

## 混合属性数据集的聚类边界检测技术

李向丽, 耿鹏, 邱保志

(郑州大学 信息工程学院, 郑州 450001)

**摘要:** 为了满足数据分析中获取含有混合属性的数据集聚类的边界需求, 提出一种混合属性数据集的聚类边界检测算法(BERGE)。该算法利用模糊聚类隶属度定义边界因子以识别候选边界集, 然后运用证据积累的思想提取聚类的边界。在综合数据集和真实数据集上的实验结果表明, BERGE算法能有效地检测混合属性数据集、数值属性数据集以及分类属性数据集的聚类边界, 与现有同类算法相比具有更高的精度。

**关键词:** 混合属性; 高维数据; 聚类边界; 边界因子; 证据积累

**中图分类号:** TP311

**文献标志码:** A

### Clustering boundary detection technology for mixed attribute data set

LI Xiang-li, GENG Peng, QIU Bao-zhi

(School of Information and Engineering, Zhengzhou University, Zhengzhou 450001, China. Correspondent: GENG Peng, E-mail: gengpengtianxia@163.com)

**Abstract:** To meet with the need of extracting clustering boundary from mixed attribute data set in field of data analysis, a clustering boundary detection algorithm for the mixed attribute data set is proposed, named BERGE. BERGE defines the boundary factor based on the membership of fuzzy clustering to recognition the candidate boundary set, and utilizes the idea of evidence accumulation to extract clustering boundary from the candidate boundary set. The experimental results on synthetic data sets and real data sets show that BERGE can effectively detect clustering boundary of mixed attribute data sets, numerical attribute data sets and categorical attribute data sets with higher accuracy compared with the existing similar algorithms.

**Keywords:** mixed attribute; high-dimensional data; clustering boundary; boundary factor; evidence accumulation

### 0 引言

聚类分析是数据挖掘中非常活跃的研究方向之一, 其目标是将相似的对象尽可能地聚集在一起, 而相异的对象尽可能地分离, 形成一个个聚集簇<sup>[1]</sup>。它广泛地应用于图像处理、信息检索、气象学、商业等领域。聚类的边界点位于簇的边沿, 它们通常具有多个簇的聚类特征, 其归属并不明确<sup>[2]</sup>。准确识别聚类边界对提高聚类精度和研究聚类特征具有重要作用。因此, 聚类的边界点检测在疾病防治、医药研制、虚拟现实等领域具有重要的研究价值和应用价值。

DBSCAN(Density-based spatial clustering of applications with noise)算法<sup>[3]</sup>基于密度定义了聚类边界点的概念, 但是并没有给出获取完整聚类边界的方法。Xia等<sup>[2]</sup>提出的BORDER边界点检测算法利用核心点的反向 $K$ 近邻个数大于边界点的反向 $K$ 近邻个

数这一特性来检测聚类的边界, 该算法对于不含噪声的数值属性数据集具有良好的边界检测结果, 但不能区分噪声点和边界点。BRIM算法<sup>[4]</sup>依据数据点在其正负半领域内分布不均匀的特点来获取聚类的边界, 解决了BORDER算法不能区分边界点和噪声点的问题, 但是BRIM算法仍然有可能将靠近簇的噪声点误认为边界点。BDDTS算法<sup>[5]</sup>首先根据双阈值形成候选边界集, 然后对其进行二次处理获取边界点。虽然BDDTS算法使用多次处理策略能够提高边界检测的精度, 但是需要多个输入参数。为了进一步提高聚类边界检测的精度, 基于边界点的变异系数大于内部点及噪声点的变异系数这一事实, 薛丽香等<sup>[6]</sup>提出了BAND算法, 但是该算法不能对分类属性数据集及混合属性数据集进行有效的聚类边界检测。

CBORDER算法<sup>[7]</sup>是针对上述边界检测算法不

收稿日期: 2013-09-15; 修回日期: 2014-03-13.

基金项目: 河南省重点科技攻关项目(112102310073).

作者简介: 李向丽(1965—), 女, 副教授, 从事计算机网络与数据挖掘等研究; 耿鹏(1990—), 男, 硕士生, 从事数据挖掘的研究。

能用于高维分类属性数据集的问题而提出的,该算法利用随机分配初始聚类中心和边界度对类进行划分并获取记录边界点的证据,然后运用证据积累的思想获得聚类的边界. CBORDER 采用证据积累能有效地消除随机分配初始聚类中心带来的误差,同时能够检测出高维分类属性数据集的聚类边界. 但是边界度的定义只考虑了二分类情况,没有考虑数据集存在多类的情况.

针对上述聚类边界点检测算法均不能处理混合属性数据集的问题,本文提出一种能够有效检测混合属性数据集聚类边界的算法 (BERGE). 该算法引入模糊聚类的思想,根据数据对象的隶属度定义边界因子,克服了当数据集存在多个簇时 CBORDER 算法利用二分类定义边界度不够精确的问题,从而进一步提高了聚类边界检测的精度,而且能够有效地检测数值属性数据集、分类属性数据集以及混合属性数据集的聚类边界.

## 1 BERGE 算法

给定一个混合属性数据集  $D$ , 其属性集为  $\{A_i | i = 1, 2, \dots, m\}$ , 其中分类属性和数值属性的个数分别为  $m_C$  和  $m_N$ , 不妨设分类属性位于数值属性之前,用  $S_i$  表示第  $i$  个属性的取值集合或区间. 同时用  $r$  表示一个数据对象,用  $r_{A_i}$  表示数据对象  $r$  在  $A_i$  属性上的取值. 对于任意的  $r \in D$ , 有  $r \in \prod_{i=1}^m S_i$ .

### 1.1 相关概念

**定义 1** (支持度)<sup>[8]</sup> 给定类  $C$ , 分类属性  $A_i$  取值  $a_i \in S_i$ , 用  $C|S_i$  表示簇  $C$  在属性取值集合  $S_i$  上的投影, 则  $a_i$  在  $C$  中关于属性  $A_i$  的支持度定义为  $C|S_i$  中包含  $a_i$  的次数, 即

$$\text{Sup}_{(C|S_i)}(a_i) = |\{r | r \in C, r_{A_i} = a_i\}|. \quad (1)$$

**定义 2** (对象之间的距离)<sup>[8]</sup> 数据对象  $p, q$  在属性  $A_i$  上的距离  $d(p, q)$  定义为每个属性上距离的平均值. 对于二值属性或分类属性, 有

$$\text{dis}(p_{A_i}, q_{A_i}) = \begin{cases} 1, & p_{A_i} = q_{A_i}; \\ 0, & p_{A_i} \neq q_{A_i}; \end{cases} \quad (2)$$

对于连续数值属性, 有

$$\text{dis}(p_{A_i}, q_{A_i}) = |p_{A_i} - q_{A_i}|; \quad (3)$$

则  $p$  与  $q$  之间的距离

$$d(p, q) = \frac{\sum_{i=1}^m \text{dis}(p_{A_i}, q_{A_i})}{m}. \quad (4)$$

**定义 3** (对象到簇的距离)<sup>[8]</sup> 给定类  $C$ ,  $C$  的描述信息 (CDI) 定义为由分类属性中不同取值的统计频度和数值属性的质心组成, 即

$$\begin{aligned} \text{CDI} &= \{ \langle \text{Cla}_i, \text{Num} \rangle | \text{Cla}_i = \\ & \{ (a_i, \text{Sup}_{C|S_i}(a_i)) | a_i \in S_i \}, 1 \leq i \leq m_C, \\ & \text{Num} = (A_{m_C+1}, \dots, A_{m_C+m_N}) \}. \end{aligned} \quad (5)$$

数据对象  $p$  到簇  $C$  的距离  $d(p, C)$  定义为  $p$  与类  $C$  的描述信息之间的距离, 即

$$d(p, C) = \sum_{i=1}^m \text{dis}(p_{A_i}, C|S_i), \quad (6)$$

其中  $\text{dis}(p_{A_i}, C|S_i)$  表示  $p$  与簇  $C$  在属性  $A_i$  上的距离.

对于分类属性, 其值定义为 1 减去  $p$  与  $C$  中每个对象在属性  $A_i$  上的距离的均值, 即

$$\text{dis}(p_{A_i}, C|S_i) = 1 - \frac{\text{Sup}_{C|S_i}(p_{A_i})}{|C|}. \quad (7)$$

$p_{A_i}$  在簇  $C$  中的支持度越大, 说明  $p$  与  $C$  在属性  $A_i$  上的距离越小, 且取值范围为  $[0, 1]$ .

对于数值属性, 则定义为  $p$  与  $C$  在属性  $A_i$  上的距离与该属性取值区间之比, 即

$$\text{dis}(p_{A_i}, C|S_i) = \frac{|p_{A_i} - C_{A_i}|}{\max(S_i) - \min(S_i)}. \quad (8)$$

$p$  与  $C$  在属性  $A_i$  上的差值越小, 说明它们之间的距离越小, 且取值范围为  $[0, 1]$ .

**定义 4** (对象到簇的隶属度)<sup>[1]</sup> 数据对象  $r$  到  $k$  个簇的隶属度  $u_i$  定义如下:

$$u_i = \frac{1}{\sum_{j=1}^k \frac{1}{d(r, C_j)}}, 1 \leq i \leq k, \text{ 且满足 } \sum_{i=1}^k u_i = 1. \quad (9)$$

对象  $r$  到簇  $C_i$  的距离越小,  $r$  属于  $C_i$  的隶属度越大; 反之隶属度越小.

**定义 5** (边界因子)<sup>[9]</sup> 依据数据记录  $r$  到  $k$  个簇的距离和隶属度,  $r$  的边界因子 BOF 定义为

$$\text{BOF}(r) = \left( - \sum_{i=1}^k u_i \log(u_i) \right) \times \left( \sum_{i=1}^k u_i d(r, C_i) \right). \quad (10)$$

其中:  $u_i$  表示对象  $r$  属于簇  $C_i$  的隶属度,  $d(r, C_i)$  表示  $r$  与  $C_i$  之间的距离. 边界因子的第 1 部分使用熵确定对象  $r$  到各个簇的不确定性. 数据对象越靠近簇的边沿, 其隶属度的熵越大; 反之越小. 第 2 部分是对象  $r$  到所有簇的平均距离, 数据对象越靠近簇的内部, 其平均距离越小; 反之越靠近簇的边沿, 其平均距离越大. BOF( $r$ ) 值越大, 对象  $r$  越有可能是簇的边界点. 对于含有噪声的数据集, BOF 取值最大的若干个对象可能是噪声点, 在提取聚类边界时应剔除.

### 1.2 算法描述

BERGE 算法利用证据积累的思想, 将获取聚类边界分为两个阶段. 第 1 阶段是证据积累. 首先随机初始化  $k$  个簇质心; 然后利用模糊聚类的思想计算每

个对象到各个簇的距离和隶属度, 根据隶属度将数据对象划分到所属隶属度最大的簇中, 并更新相应簇的描述信息; 最后, 在每个数据对象划分完成之后, 根据各个对象的隶属度计算其边界因子, 将边界因子中最大的  $w \times n$  个对象作为聚类边界进行标记, 并将其边界证据增加 1, 此过程需要迭代  $\lfloor \sqrt{n} \rfloor$  次, 即记录边界点的证据。其中: 参数  $w$  是边界比例, 其取值代表边界点在数据集中占有的比例大小;  $n$  是数据集中的对象个数。

第 2 个阶段是边界获取。对数据对象的边界证据进行遍历, 将具有边界证据个数大于  $\beta \times \sqrt{n}$  的对象作为边界点输出。其中, 边界阈值参数  $\beta$  控制边界点的多少, 表示边界点在候选边界集中所占的比例大小。

BERGE 算法描述如下。

输入: 数据集  $D$ , 簇个数  $k$ , 边界比例  $w$ , 边界阈值  $\beta$ ;

输出: 数据集的边界点集合。

Step 1: 证据积累。迭代执行  $\lfloor \sqrt{n} \rfloor$  次。

Step 1.1: 随机初始化  $k$  个簇质心, 并更新初始簇的描述信息。

Step 1.2: 从剩余的数据集中选择一个数据对象, 根据定义 3 计算对象到各个簇的距离, 并根据定义 4 计算其隶属度。将对象划分到具有最大隶属度的簇中, 同时更新相应簇的描述信息。

Step 1.3: 若有剩余数据对象, 则转 Step 1.2, 否则转 Step 1.4。

Step 1.4: 根据定义 5 计算每个数据对象的边界因子, 对边界因子中最大的前  $w \times n$  个对象进行标记, 其边界证据增加 1。

Step 2: 边界获取。遍历每个数据对象的边界证据, 将值大于  $\beta \times \sqrt{n}$  的对象作为边界点输出。

在 Step 1.2 中, 当有数据对象  $r$  合并到簇  $C_j$  中时, 要对  $C_j$  的描述信息进行更新。假定对象  $r$  的属性  $A_i$  取值为  $a_i$ , 对于分类属性, 应对簇  $C_j$  的描述信息执行  $\text{Sup}_{C_j|S_i}(a_i)$  增加 1; 对于数值属性, 则根据

$$C_j|S_i = \left( \sum_{r \in C_j} u_j \times r_{A_i} \right) / \sum_{r \in C_j} u_j$$

重新计算簇  $C_j$  在属性  $A_i$  上的质心。

若数据集  $D$  中包含噪声点, 则在 Step 1.4 中, 对边界因子取值最大的前  $w \times n$  个对象进行标记时, 可将 BOF 取值最大的若干个对象作为噪声点剔除, 以提高 BERGE 算法检测边界点的精度。

## 2 实验结果及分析

实验环境: CPU 为 Intel(R) Core(TM) i3-2130 3.40 GHz, 内存为 2 GB, 操作系统为 Microsoft Windows 7,

算法编程环境为 Microsoft Visual Studio 2010。

### 2.1 实验结果

为了验证 BERGE 算法在混合属性数据集上的有效性, 本文首先在含有噪声的二维数值属性数据集上验证 BERGE 算法检测边界的能力; 然后在 UCI 数据集上与 CBORDER 算法进行边界检测能力的比较; 最后使用医学阿尔茨海默病真实数据集验证算法对混合属性数据集进行边界检测的能力。

图 1(a) 的数据集来源于文献 [10], 它是包含 7832 个点并带有噪声的数据集; 图 1(b) 是 BORDER 算法的结果 ( $k = 50, n = 500$ ); 图 1(c) 是 BRIM 算法的结果 ( $\text{Eps} = 60, 100$ ); 图 1(d) 是本文算法的运行结果 ( $k = 2, w = 0.25, \beta = 0.85$ )。图 2(a) 是含有大量噪声, 且不均匀的数据集, 来源于文献 [10], 共包含了 5034 个点; 图 2(b) 是 BORDER 算法的结果 ( $k = 120, n = 1200$ ); 图 2(c) 是 BRIM 算法的结果 ( $\text{Eps} = 40, 60$ ); 图 2(d) 是本文算法的运行结果 ( $k = 5, w = 0.2, \beta = 0.9$ )。可以看出, BERGE 算法利用证据积累的思想能够消除随机初始化簇质心带来的误差, 通过多次实验, 该算法可以有效地检测出聚类的边界点, 并且能够去除噪声。

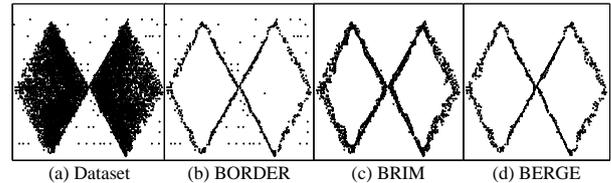


图 1 3 种算法的边界检测结果对比 (1)

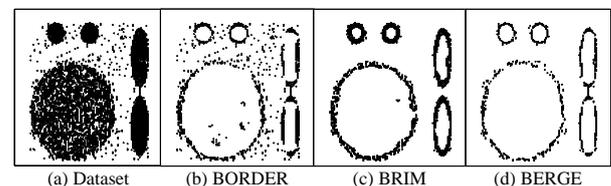


图 2 3 种算法的边界检测结果对比 (2)

对于分类属性数据集, 本文使用检出率、准确率、召回率、F-measure 来度量算法的有效性, 其定义如下:

检出率 = 检测边界个数 / 实际边界个数,

准确率 = 检测正确边界个数 / 检测边界个数,

召回率 = 检测正确边界个数 / 实际边界个数,

F-measure =  $2 / (1 / \text{准确率} + 1 / \text{召回率})$ 。

检出率表明算法检测出边界的能力, 检出率越大, 说明算法检测出边界的能力越强; 准确率表明算法辨识边界的能力, 准确率越大, 说明算法检测出边界的精度越高; 召回率表明算法检测出的边界是实际边界

的比率; F-measure 的值越大, 说明算法检测边界的性能越好.

采用的分类属性数据集是 CBORDER 算法使用的 UCI 数据集集中的 Mushroom 和 Soybean 数据集. Mushroom 数据集包含 8 124 条数据记录, 22 维分类属性. 包含了 2 个类 (有毒和无毒), 利用自然聚类获取的边界点个数是 2 075 个聚类边界点, 这些边界点表示难以区分是否有毒的蘑菇种类. CBORDER 算法采用的参数为:  $k = 2, \delta = 0.85, \beta = 0.25$ ; BERGE 算法使用的参数为:  $k = 2, w = 0.28, \beta = 0.84$ . Soybean 数据集包含 47 条数据记录和 35 维分类属性, 分为 4 个聚类, 利用自然聚类获取的边界点个数是 13 个边界点. CBORDER 算法采用的参数为:  $k = 2, \delta = 0.85, \beta = 0.05$ ; BERGE 算法使用的参数为:  $k = 4, w = 0.55, \beta = 0.8$ . 从表 1 和表 2 中的数据结果可以看出: BERGE 算法比 CBORDER 算法具有更高的准确率和召回率; F-measure 结果的对比说明对于分类属性数据集的聚类边界检测, BERGE 算法比 CBORDER 算法具有更好的检测性能.

表 1 Mushroom 数据集的边界点检测结果对比

检测结果	算 法	
	CBORDER	BERGE
实际边界个数	2 075	2 075
检测边界个数	1 563	1 672
检测正确边界个数	1 263	1 548
检出率/%	75.98	80.58
准确率/%	80.81	92.58
召回率/%	60.88	74.60
F-measure	0.694 4	0.826 2

表 2 Soybean 数据集的边界点检测结果对比

检测结果	算 法	
	CBORDER	BERGE
实际边界个数	13	13
检测边界个数	13	12
检测正确边界个数	12	12
检出率/%	100.00	92.31
准确率/%	92.31	100.00
召回率/%	92.31	92.31
F-measure	0.923 1	0.960 0

医学阿尔茨海默病数据集 Alzheimer 是郑州大学数据挖掘小组建立的标准混合属性数据集, 共有 219 个数据记录, 每个数据记录都具有 21 个属性, 其中, 前 17 个属性为分类属性, 后 4 个属性为数值属性. 有 143 人属于正常人, 其中包括 32 个可疑阿尔茨海默病患者; 阿尔茨海默病患者共有 76 人, 分为轻度痴呆、中度痴呆和重度痴呆. 这些可疑阿尔茨海默病患者即为需要检测的聚类边界点, 他们是具有阿尔茨海

默疾病特征但并没有患病的人群, 将这些边界人群检测出来进行分析可能会揭示该疾病更深层次的患病机理及易患病的潜在人群特征, 有助于对其进行及早的预防和医药治疗.

首先, 采用 BDDTS 算法对由后 4 个数值属性形成的数据集进行边界点检测, 参数取值为:  $k = 10, \lambda_1 = 0.5, \lambda_2 = 0.6$ ; 采用 CBORDER 算法对由前 17 个分类属性形成的数据集进行边界点检测, 参数取值为:  $k = 4, \delta = 0.78, \beta = 0.8$ . 然后, 利用 BERGE 算法对整个 Alzheimer 混合属性数据集进行边界点检测, 参数取值为:  $k = 4, w = 0.45, \beta = 0.9$ . 实验结果 (见表 3) 表明, BERGE 算法能够有效地检测混合属性数据集的聚类边界, 比 CBORDER、BDDTS 算法具有更好的边界检测性能.

表 3 Alzheimer 数据集的边界点检测结果对比

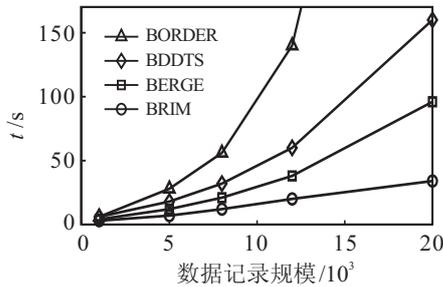
检测结果	算 法		
	CBORDER	BDDTS	BERGE
实际边界个数	32	32	32
检测边界个数	42	48	41
检测正确边界个数	21	26	30
检出率/%	131.25	150.00	128.13
准确率/%	50.00	54.17	73.17
召回率/%	65.63	72.22	93.75
F-measure	0.567 6	0.619 1	0.821 9

## 2.2 复杂度分析及效率对比

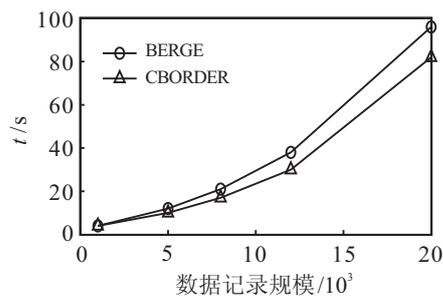
假定数据集  $D$  包含  $n$  个数据对象,  $m$  个属性, 且将  $D$  划分为  $k$  个簇. BERGE 算法的执行分为证据积累和边界获取两个阶段: 第 1 阶段需要执行一个  $\lfloor \sqrt{n} \rfloor$  次循环, 每次循环需要一个时间复杂度为  $O(kmn)$  的距离计算和一个  $O(kn)$  的边界因子计算以及  $O(w \times n \log(w \times n))$  的记录边界点证据, 因此第 1 阶段的时间复杂度为  $O(n^{3/2} \log n)$ ; 第 2 阶段需要遍历边界证据以获取数据集的边界点, 时间复杂度为  $O(n)$ . 综上所述, BERGE 算法总的时间复杂度为  $O(n^{3/2} \log n)$ , 因为  $n > 16$  时,  $\log n < \sqrt{n}$ , 所以 BERGE 算法的时间复杂度在  $(n^{3/2}, n^2)$  区间内.

为了验证 BERGE 算法的运行效率, 分别选用二维数值属性和高维分类属性的综合数据集, 选取数据规模从 1 000 ~ 20 000 进行实验. 在二维数值属性数据集上, 与 BORDER、BDDTS、BRIM 算法进行比较: BORDER 算法的时间复杂度为  $O(kn^2)$ , BDDTS 算法为  $O(n^2)$ , BRIM 的时间复杂度为  $O(n \log n)$ . 在分类属性数据集上与 CBORDER 算法进行效率比较, CBORDER 算法的时间复杂度为  $O(mn^{3/2})$ . 从图 3(a) 可以看出, 在二维数值属性数据集上, BERGE 算法的运行效率低于 BRIM 算法, 但是高于 BDDTS 算法、

BORDER 算法; 从图 3(b) 可以看出, 在分类属性数据集上, BERGE 算法的运行效率略低于 CBORDER 算法。同时, 图 3 也验证了上述时间复杂度分析的正确性。



(a) 数值属性数据集的算法运行效率对比



(b) 分类属性数据集的算法运行效率对比

图 3 算法的运行效率对比

### 2.3 参数讨论

BERGE 算法的参数为: 簇个数  $k$ 、边界比例  $w$ 、边界阈值  $\beta$ 。参数  $k$  为预测的聚类个数。参数  $w$  为边界点占有的比例大小, 用来确定标记多少候选边界点。当  $w$  的值增加时, 相应的候选边界集中的对象个数将会增加; 反之, 相应的候选边界集中的对象个数将会减少。参数  $\beta$  用于控制获取的边界点在候选边界集中所占的比例, 当  $\beta$  增加时, 获取的边界点减少; 反之, 获取的边界点增多。大量实验结果表明, 当  $0.2 \leq w \leq 0.8$ ,  $0.6 \leq \beta \leq 0.9$  时, 算法效果最好。

## 3 结 论

本文利用模糊聚类和证据积累的思想提出了一种聚类边界检测算法 BERGE。它不但可以有效地检测混合属性数据集中聚类的边界, 而且还可以用于数值属性数据集和分类属性数据集的聚类边界检测, 具有检测精度高、对噪声有一定的抑制作用等特点。

### 参考文献(References)

- [1] Tan P N, Michael Steinbach, Vipin Kumar. Introduction to data mining[M]. New Jersey: Pearson Education, 2007: 305-402.
- [2] Xia C, Hsu W, Lee M L, et al. BORDER: An efficient computation of boundary points[J]. IEEE Trans on

- Knowledge and Data Engineering, 2006, 18(3): 289-303.
- [3] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]. Int Conf on Knowledge Discovery and Data Mining. Portland: ACM, 1996: 226-231.
- [4] Qiu B Z, Yue F, Shen J Y. BRIM: An efficient boundary points detecting algorithm[C]. Advances in Knowledge Discovery and Data Mining. Berlin: Springer, 2007: 761-768.
- [5] Qiu B Z, Wang S. A boundary detection algorithm of clusters based on dual threshold segmentation[C]. The 7th Int Conf on Computational Intelligence and Security(CIS). Sanya: IEEE, 2011: 1246-1250.
- [6] 薛丽香, 邱保志. 基于变异系数的边界点检测算法[J]. 模式识别与人工智能, 2009, 22(5): 799-802. (Xue L X, Qiu B Z. Boundary points detection algorithm based on coefficient of variation[J]. Pattern Recognition and Artificial Intelligence, 2009, 22(5): 799-802.)
- [7] 邱保志, 王波. 分类数据的聚类边界检测技术[J]. 计算机应用, 2012, 32(6): 1654-1656. (Qiu B Z, Wang B. Cluster boundary detection technology for categorical data[J]. J of Computer Applications, 2012, 32(6): 1654-1656.)
- [8] 蒋盛益, 李庆华. 一种基于引力的聚类方法[J]. 计算机应用, 2005, 25(2): 286-288. (Jiang S Y, Li Q H. Gravity-based clustering approach[J]. J of Computer Applications, 2005, 25(2): 286-288.)
- [9] 王丹, 毛紫阳, 吴孟达. 融合 Shadowed Sets 聚类的离群点检测算法[J]. 计算机科学与探索, 2012, 6(11): 985-993. (Wang D, Mao Z Y, Wu M D. Outlier detection algorithm on Shadowed Sets clustering[J]. J of Frontiers of Computer Science and Technology, 2012, 6(11): 985-993.)
- [10] Desoer C A. Slowly varying system  $x = A(t)x$ [J]. IEEE Trans on Automatic Control, 1969, 14(6): 780-781.
- [11] 邱保志, 曹鹤玲. 一种高效的基于联合熵的边界点检测算法[J]. 控制与决策, 2011, 26(1): 71-74. (Qiu B Z, Cao H L. An efficient boundary points detecting algorithm based on joint entropy[J]. Control and Decision, 2011, 26(1): 71-74.)
- [12] 邱保志, 杨洋, 杜效伟. BRINK: 基于局部质变因子的聚类边界检测算法[J]. 郑州大学学报: 工学版, 2012, 33(3): 117-121. (Qiu B Z, Yang Y, Du X W. BRINK: An algorithm of boundary points of clusters detection based on local qualitative factors[J]. J of Zhengzhou University: Engineering Science, 2012, 33(3): 117-121.)

(责任编辑: 李君玲)