每个数据点进行加权，用加权中点 $v_i=\sum_{j=1}^{N}u_{ij}^m x_j \Big/ \sum_{j=1}^{N}u_{ij}^m$ 作为下一次迭代时的类中心 $v_i(t+1)$ ．

同理，(10) 式也是用加权中点作为下一次迭代时的类中心．只是被加权的不是数据点 x_j ，而是 x_j^* ．如图 2 所示．其中 $x_j^*=x_j-y_j$ ．如果 x_j 在圆上，则 x_j^* 就是类中心．

(9) 式和 (10) 式在求解类中心上表现出的不同的几何意义，可以用图 3 表示．几何意义的不同，对 FCSS1 和 FCSS2 的性能有很大影响．

2.2 类中心迭代公式的代数意义

既然 (10) 式是 (8) 式的精确解，我们就以 (10) 式为基准将 (9) 式改写如下：

$$v_i=\frac{\sum_{j=1}^{N}u_{ij}^m\begin{pmatrix}x_{j1}-r_i\cos\theta\\ x_{j2}-r_i\sin\theta\end{pmatrix}+\sum_{j=1}^{N}u_{ij}^m\begin{pmatrix}r_i\cos\theta\\ r_i\sin\theta\end{pmatrix}}{\sum_{j=1}^{N}u_{ij}^m}=v_i^*+\Delta v \tag{11}$$

其中 $\Delta v=\dfrac{\sum_{j=1}^{N}u_{ij}^m\begin{pmatrix}r_i\cos\theta\\ r_i\sin\theta\end{pmatrix}}{\sum_{j=1}^{N}u_{ij}^m}=\dfrac{\sum_{j=1}^{N}u_{ij}^m y_j}{\sum_{j=1}^{N}u_{ij}^m}$ 。

从 (11) 式可以得到：(9) 式给出的解是精确解 (最优解) 加上一个增量 Δv，或者说加上一个步长 Δv ．

2.3 FCSS1 和 FCSS2 性能的比较

本文对 FCSS1 和 FCSS2 进行了大量实验. 实验表明: FCSS1 算法和 FCSS2 算法各有优缺点. FCSS1 收敛速度快, 聚类结果有偏差; FCSS2 收敛速度慢, 聚类结果很准确. 本文给出两个相交圆壳情况下, FCSS1 和 FCSS2 聚类结果的例子. 如图 5 至图 7 所示.

从图 5 看到: FCSS1 的圆心轨迹是折线型, 两个圆心从初始化位置迅速向正确的圆心位置靠近, 收敛时圆心和半径的值与正确值相比有偏差; FCSS2 的圆心轨迹很平滑, 两个圆心由初始化位置缓慢地靠近正确的圆心位置, 收敛时, 圆心和半径的值与正确值相比偏差很微小. 圆心和半径的收敛曲线见图 6 和图 7 .

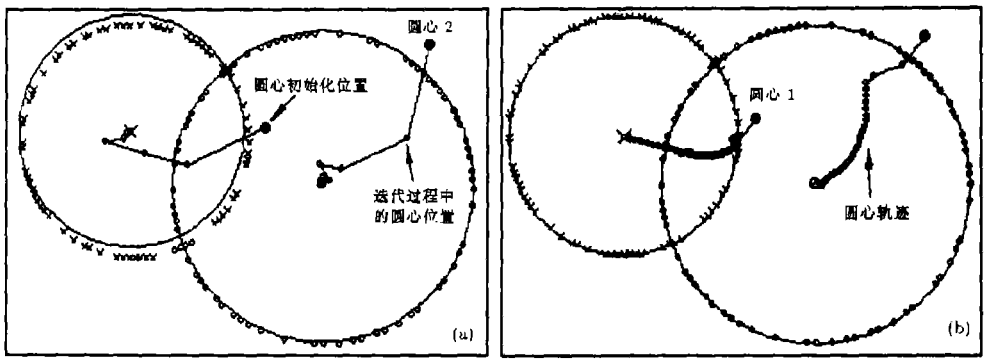
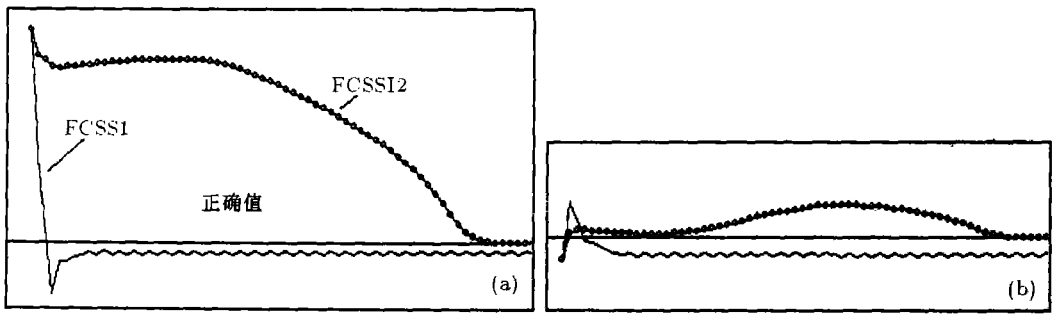
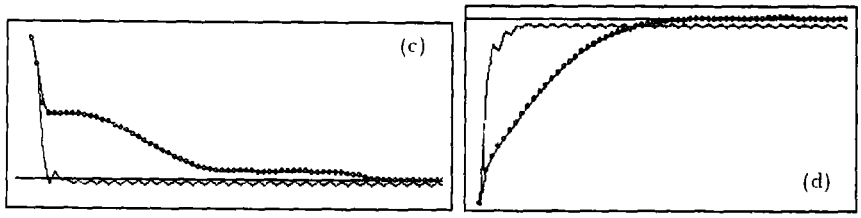


图 5 两个相交圆壳的聚类结果
(a) FCSS1 聚类结果 (b) FCSS2 聚类结果



(a) 圆心 1 的 x 坐标收敛曲线 (b) 圆心 1 的 y 坐标收敛曲线



(c) 圆心 2 的 x 坐标收敛曲线 (d) 圆心 2 的 y 坐标收敛曲线

图 6 圆心收敛曲线

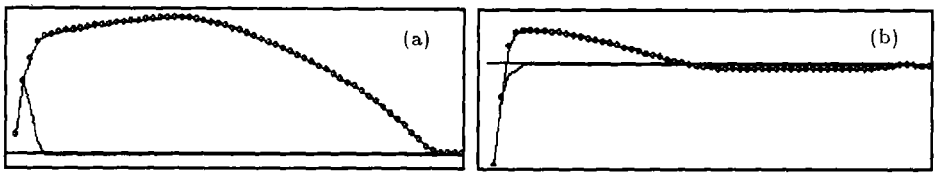


图 7　半径收敛曲线
(a) 圆 1 的半径收敛曲线　(b) 圆 2 的半径收敛曲线

理论上讲，FCSS2 中类中心、半径和隶属度的迭代公式都是最优解，所以 FCSS2 的迭代过程是一个逐步优化迭代的过程，类中心和半径应该很平滑地收敛于正确的值，这与实验结果吻合．但是，平滑收敛的代价是：当初始化不好时，圆心和半径平滑收敛过程通常很慢．而 FCSS1 中类中心的迭代公式是数值解，不是最优解．每次迭代的类中心都是最优解加上一个增量，或者说，都是最优解加上一个步长．这就使得类中心在迭代过程出现跳跃，迭代轨迹呈折线型，在开始迭代时，这种跳跃使类中心迅速向正确值收敛，而在类中心离正确值很近时，这种跳跃使类中心在正确值处产生微小的抖动，造成收敛时聚类结果的偏差．类中心的跳跃无疑对半径的收敛造成影响，使半径的收敛曲线也出现跳跃．实验结果也证明了这一点．

文献 [7] 中提出了 FCSS2 的初始化方法，加快 FCSS2 的收敛速度．但是，初始化时，首先要按有关文献提出的方法判断数据分布是同心圆的情况，还是相交圆的情况．然后，再分别按不同方法初始化．

鉴于 FCSS1 和 FCSS2 的特点，本文提出一种 FCSS 方法．不仅不需要文献 [7] 中单独的初始化过程，而且聚类效果很好．

3 聚类新算法

FCSS1 和 FCSS2 基于同一距离测度，仅仅由于它们采用不同的类中心迭代公式，使得两种算法表现出不同的性能．本文提出一种算法，不妨记为 α-FCSS．该算法收敛速度快、聚类结果准确．

3.1 α-FCSS 算法

既然 FCSS1 和 FCSS2 性能的不同是通过类中心迭代公式表现出来的，要想在收敛速度和聚类准确性上获得很好的效果，就必须采用新的类中心迭代公式．α- FCSS 算法中采用类中心迭代公式如下：

$$v_i = (1-\alpha)\frac{\sum_{j=1}^{N} u_{ij}^m x_j}{\sum_{j=1}^{N} u_{ij}^m} + \alpha\frac{\sum_{j=1}^{N} u_{ij}^m \begin{pmatrix} x_{j1} - r_i\cos\theta \\ x_{j2} - r_i\sin\theta \end{pmatrix}}{\sum_{j=1}^{N} u_{ij}^m} \tag{12}$$

即

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^m \begin{pmatrix} x_{j1} - \alpha r_i\cos\theta \\ x_{j2} - \alpha r_i\sin\theta \end{pmatrix}}{\sum_{j=1}^{N} u_{ij}^m} = \frac{\sum_{j=1}^{N} u_{ij}^m (x_j - \alpha y_j)}{\sum_{j=1}^{N} u_{ij}^m} \tag{13}$$

其中 $\alpha \in [0,1]$．记 $x_j^{**} = x_j - \alpha y_j$，则 α-FCSS 算法中类中心 v_i 是 x_j^{**} 的加权中点．x_j，x_j^* 和 x_j^{**} 的位置关系示于图 4．(13) 式还可以表示为下式：

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^m \begin{pmatrix} x_{j1} - r_i\cos\theta \\ x_{j2} - r_i\sin\theta \end{pmatrix} + (1-\alpha)\sum_{j=1}^{N} u_{ij}^m \begin{pmatrix} r_i\cos\theta \\ r_i\sin\theta \end{pmatrix}}{\sum_{j=1}^{N} u_{ij}^m} = v_i^* + (1-\alpha)\Delta v \tag{14}$$

α–FCSS 算法中的类中心也是精确解 (最优解) 加上一个增量，或者说步长．只是步长大小为 $(1-\alpha)\Delta v$ ． α–FCSS 算法中仍旧用 (6) 、 (7) 式作为隶属度和半径的迭代公式．

当 $\alpha=0$ ， α–FCSS 算法退化为 FCSS1 算法，当 $\alpha=1$ 时， α– FCSS 算法退化为 FCSS2 算法． α 可以是常数，也可以是时变的．当 α 为常数时， α–FCSS 算法的性能在聚类过程中，由于步长的存在，类中心和半径的调整速度比 FCSS2 快，同样由于步长，在收敛时， α–FCSS 算法存在偏差，只是与 FCSS1 相比， $(1-\alpha)$ 的衰减作用使偏差要小．通常， α 是时变的．在聚类初期， α 值可以较小，根据 (14) 式， α–FCSS 算法的类中心主要由 FCSS1 算法确定，因此， α–FCSS 算法很快地向最优解靠近．随着迭代次数的增加， α 值可以逐步增大，使 α–FCSS 的类中心逐步由 FCSS2 确定， α–FCSS 算法可以平滑地收敛于最优解，而且偏差微小．

在聚类初期， α 值较小， α–FCSS 算法的类中心主要是由 FCSS1 算法确定的过程，可以看成用 FCSS1 初始化 α–FCSS 的过程．随着迭代次数的增加， α 值逐步增大， α–FCSS 的类中心是逐步由 FCSS2 确定的过程，可以看成在初始化很好的情况下， α–FCSS 是用 FCSS2 聚类的过程．这样， α-FCSS 在聚类速度和准确性上都能得到满意的结果．

3.2 新算法的实验结果

对于图 5 给出的聚类问题，取 $\alpha=5$，得到实验结果见表 1 和图 8．为进一步说明 α-FCSS 算法性能，又取 $\alpha=1-e^{-\beta t}$ ，其中 $\beta=0.5$, t 为迭代次数．实验结果见表 1 和图 8 ．从表 1 中可以看到， α-FCSS 算法的收敛速度快，聚类结果准确．

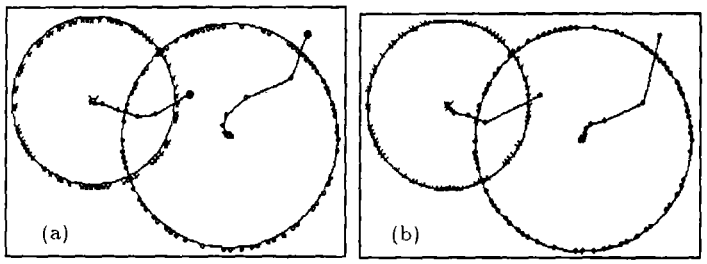


图 8 α-FCSS 算法聚类结果
(a) $\alpha=0.5$
(b) $\alpha=1-e^{-\beta t}$ ，其中 β =0.5

表 1 FCSS1 、 FCSS2 和 α-FCSS 的性能比较

		圆壳 1		圆壳 2		收敛速度
		圆心	半径	圆心	半径	(以迭代次数计)
正确值		15, 20	12	35, 25	16	
FCSS1		14.33, 19.09	11.93	34.95, 24.43	15.97	8
FCSS2		15.02, 20.00	12.00	35.00, 25.00	16.00	64
α-FCSS	$\alpha=0.5$	14.88, 19.69	12.00	35.16, 24.61	16.00	10
	$\alpha=1-e^{-\beta t}$	15.00, 19.99	12.00	35.01, 24.98	15.99	8

文献 [7] 中给出了四个圆壳聚类的例子．本文用该例说明 α-FCSS 算法在类数较多时的性能，见图 9 ．为便于比较，图 9 中聚类未采用随机初始化，而是给出较好的初始化原型．从图中可以看到：在初始化较好时， α-FCSS 算法和 FCSS2 都得到很好的聚类结果，而且 FCSS2 收敛速度较快．但是， α-FCSS 算法中各圆心向正确位置调整的速度更快．由于篇幅所限，余例略．

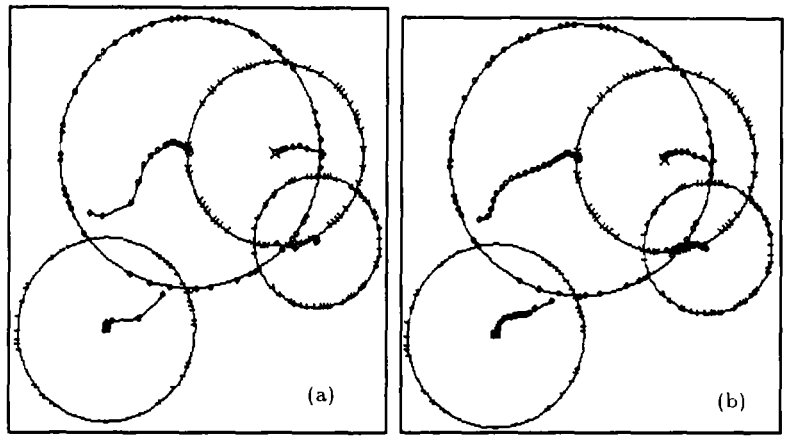


图 9　α-FCSS 算法与 FCSS2 算法的比较
(a) 新算法　(b) FCSS2

4 结束语

本文在分析基于同一距离测度的两种球壳聚类算法 FCSS1 和 FCSS2 性能的基础上，提出一种新算法 α-FCSS，该算法收敛速度快，聚类结果准确。

参 考 文 献

[1] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York, Plenum Press, 1981, Chapter 2.

[2] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Englewood Cliffs, New Jersey, Prentice Hall, 1988, Chapter 4.

[3] J. C. Bezdek, C. Coray, R. Gunderson, J. Watson, Detection and characterization of cluster substructure I, linear structure, Fuzzy c-lines, SIAM J. Appl. Math., 1981, 40(3), 339–357.

[4] J. C. Bezdek, C, Coray, R. Gunderson, J. Watson, Detection and characterization of cluster substructure II, Fuzzy c-varieties and convex combinations thereof, SIAM J. Appl. Math., 1981, 40(3), 358–372.

[5] R. N. Dave, Fuzzy shell clustering and applications to circle detection in digital images, Int. J. General Syst., 1990, 16(4), 343–345.

[6] R. N. Dave, Generalized fuzzy c-shells clustering and detection of circular and elliptical boundaries, Pattern Recognition, 1992, 25(7), 713–721.

[7] Y. Man, I. Gath, Detection and separation of ring-shaped clusters using fuzzy clustering, IEEE Trans. on PAMI, 1994, PAMI-16(8), 855–861.

[8] R. Krishnapuram, O. Nasraoui, H. Frigui, The fuzzy c-spherical shells algorithm, a new approach, IEEE Trans. on NN, 1992, NN-3(5), 663–671.

[9] H. Frigui, R. Krishnapuram, A comparison of fuzzy shell clustering methods for the detection of ellipses, IEEE Trans. on Fuzzy System, 1996, 4(2), 193–199.

[10] R. N. Dave, K. Bhaswan, Adaptive fuzzy c-shells clustering and detection of ellipses, IEEE Trans. on NN, 1992, NN-3(5), 643–662.

[11] F. Hoeppner, Fuzzy shell clustering algorithms in image processing, fuzzy c-rectangular and 2-rectangular shells, IEEE Trans. on Fuzzy System., 1997, 5(4), 599–613.

A STUDY ON FUZZY C-SPHERICAL SHELL(FCSS) CLUSTERING ALGORITHMS

Wei Limei Xie Weixin*

(*Lab. 202, School of Electronic Engineering, Xidian University, Xi'an* 710071, China)

*(*Shenzhen University, Shenzhen* 518060, China)

Abstract The two Fuzzy C-Spherical Shell (FCSS) clustering algorithms based on the same object function are analyzed and a new algorithm is proposed. The properties of the new algorithm in both the speed and the accuracy are appropriate.

Key words Fuzzy C-Spherical Shell(FCSS), Object function, Prototype

魏立梅： 女， 1970 年生，博士，研究方向包括模式识别、信号处理、模糊理论等方面.

谢维信： 男， 1941 年生，教授，博士生导师，深圳大学校长，研究方向包括模糊理论、图像处理、模式识别、神经网络、计算机视觉等.