文章编号: 1001-0920 (2015) 10-1766-07 DOI: 10.13195/j.kzyjc.2014.1085

基于混合蛙跳与阴影集优化的粗糙模糊聚类算法

蒙祖强, 胡玉兰, 蒋 亮, 常红岩

(广西大学 计算机与电子信息学院，南宁 530004)

摘 要: 针对粗糙模糊聚类算法对初值敏感、易陷入局部最优和聚类性能依赖阈值选择等问题, 提出一种混合蛙跳与阴影集优化的粗糙模糊聚类算法 (SFLA-SRFCM). 通过设置自适应调节因子, 以增加混合蛙跳算法的局部搜索能力; 利用类簇上、下近似集的模糊类内紧密度和模糊类间分离度构造新的适应度函数; 采用阴影集自适应获取类簇阈值. 实验结果表明, SFLA-SRFCM 算法是有效的, 并且具有更好的聚类精度和有效性指标.

关键词: 粗糙集；阴影集；粗糙模糊聚类；混合蛙跳算法

中图分类号: TP273 **文献标志码:** A

Shuffled frog leaping algorithm and shadowed sets-based rough fuzzy clustering algorithm

MENG Zu-qiang, *HU Yu-lan*, *JIANG Liang*, *CHANG Hong-yan*

(College of Computer and Electronics Information，Guangxi University，Nanning 530004，China．Correspondent: MENG Zu-qiang，E-mail：zqmeng@126.com)

Abstract: For the problem that the rough fuzzy clustering algorithm is sensitive to the initial value, easy to fall into a local optimal solution, and the clustering performance of algorithm depends on the selection of threshold, a rough fuzzy clustering algorithm based on the shuffled frog leaping algorithm and shadowed sets(SFLA-SRFCM) is proposed. The adaptive factor is developed to enhance the local search ability, the within cluster tighness and the between cluster scatter of fuzzy lower approximate sets and fuzzy upper approximate sets are used to construct a new fitness function. Shadowed sets are applied to obtain the threshold adaptively. Experimental results show that SFLA-SRFCM is effective and has better clustering accuracy and validity index.

Keywords: rough sets；shadowed sets；rough fuzzy clustering；shuffled frog leaping algorithm

0 引 言

聚类分析是一种重要的智能信息处理方法, 已经被广泛应用在模式识别、数据挖掘以及生物信息发现等领域[1]. 聚类的目的是根据一定的测度准则, 将数据集划分成若干个簇, 使得簇内对象尽可能相似, 簇间对象尽可能相异. 根据对象隶属关系明确与否, 现有的聚类算法分为硬聚类和软聚类两种, 后者因扩展了隶属度的取值范围, 具有更强的适应性.

粗糙集和模糊集提供了处理不确定问题的数学框架, 二者在一定程度上具有互补性. 为了适应复杂聚类问题的应用需求, 许多学者致力于实现多种软计算方法在聚类中的融合统一. 最具代表性的是 Mitra 等[2]提出的粗糙模糊聚类算法 RFCM, 该算法结合了粗糙集和模糊集两种软数据处理方法, 通过粗糙集粗糙划分论域, 利用模糊集中隶属函数处理模糊边界区域, 并引入模糊度量策略以提高聚类对不同参数选择的鲁棒性. 与其他聚类算法相比, RFCM 算法具有更好的数据处理能力, 但其本质上还是采用梯度下降方法寻求问题的最优解, 算法对初始聚类中心敏感, 且聚类性能和泛化能力受阈值参数的直接影响, 初始聚类中心选取和参数选择成为 RFCM 算法推广的最大瓶颈. 针对这些问题, 研究者们提出了许多改进算法: Maji 等[3]认为粗糙集下近似区域样本的归类属性是明确的, 以下近似集样本隶属度值为 1 改进原始初始聚类中心的计算方法, 取得了较好的聚类结果; 姚丽娟等[4]通过核函数将样本映射到高维空间, 采用粒

收稿日期: 2014-07-09；**修回日期:** 2014-10-27.

基金项目: 国家自然科学基金项目(61363027)；广西自然科学基金项目(2012GXNSFAA053225).

作者简介: 蒙祖强(1974−), 男, 教授, 博士生导师, 从事粒度计算、知识发现等研究；胡玉兰(1989−), 女, 硕士生, 从事数据挖掘、粗糙集的研究.

子群算法优化初始聚类中心, 但算法没有考虑阈值参数对聚类性能的影响; 王学恩等[5]通过方差划分构建评价函数寻求类簇最优阈值, 并基于样本点模糊性值和粗糙度确定区域重要性权重, 提出了基于不确定性度量的参数选择方法, 但算法收敛速度较慢; 周杰等[6]利用阴影集理论为每个类簇设置不同阈值, 提高了聚类精度, 但未考虑初始聚类中心的选择差异性影响, 容易陷入局部最优; Peters[7]应用拉普拉斯无差别原理改进了粗糙聚类算法的均值, 使算法具有稳健的下近似集权重, 是粗糙集聚类问题新的突破.

混合蛙跳算法 (SFLA)[8]是一种模拟青蛙觅食过程的新进化算法, 该算法的数学模型建立在子群内青蛙个体的模因进化和子群间利用模因信息进行全局混洗的基础上. SFLA 算法结合局部深度搜索策略和全局信息交换技术, 实现了蛙群之间信息的有效共享和更新, 从而快速准确地搜索到最佳位置. 该算法具有并行性、收敛速度快、需设置的参数较少以及编程易于实现等优点, 已成为智能搜索算法研究的热点.

在文献 [6] 的基础上, 本文提出一种基于混合蛙跳与阴影集的粗糙模糊聚类算法 (SFLA-SRFCM). 该算法首先构建自适应调节因子来控制最差青蛙的更新尺度, 然后利用改进的混合蛙跳算法对粗糙模糊聚类算法的初始聚类中心进行优化, 并通过阴影集构建函数自适应获取聚类阈值. 实验结果表明, SFLA-SRFCM 算法比 FCM、RFCM、SRFCM和 SFLA-KM 算法的寻优能力更强, 聚类性能更佳.

1　相关知识介绍

1.1　粗糙集理论

粗糙集理论是一种处理不确定、不完备信息的数学工具. 经典粗糙集理论通过等价关系划分论域, 并利用上、下近似集描述不确定信息. 下面简要介绍本文用到的一些基本概念.

定义 1　四元组 $\mathrm{IS}=(U,A,V,f)$ 表示一个信息系统. 其中: U 为论域, A 为非空属性集, V 为属性值集, V_a 为属性 a 的值域; $f:U\times A\to V$ 为信息函数, 用来计算对象在属性上的取值, 对于 $\forall a\in A, x\in U$, 有 $f(x,a)\in V_a$.

定义 2　$B\subseteq A$, $R(B)=\{(x,y)|x,y\in U,\forall b\in B, f(x,b)=f(y,b)\}$, 称 R 为不可区分二元关系.

定义 3　$X\subseteq U$, $B\subseteq A$, X 关于 B 的下近似集和上近似分别定义为

$$\underline{B}(X)=\{x\in U|[x]_B\subseteq U\},$$
$$\overline{B}(X)=\{x\in U|[x]_B\bigcap X\neq\phi\}.$$

上近似集与下近似集的差为边界集 $\mathrm{BN}_B(X)=\overline{B}(X)-\underline{B}(X)$, 其中 $[x]_B=U/R(B)$ 为 X 在 B 上的等价类.

$(\underline{B}(X),\overline{B}(X))$ 是 X 的粗糙表示, $\underline{B}(X)$ 为完全属于 X 的对象构成的集合, $\overline{B}(X)$ 为可能属于 X 的对象构成的集合, $\mathrm{BN}_B(X)$ 为可能属于 X, 也可能不属于 X 的对象构成的集合. 边界区域 $\mathrm{BN}_B(X)$ 越大, 知识的不确定程度越高.

1.2　模糊聚类算法

模糊聚类算法 (FCM) 引入模糊集模糊化聚类结构, 通过迭代不断更新聚类中心和隶属度矩阵. FCM 算法的最小化目标函数为

$$\min\sum_{i=1}^{N}\sum_{k=1}^{C}\mu_{ik}^{m1}||x_i-v_k||^2. \tag{1}$$

其中: $\sum_{k=1}^{C}\mu_{ik}=1$, $0\leqslant\sum_{i=1}^{n}u_{ik}\leqslant n$, $\mu_{ik}\in[0,1]$; N 为样本数; C 为聚类数目; v_k 为类 C_k 的聚类中心; m_1 为模糊指数, 一般取值为 2. 利用拉格朗日函数求解得到聚类中心和隶属度矩阵分别为

$$v_k=\frac{1}{\sum_{i=1}^{N}\mu_{ik}^{m1}}\sum_{i=1}^{N}\mu_{ik}^{m1}x_i, \tag{2}$$

$$\mu_{ik}=1\Big/\sum_{s=1}^{C}\Big(\frac{\|x_i-v_k\|^2}{\|x_i-v_s\|^2}\Big)^{\frac{2}{m_1-1}}. \tag{3}$$

FCM 算法对初值敏感, 并且对边界不确定数据的聚类能力较差. Lingras 等[9]将粗糙集引入聚类算法中, 以类下近似集和边界集表示类簇, 增强了算法对边界样本的处理能力, 但不能处理模糊信息. 随后, 文献 [2] 在 Lingras 粗糙聚类算法的基础上, 提出了一种软集合并粗糙模糊聚类算法 RFCM. 为了便于本文讨论, 下面简单介绍 RFCM 算法的基本思想.

1.3　粗糙模糊聚类算法

设 X 是 d 维数据集, C_k 表示第 k 个类簇, C_k 对应的聚类中心用 v_k 表示, $\underline{C_k}$ 和 $\overline{C_k}$ 为类 C_k 的下近似集和上近似集, $C_k^B=\overline{C_k}-\underline{C_k}$ 为类 C_k 的边界集. 参数 w_l 和 w_b 分别表示下近似区域和边界区域的相对重要性. RFCM 算法聚类中心计算方法如下:

$$v_k=\begin{cases}w_lA_1+w_bB_1,\ \underline{C_k}\neq\phi,\ C_k^B\neq\phi;\\ A_1,\ \underline{C_k}\neq\phi,\ C_k^B=\phi;\\ B_1,\ \underline{C_k}=\phi,\ C_k^B\neq\phi.\end{cases} \tag{4}$$

其中

$$A_1=\sum_{x_i\in\underline{C_k}}u_{ik}^{m_1}x_i\Big/\sum_{x_i\in\underline{C_k}}u_{ik}^{m_1},$$
$$B_1=\sum_{x_i\in C_k^B}u_{ik}^{m_1}x_i\Big/\sum_{x_i\in C_k^B}u_{ik}^{m_1}.$$

u_{ik} 的计算公式与式(3)相同. 若 μ_{ik} 和 μ_{il} 表示 x_i 到各聚类中心的最大和次大隶属度值, 如果 $\mu_{ik}-\mu_{il}\leqslant\delta$, 则 $x_i\in\overline{C_k}$, 且 $x_i\in\overline{C_l}$; 否则 $x_i\in\underline{C_k}$, 且 $x_i\in\overline{C_k}$. 当 $\overline{C_k}-\underline{C_k}=\phi$ 时, 即边界区域为空集时, RFCM 算法即为 FCM 算法.

RFCM 算法的性能直接受阈值 δ 的影响, 并且初始聚类中心的选择影响算法的收敛特征. 为了提高 RFCM 算法的聚类性能, 需要同时考虑参数获取和初值优化的方法.

2 混合蛙跳算法与阴影集优化粗糙模糊聚类方法

本节中, 首先介绍混合蛙跳算法和阴影集理论; 然后构建自适应调节因子控制最差青蛙的更新尺度, 基于聚类中心编码青蛙个体, 利用阴影集构建关于阈值的目标函数获取最佳阈值, 并通过类簇上、下近似集的模糊类内聚集度和模糊类间分离度构造适应度函数; 最后给出算法的详细描述及复杂度分析.

2.1 混合蛙跳算法

SFLA 算法的基本原理.

Step 1: 随机产生初始蛙群 $F=\{Y_1,Y_2,\cdots,Y_M\}$, $M=m\cdot n$, m 为子群数目, n 为子群内青蛙个体数目, 青蛙个体 $Y_k=\{Y_{k1},Y_{k2},\cdots,Y_{kd}\}$, d 为解空间的维数.

Step 2: 计算青蛙个体的适应度值 $f(i)$, 并按适应度值降序排列青蛙个体.

Step 3: 按式(5)划分蛙群, 确定群体适应度最好青蛙 Y_g 以及子群内适应度最差和最好青蛙 Y_w 和 Y_b.

$$Z^k=\{Y_{k+m(e-1)}\in F|1\leqslant e\leqslant n\},\ 1\leqslant k\leqslant m; \quad (5)$$

$$D_j=\mathrm{rand}(\,)\cdot(Y_b-Y_w); \quad (6)$$

$$Y_w=Y_w(\mathrm{current})+D_j,\ -D_{\max}\leqslant D_j\leqslant D_{\max}. \quad (7)$$

其中: $D_{\max}$ 为最大移动距离, rand( ) 为 $[0,1]$ 内随机数.

Step 4: 子群内部执行式(6)和(7), 若新个体 Y_w 优于当前最差青蛙个体, 则用 Y_w 取代原子群最差青蛙; 若没有改进, 则用 Y_g 代替 Y_b, 并重新执行式(6)和(7)更新当前子群最差青蛙; 若还没有改进, 则随机生成青蛙 W 代替原最差青蛙 Y_w, 子群内最大迭代次数 T_{per}.

Step 5: 子群局部搜索完成后, 合并青蛙子群, 按适应度值降序排列青蛙群体, 并更新群体适应度最好青蛙 Y_g. 若算法收敛, 则停止搜索, 否则返回 Step 3, 直到算法收敛到最优解.

2.2 阴影集

阴影集理论通过三值逻辑映射保留核心模糊信息, 本质上是对模糊集概念的扩展. 假定 X 是模糊集, 阴影集将 X 中的每个元素映射到三值逻辑空间 {0,(0,1), 1}, 0 表示样本不属于 X, 1 表示样本属于 X, (0,1) 表示样本可能属于 X, 也可能不属于 X, 其构成集合的阴影区域. 阴影集的直观表示如图 1 所示.

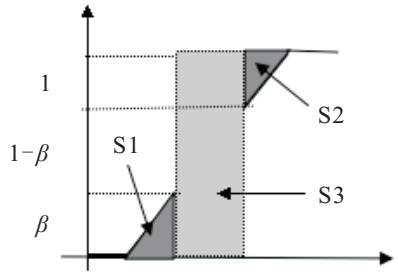

图 1 阴影集直观表示

设 $f(x)$ 为模糊集 X 的隶属度函数, 目标函数可以表示成隶属度函数的积分形式, 最小化 $V(\beta)$ 得到最优解 β.

$$V(\beta)=\left|\int_{-\infty}^{\beta}f(x)\mathrm{d}x+\int_{1-\beta}^{\infty}(1-f(x))\mathrm{d}x-\int_{\beta}^{1-\beta}\mathrm{d}x\right|, \quad (8)$$

其中 $\beta\in(0,0.5)$. 式(8)中的 3 项分别对应图 1 中 S1、S2 和 S3 三个划分区域, S1 为隶属度值减小的区域, S2 为隶属度值变大的区域, S3 为产生阴影集的区域.

2.3 改进的粗糙模糊聚类算法

2.3.1 自适应调节因子

经典混合蛙跳算法中, 当前位置和上次移动距离在进化过程中不具备记忆功能, 算法局部更新和全局收敛速度较慢. 本文在传统距离更新公式中增加上一次移动距离, 并引入调节因子 w_1 和 w_2 控制算法搜索方向. 其中: w_1 表示上一次移动距离在新移动距离中的保留值, 大小表示对上一次距离的继承多少; w_2 表示当前位置的惯性因子, 用来确定当前位置对新位置的贡献大小. 改进的移动距离和最差青蛙更新公式分别为

$$D_j=w_1D_j+\mathrm{rand}(\,)\cdot(Y_b-Y_w), \quad (9)$$

$$Y_w=w_2Y_w(\mathrm{current})+D_j. \quad (10)$$

w_1 和 w_2 随迭代次数自适应调整为

$$w_1=w_1^{\min}+(w_1^{\max}-w_1^{\min})((T-t)/T), \quad (11)$$

$$w_2=w_2^{\max}-(w_2^{\max}-w_2^{\min})\times\mathrm{e}^{-t(kf(Y_g)/f(Y_w))}. \quad (12)$$

其中: T 为子群总迭代次数, t 为子群当前迭代次数, $k>0$ 为最优青蛙与最差青蛙之间的平衡因子, $w_1\in[w_1^{\min},w_1^{\max}]$, $w_2\in[w_2^{\min},w_2^{\max}]$, $f(Y_g)$ 为全局最优

适应度, $f(Y_w)$ 为子群最差青蛙适应度. 如果采用式 (9) 和 (10) 更新之后的青蛙适应度优于原青蛙个体, 则用更新之后的青蛙代替原个体; 否则, 用 Y_g 代替 Y_b, 并执行式 (6) 和 (7) 更新当前子群最差青蛙. 如果还没有改进, 则随机生成一只青蛙 W 代替原 Y_w. 通过对当前位置和上一次移动距离的记忆, 使混合蛙跳算法具有一定的学习能力, 因而算法的寻优能力更强.

2.3.2　编码方式

本文采用浮点数方式进行编码. 在聚类过程中, 聚类的核心是找到一组最优聚类中心, 因此本文选取聚类中心编码青蛙个体. 设聚类数目为 C, 样本维数为 D, 青蛙个体 x 可以用 $C \times D$ 维的向量表示, $x = \{v_1, v_2, \cdots, v_C\}$, v_k 为第 k 类的聚类中心, 且 $v_k = \{v_{k1}, v_{k2}, \cdots, v_{kD}\}$ 是 D 维特征向量, 每只青蛙个体 x 代表问题的一个可行解.

2.3.3　阴影集获取阈值参数

设类 C_k 中各样本的隶属度值为 $\mu_{1k}, \mu_{2k}, \cdots, \mu_{Nk}$, 最大和最小隶属度值为 $\mu_{\max k}$ 和 $\mu_{\min k}$, 则目标函数表示为满足条件的相应隶属度值之和, 即

$$V(\beta) = |S_1 + S_2 - S_3|. \tag{13}$$

其中

$$S_1 = \sum_{i,\mu_{ik}<\beta_k} \mu_{ik}, \tag{14}$$

$$S_2 = \sum_{i,\mu_{ik}>\mu_{\max k}-\beta_k} (1-\mu_{ik}), \tag{15}$$

$$S_3 = \mathrm{card}(\mu_{ik} | \beta_k \leqslant \mu_{ik} \leqslant \mu_{\max k} - \beta_k). \tag{16}$$

β 的取值范围为 $[\mu_{\min k}, (\mu_{\min k} + \mu_{\max k})/2]$. S_1 为不属于类 C_k 的样本隶属度之和; S_2 为完全属于类 C_k 的样本隶属度之和; S_3 为可能属于也可能不属于类 C_k 的样本数量. 最小化 $V(\beta)$, 得到类 C_k 的最优阈值 β_k.

2.3.4　适应度函数构造

混合蛙跳算法中, 适应度函数是调节青蛙个体向最佳食物位置移动的唯一依据. RFCM 算法的类簇结构不仅与下近似区域有关, 而且与边界区域也密切关联, 每只青蛙实质上产生数据集的一种覆盖, 每个类簇用上、下近似集表示. 准则函数为

$$f = \frac{w_l \sum_{k=1}^{C} \sum_{x_i \in \underline{C_k}} \mu_{ik}^{m_1} d_{ik}^2 + w_b \sum_{k=1}^{C} \sum_{x_i \in C_k^B} \mu_{ik}^{m_1} d_{ik}^2}{\sum_{i=1}^{C} \sum_{j=1}^{C} \|v_i - v_j\|^2}. \tag{17}$$

其中: $d_{ik} = \|x_i - v_k\|$; 分子中 $\sum_{k=1}^{C} \sum_{x_i \in \underline{C_k}} \mu_{ik}^{m_1} d_{ik}^2$ 为模糊下近似集紧密度; $\sum_{k=1}^{C} \sum_{x_i \in (\overline{C_k} - \underline{C_k})} \mu_{ik}^{m_1} d_{ik}^2$ 为模糊边界集紧密度; 分母 $\sum_{i=1}^{C} \sum_{j=1}^{C} \|v_i - v_j\|^2$ 为所有类簇的类间分离度; w_l 和 w_b 平衡下近似集与边界集的重要性. 式 (17) 的分子越小, 类簇聚集程度越高, 分母越大, 类簇的分离程度越大. 当 f 取最小值时, 聚类效果最好.

2.4　SFLA-SRFCM 算法描述

本文算法的基本思想: 将改进混合蛙跳算法与粗糙模糊 C 均值聚类算法相结合, 通过阴影集选取合适的阈值, 将数据集划分成若干个下近似集和边界集, 并设置权重参数调节下近似集和边界集的重要性. 所提出的算法同时解决了初值敏感和参数选取对聚类性能的影响. SFLA-SRFCM 算法描述如下.

输入: 数据集 X 和聚类数目 C;

输出: 最优阈值、类簇中心、下近似集和边界集.

Step 1: 设置初始参数. 子群个数 m, 子群青蛙个数 n, 青蛙群体总数 $M = m \cdot n$, 蛙群最大迭代次数 T_{total}, 子群内最大迭代次数 T_{per}, 模糊指数 m_1.

Step 2: 蛙群初始化. 在数据集 X 中随机选取 C 个样本作为初始聚类中心, 将选取的聚类中心按照编码规则生成青蛙个体的初始位置. 反复执行 M 次, 生成初始种群 $F_1, F_2, \cdots, F_M$, F_i 代表问题的一个可行解.

Step 3: 根据式 (3) 计算每只青蛙对应的隶属度矩阵 $U^{(t)}$, t 为当前迭代次数.

Step 4: 针对每只青蛙个体, 基于青蛙对应的隶属度矩阵, 利用阴影集最优化目标函数 (13) 获取每个类簇的最优阈值 β_k, $k = 1, 2, \cdots, C$.

Step 5: 根据阈值 β_k, 利用下两式确定每个类簇 C_k $(k = 1, 2, \cdots, C)$ 的下近似集和模糊边界集:

$$\underline{C_k} = \{x_i | u_{\max k} - u_{ik} \leqslant \beta_k\}, \tag{18}$$

$$C_k^B = \{x_i | \beta_k < u_{ik} < u_{\max k} - \beta_k\}. \tag{19}$$

Step 6: 根据式 (17) 计算群体 F 中每只青蛙的适应度值 $f(i)$ $(i = 1, 2, \cdots, M)$, 并按适应度值从小到大排列青蛙个体, 确定蛙群最优青蛙 y_g 和隶属度矩阵 U_0.

Step 7: 利用式 (5) 进行子群划分, 依据适应度值确定每个子群中最差青蛙 y_w 和最好青蛙 y_b, 利用改进后的子群更新方式 (9) 和 (10) 更新每个子群最差青蛙 y_w, 直到迭代次数为子群内最大迭代 T_{per} 为止, 将更新后的青蛙子群混合.

Step 8: 对于混合后的每只青蛙, 根据式 (4) 计算新的聚类中心, 并计算每只青蛙的适应度函数值, 确

定新蛙群最优青蛙 y_g 和隶属度矩阵 U_g.

Step 9: 如果 $|U_g - U_0| < \varepsilon$, 则算法结束, 输出最优阈值、类簇中心、类簇下近似集和边界集; 否则, 转入Step 3, $t = t + 1$.

2.5 时间复杂度分析

设样本总数为 N, 算法收敛时最大迭代次数为 T, 种群规模为 M. 对于每只青蛙个体, 计算其划分矩阵的时间复杂度为 $O(CN)$ (Step 3); 获取每只青蛙所有类簇阈值的时间复杂度为 $O(CL)$ (Step 4), L 为最小化式 (13) 时 β_k 的取值个数, $L < N$; 计算类簇下近似集与边界集时间复杂度为 $O(CN)$ (Step 5); 完成所有子群操作的时间复杂度为 $O(T_{\rm per}CNM)$ (Step 3 ~ Step 7), 更新所有青蛙个体聚类中心的时间复杂度为 $O(CM)$ (Step 8), 则总的时复杂度为 $O(T_{\rm per}CNMT + CMT) = O(T_{\rm per}CNMT)$. SFLA-SRFCM 算法与种群规模密切相关, 当种群规模较小时, 与 SRFCM 算法时间复杂度相同; 当种群规模较大时, 改进算法比 SRFCM 算法的运行时间多一些, 但其聚类效果明显提高.

3 实验结果与分析

本文所有测试均在 Pentium IV, 2.8 GHz, 32 GB RAM 计算机, Matlab 7.1.0 平台上仿真实现. 本文进行了两个实验, 第 1 个是人造数据实验, 用来验证算法的有效性; 第 2 个是 UCI 数据集上的实验, 用来验证算法对真实数据的处理能力. 同时, 本文将它与 FCM 模糊聚类算法、粗糙模糊聚类 (RFCM) 算法、文献 [6] 提出的阴影集粗糙模糊聚类 (SRFCM) 算法以及文献 [10] 给出的基于 SFLA 的 K-Means 聚类进行对比分析.

相关参数设置如下: 子群个数 $m = 20$, 每个子群包含青蛙个数 $n = 10$, 青蛙群体总数 $M = m \cdot n = 200$, 蛙群最大迭代次数 $T_{\rm total} = 100$, 子群内最大迭代次数 $T_{\rm per} = 50$, 模糊指数 $m_1 = 2$, $w_l = 0.85$, $w_b = 0.15$, 算法终止阈值 $\varepsilon = 0.000\,001$. $w_1^{\min} = w_2^{\min} = 0.4$, $w_1^{\max} = w_2^{\max} = 1$. 利用阴影集计算类簇阈值时, β 在可行域内以等间距 0.1 选取若干个点作为候选阈值, 从中选取使式 (13) 取最小值的解作为类簇的最佳阈值. 算法均运行 20 次, 取均值为最终结果.

3.1 人造数据集实验

人造数据集 Dataset 是满足高斯分布的二维数据集, 包含 100 个样本点, 分为 3 类, 类簇 1 和类簇 2 均有 40 个样本点, 类簇 3 有 20 个样本点, 类簇部分元素交叉, 数据分布如图 2 所示. 图 3 ~ 图 6 为 FCM、RFCM、SRFCM 和 SFLA-SRFCM 算法产生的近似区域和中心点分布图, SFLA-KM 算法聚类效果图与 FCM 相同.

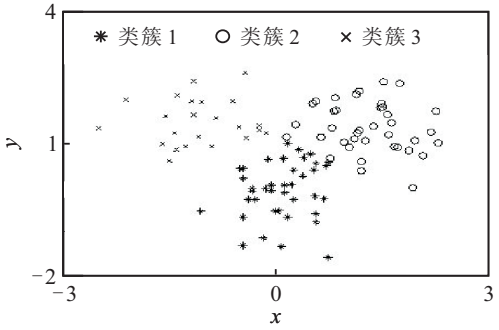

图 2 人造数据集 Dataset 的数据分布

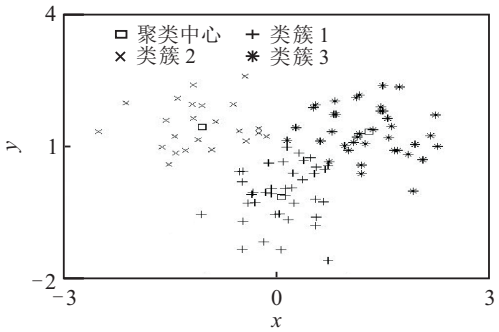

图 3 FCM 算法的聚类结果

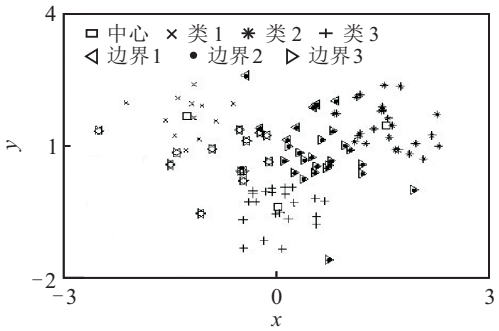

图 4 RFCM 算法的聚类结果

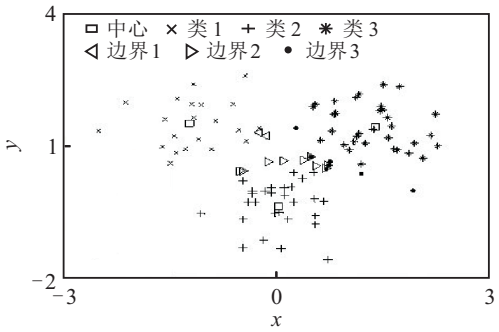

图 5 SRFCM 算法聚类中心和边界分布

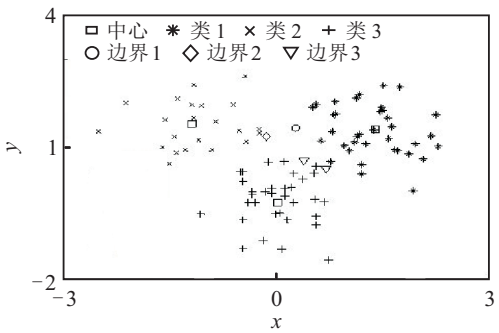

图 6 SFLA-SRFCM 算法聚类中心和边界分布

从图4可以看出, RFCM可以得到较好的聚类中心, 但被划入边界区域的样本较多, 部分不该归入边界集的样本被错误地划分到边界集, 聚类的不确定程度较高.

图5只有小部分样本划入边界区域, 但边界2中有些本来应该划分到下近似集的却也被划分到了边界集; 图6边界样本明显减少, 得到的聚类中心更符合实际情况. 由此可知: 通过阴影集获取聚类阈值的同时, 采用混合蛙跳算法优化初始聚类中心, 可以有效降低算法被错分的概率. SFLA-SRFCM算法通过引入混合蛙跳算法优化初始聚类中心, 使得算法对任意初始值都可以快速搜索到最优聚类结果, 提高了算法的聚类性能.

聚类准确率为正确分类样本数与样本总数的比值. 本文算法平均正确分类的样本数为194, 聚类准确率为194/200 = 0.97; FCM、RFCM、SRFCM算法平均正确分类的样本数均为180, 聚类准确率均为180/200 = 0.90. 从图7可以看出, SFLA-SRFCM算法的寻优性能明显优于其他4种聚类算法, 可以在第3次迭代时即收敛到全局最优.

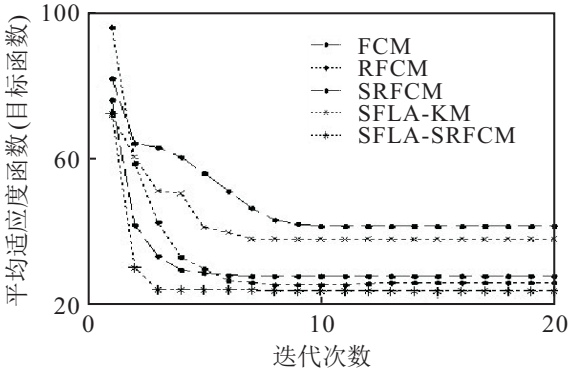

图 7　人造数据适应度函数随迭代次数的变化曲线

3.2　UCI标准数据集实验

4组UCI数据集分别为Iris、Wine、Ionosphere和Isolet5. Iris为包含150个样本的4维数据集, 样本被分成3类, 其中两类样本部分交叉, 该数据集是检验聚类算法性能最常用的标准数据. Wine为包含178个样本的7维数据集, 该数据集3个类分布较为集中, 用于验证算法对复杂分布数据的聚类能力. Ionosphere和Isolet5为较高维数的数据集, Ionosphere包含351个样本, 样本维数34, Isolet5包含1 559个样本, 样本维数617, 这两组数据均用来验证算法对高维数据的处理能力. 数据详细描述如表1所示.

在实验测试过程中, 本文采用Davies-Bouldin (DB)[11]和Calinski-Harabasz(CH)[12]指标考察算法的有效性, 并使用准确率评价算法的聚类质量. 其中: DB指标为两两类簇的紧致性与分离性比值的最大值的平均值, DB值越小, 聚类效果越好; CH指标为所有样本之间的距离平方和与所有类内样本之间的距离平方和的差与类内样本之间距离平方和的比值, CH越大, 聚类结果越准确. 5种算法在不同数据集上的聚类结果比较如表2～表5所示.

表 1　UCI实验数据集

名称	样本数	属性维数	类簇个数
Iris	150	4	3
Wine	178	7	3
Ionosphere	351	34	2
Isolet5	1 559	617	26

表 2　Iris数据集上的聚类结果比较

算法	DB	CH	准确率	时间/s
FCM	0.698 7	356.979 6	0.893 3	0.45
RFCM	0.647 0	348.715 1	0.906 7	0.51
SRFCM	0.700 6	356.564 4	0.913 3	0.62
SFLA-KM	0.688 1	359.845 1	0.900 0	13.79
本文算法	**0.629 7**	**515.270 2**	**0.946 7**	**1.28**

表 3　Wine数据集上的聚类结果比较

算法	DB	CH	准确率	时间/s
FCM	1.113 2	83.135 1	0.949 4	0.53
RFCM	1.099 1	83.256 5	0.966 3	0.38
SRFCM	1.294 6	83.265 6	0.971 9	0.57
SFLA-KM	1.106 0	83.263 4	0.966 3	14.95
本文算法	**1.092 0**	**83.356 7**	**0.977 5**	**7.61**

表 4　Ionosphere数据集上的聚类结果比较

算法	DB	CH	准确率	时间/s
FCM	1.530 6	117.524 0	0.689 4	0.26
RFCM	1.527 4	117.642 1	0.700 9	0.43
SRFCM	1.524 0	118.221 6	0.712 3	0.57
SFLA-KM	1.511 4	118.817 9	0.709 4	25.35
本文算法	**1.498 2**	117.654 1	**0.763 7**	**8.59**

表 5　Isolet5数据集上的聚类结果比较

算法	DB	CH	准确率	时间/s
FCM	2.150 6	49.456 2	0.606 0	12.37
RFCM	2.077 2	52.580 7	0.682 5	19.85
SRFCM	1.907 7	52.877 8	0.702 7	23.46
SFLA-KM	1.554 5	54.731 5	0.704 1	43.93
本文算法	**1.529 4**	**66.256 0**	**0.738 9**	**26.82**

从表2～表5可以看出, 基于混合蛙跳与阴影集的粗糙模糊聚类算法比其他算法在聚类性能上都有较大提高. 在Iris和Wine数据集上, SFLA-SRFCM算法DB值最小, CH值最大, 在这两个数据集上聚类准确率比FCM算法分别提高了5.34 %和2.81 %. 在Ionosphere数据集上, 新算法的DB指标和准确率最高, 但CH指标值比SRFCM算法和SFLA-KM算法略小. 这是因为Ionosphere数据集非线性分离, 而CH指

标主要度量类内紧致性, 算法收敛时类间分离性不一定达到最大, 本文算法为了使类分离程度更大, 将部分边界样本划分到内层球, 总的类内距离减小, 因而CH值稍小一些. CH指标值略小并不影响本文算法的聚类效果, 而且本文算法在所有测试的数据集上聚类准确率都明显提高了. 在维数较高的Isolet5数据集上, 本文算法的DB值最小, CH值最大, 聚类准确率最高, 即使对于较高维数的数据集, 本文算法也可以获得较好的聚类性能.

从表2～表5还可以看出, 尽管新算法相对于FCM、RFCM和SRFCM算法的运行时间要长, 但是新算法相对于其他混合蛙跳优化的聚类算法, 运行时间要少. 这是因为通过阴影集获取聚类阈值之后, 算法可以很快计算出正确的下近似集和边界集, 从而快速搜索到最佳聚类中心. 这表明, 通过阴影集获取聚类阈值的同时, 使用混合蛙跳策略优化算法的初始聚类中心, 可以更好地描述聚类结果, 有效提高算法的聚类质量.

4 结 论

本文提出了一种基于混合蛙跳与阴影集的粗糙模糊聚类算法(SFLA-SRFCM). 所提出的算法结合了SFLA的搜索特性和阴影集不确定知识的处理特点, 有效解决了粗糙模糊聚类算法初始聚类中心敏感和阈值选择的问题, 提高了算法的适应性和鲁棒性. 通过人造数据集和UCI标准数据集实验表明, 所提出的算法具有更好的聚类性能, 但还需进一步研究算法的扩展性以及粗糙模糊聚类算法对大规模非完备数据的处理能力.

参考文献(References)

[1] Jain A K. Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[2] Mitra S, Banka H, Pedrycz W. Rough fuzzy collaborative clustering[J]. IEEE Trans on Systems, Man, and Cybernetics, PartB: Cybernetics, 2006, 36(4): 795-805.

[3] Maji P, Pal S K. RFCM: A hybrid clustering algorithm using rough and fuzzy sets[J]. Fundamenta Informaticae, 2007, 80(4): 475-496.

[4] 姚丽娟, 罗可. 基于粒子群的粗糙核聚类算法[J]. 计算机应用研究, 2012, 29(8): 2854-2857.
(Yao L J, Luo K. Rough kernel clustering algorithm based on particle swarm optimization[J]. Application Research of Computer, 2012, 29(8): 2854-2857.)

[5] 王学恩, 韩德强, 韩崇昭. 采用不确定性度量的粗糙模糊C均值聚类参数获取方法[J]. 西安交通大学学报, 2013, 47(6): 55-60.
(Wang X E, Han D Q, Han C Z. A selection method for parameters of rough fuzzy C-means clustering based on uncertainty measurement[J]. J of Xi'an Jiaotong University, 2013, 47(6): 55-60.)

[6] Zhou J, Pedrycz W, Miao D. Shadowed sets in the characterization of rough-fuzzy clustering[J]. Pattern Recognition, 2011, 44(8): 1738-1749.

[7] Peters G. Rough clustering utilizing the principle of indifference[J]. Information Sciences, 2014, 277(2): 358-374.

[8] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. J of Water Resources Planning and Management, 2003, 129(3): 210-225.

[9] Lingras P, West C. Interval set clustering of web users with rough k-means[J]. J of Intelligent Information Systems, 2004, 23(1): 5-16.

[10] Amiri B, Fathian M, Maroosi A. Application of shuffled frog-leaping algorithm on clustering[J]. The Int J of Advanced Manufacturing Technology, 2009, 45(1/2): 199-209.

[11] Davies D L, Bouldin D W. A cluster separation measure[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1979, 1(2): 224-227.

[12] Calinski T, Harabasz J. A dendrite method for cluster analysis[J]. Communications in Statistics, 1974, 3(1): 1-27.

(责任编辑: 齐 雾)