基于密度的聚类中心自动确定的混合属性数据聚类算法研究

陈晋音 [1] 何辉豪 [1]

摘 要 面对广泛存在的混合属性数据, 现有大部分混合属性聚类算法普遍存在聚类质量低、聚类算法参数依赖性大、聚类类别个数和聚类中心无法准确自动确定等问题, 针对这些问题本文提出了一种基于密度的聚类中心自动确定的混合属性数据聚类算法. 该算法通过分析混合属性数据特征, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 分析不同情况的特征选取相应的距离度量方式. 在计算数据集各个点的密度和距离分布图基础上, 深入分析获得规律: 高密度且与比它更高密度的数据点有较大距离的数据点最可能成为聚类中心, 通过线性回归模型和残差分析确定奇异点, 理论论证这些奇异点即为聚类中心, 从而实现了自动确定聚类中心. 采用粒子群算法 (Particle swarm optimization, PSO) 寻找最优 d_c 值, 通过参数 dc 能够计算得到任意数据对象的密度和到比它密度更高的点的最小距离, 根据聚类中心自动确定方法确定每个簇中心, 并将其他点按到最近邻的更高密度对象的最小距离划分到相应的簇中, 从而实现聚类. 最终将本文提出算法与其他现有的多种混合属性聚类算法在多个数据集上进行算法性能比较, 验证本文提出算法具有较高的聚类质量.

关键词 数据挖掘, 混合属性, 数据聚类, 密度, 混合距离度量

引用格式 陈晋音, 何辉豪. 基于密度的聚类中心自动确定的混合属性数据聚类算法研究. 自动化学报, 2015, **41**(10): 1798−1813

DOI 10.16383/j.aas.2015.c150062

Research on Density-based Clustering Algorithm for Mixed Data with Determine Cluster Centers Automatically

CHEN Jin-Yin[1] HE Hui-Hao[1]

Abstract For mixed data clustering, mostly current clustering algorithms have shortcomings such as low clustering efficiency, clustering parameter sensibility, clustering center number initialization and center determination difficulty. A density based cluster center self-determination mixed data clustering algorithm is proposed in this paper. Firstly, mixed data are divided into three types, including numeric dominant data, categorical dominant data and balanced data based on their data attributes analysis, and corresponding similarity metrics are designed for these three types of mixed data. Then, based on the density and distance relationship for each data object, an important conclusion is achieved that those data objects that have both higher density and larger distance than other data objects are more likely to be the cluster centers. So the linear regression model and residuals analysis are used to find those outliers that are fixed to be cluster centers automatically. The initialization value of dcis most crucial to clustering efficiency, so particle swarm optimization (PSO) algorithm is adopted to search the optimal d_c by calculating the distance and density of each data object according to the automatic method for determining the cluster centers. After the cluster centers have been found, each remaining point is assigned to the same cluster as its nearest neighbor of higher density. Finally, the performance of the proposed method is testified by a series of simulations on real-world datasets in comparison with other excellent clustering algorithms.

Key words Data mining, mixed attributes, data clustering, peak density, mixed distance measure methods

Citation Chen Jin-Yin, He Hui-Hao. Research on density-based clustering algorithm for mixed data with determine cluster centers automatically. *Acta Automatica Sinica*, 2015, **41**(10): 1798−1813

聚类是将物理或者抽象的对象集合中具有相似的对象聚集在同一个类中, 使同一个聚类形成的簇中的对象具有较高相似度, 不同簇中的对象相似度较低[1−3]. 聚类分析技术在图像处理、基因表达分析[4]、文本分析[5] 等诸多领域有着广泛的应用前景. 现实世界中产生的数据大多是同时具有取值为连续数值的数值属性和代表类别或状态的分类属性这两种属性类型[6−7]. 然而, 目前的聚类算法大多用于处理单重属性的数据, 比如 K-means[8]、Fuzzy K-means[9]、CURE[10]、DIEMA[11]、BRICH[12]、DBSCAN[13] 和基于证据推理的方法[14−15] 等, 其中

收稿日期 2015-02-03 录用日期 2015-07-14
Manuscript received February 3, 2015; accepted July 14, 2015
浙江省自然科学基金 (Y14F020092), 宁波市自然科学基金 (2013A610070) 资助
Supported by Natural Science Foundation of Zhejiang Province (Y14F020092), Natural Science Foundation of Ningbo City (2013A610070)
本文责任编委 杨健
Recommended by Associate Editor YANG Jian
1. 浙江工业大学信息工程学院 杭州 310023
1. Institute of Information Engineering, Zhejiang University of Technology, Hangzhou 310023

基于证据推理的方法, 针对模糊聚类时存在的不足, 基于证据理论有效地处理不确定性数据, 具有较高的聚类性能, 这些方法针对处理数值属性数据, 另外, K-modes[16]、Fuzzy K-modes[17]、COOLCAT[18] 等针对处理分类属性数据. 在处理混合属性数据时, 上述的算法不能得到期望的聚类结果.

到目前为止, 也有一些研究工作直接处理混合类型数据. Huang 结合 K-means 和 K-modes 算法的思想提出了 K-prototypes 算法[19] 来解决这个问题. 考虑到数据对象在簇归属上的不确定性, Chatzis 等提出了 KL-FCM-GM[20] 算法来扩展 K-prototypes 算法, KL-FCM-GM 算法是 Gath-Geva 算法[21] 的扩展, 是为高斯多项分布数据设计的, 该算法假设簇中的对象符合高斯多项分布. Zheng 等引入了进化算法框架, 提出了 EKP[22] 算法, 该算法具有全局搜索能力. Li 等提出了基于相似度的凝聚层次聚类算法 SBAC 算法[23], 该算法采用 Goodall[24] 提出的相似度量方法来测量数据对象间的相似性. Hsu 等提出了基于方差和熵的 CAVE 算法[25], 该算法首先需要对分类属性建立距离等级制度, 该制度的建立需要先验知识. Ahmad 等提出了一个 K-means 类型的算法[26] 来处理混合属性数据, 这个算法利用属性值的共现性计算分类属性值之间的距离. Ji 等提出了 IWKM[27] 和 WFK-prototypes[28] 算法, 考虑数据对象在簇归属上的不确定性的同时, 采用 Ahmad 等[26] 提出的属性重要性概念, 一定程度上提高了聚类精度. Hsu 等结合适应性共鸣理论网络和概念距离层次的思想提出了一个增量聚类算法[6].

Rodriguez 等在 *Science* 期刊上提出了一种基于中心点具有高密度 ρ 且与比它高密度点具有较大距离 δ 的假设的算法[29]. 该算法存在以下问题: 1) 算法通过数据对象 ρ 和 δ 分布图, 需要人为监督确定相应的 ρ 和 δ 值的大小, 然后确定中心点; 2) ρ 和 δ 值的分布依赖于截断距离参数 d_c, 算法聚类的质量过度依赖参数 d_c 的选择; 3) 该算法不能有效处理混合属性数据.

上述处理混合属性数据的方法大多是基于划分、层次方法上的扩展. 基于划分的方法仍存在需要确定聚类个数、对簇中心的选取敏感、不能发现任意形状的簇以及对异常点比较敏感等缺点. 同样, 基于层次的方法存在需要存储相似度矩阵, 具有较高时间和空间复杂度的缺点.

针对上述问题, 本文提出了一种基于密度的聚类中心自动确定的混合属性数据聚类算法 (Density-based clustering algorithm for mixed data with determine cluster centers automatically, DC-MDACC), 该算法通过对混合数据进行占优分析, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 针对不同情况, 选择相应的距离计算方法. 算法需要给定参数 d_c 的范围, 通过粒子群优化算法 (Particle swarm optimization, PSO) 算法寻找最优 d_c 值, 对于给定的参数 d_c, 算法可以计算得到每个数据对象的密度和到比它密度更高的点的最小距离. 根据密度和距离的分布图, 我们将高密度且与比它更高密度的数据点有较大距离的数据点作为聚类中心, 通过回归分析自动确定中心点, 并将其他点按到最近邻的更高密度对象的最小距离划分到相应的簇中, 获得最终的聚类结果.

1　混合属性距离计算方式

距离的度量是进行有意义的聚类分析的前提, 表 1 列出了基于划分聚类的经典算法的距离计算方式.

K-means 算法采用广泛使用的欧氏距离来处纯数值属性数据, K-modes 算法采用简单匹配距离处理纯分类型数据. K-prototypes 算法结合 K-means 算法和 K-modes 算法处理混合属性数据, 算法 EKP、WFK-prototypes 及一些其他算法在原距离公式中加入模糊因子、权重系数等来改进 K-prototypes 算法, 使其能更准确度量对象间相似性.

对于任意混合属性数据集, 上述算法均采用一

表 1　5 种算法的距离计算方式

Table 1　Five distance measures of partition-based clustering algorithms

算法	距离计算方式	数值型距离计算方法	分类型距离计算方法	类型
K-means[9]	$d\left(X_i,X_j\right)=\sqrt{\sum_{p=1}^{m}\left(X_i^p-X_j^p\right)^2}$	$d\left(X_i^p,X_j^p\right)=\left(X_i^p-X_j^p\right)^2$	None	数值型
K-modes[16]	$d\left(X_i,X_j\right)=\sum_{p=1}^{m}\delta\left(X_i^p,X_j^p\right)$	None	$\delta(X_i^P-X_j^P)=\begin{cases}0, & X_i^P=X_j^P\\ 1, & X_i^P\neq X_j^P\end{cases}$	分类型
K-prototypes[19]	$d\left(X_i,Q_l\right)=\sum_{j=1}^{p}\left(X_{ij}^r-q_{ij}^r\right)^2+\mu_l\sum_{j=p+1}^{m}\delta\left(X_{ij}^c,q_{ij}^c\right)$	$d\left(X_i,Q_l\right)=\left(X_i^p-Q_j^p\right)^2$	$\mathrm{d}(X_i^p,Q_j^p)=\begin{cases}0, & X_i^p=Q_j^p\\ 1, & X_i^p\neq Q_j^p\end{cases}$	混合型
EKP[22]	$d\left(X_i,Q_l\right)=\sum_{j=1}^{p}\left(X_{ij}^r-q_{ij}^r\right)^2+r\sum_{j=p+1}^{m}\delta\left(X_{ij}^c,q_{ij}^c\right)$	$d\left(X_i,Q_l\right)=\left(X_i^p-Q_j^p\right)^2$	$\mathrm{d}(X_i^p,Q_j^p)=\begin{cases}0, & X_i^p=Q_j^p\\ 1, & X_i^p\neq Q_j^p\end{cases}$	混合型
WFK-prototypes[28]	$d\left(X_i,Q_l\right)=\sum_{l=1}^{p}\left(s_l\left(X_{ij}^r-q_{ij}^r\right)^2\right)+\sum_{l=p+1}^{m}\varphi\left(X_{ij}^c,v_{ij}^c\right)^2$	$d\left(X_i,Q_l\right)=s_l\left(X_i^p-Q_j^p\right)^2$	$\varphi\left(X_i^p,Q_j^p\right)^2$	混合型

致的距离度量方式, 实际应用中由于数据集中混合属性维度对最终聚类存在不一样的重要性.

例如, Zoo 动物数据集中每个对象均有特征: 腿 (数值型)、头发 (分类型)、羽毛 (分类型) 等共 1 维数值属性, 16 维分类属性. 例如 3 个动物信息如表 2 所示.

表 2 动物数据集中样本对象

Table 2 Sample object of zoo data set

对象	分类属性	数值属性	类标
deer	1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1	4	1
dolphin	0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1	0	1
frog	0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0	4	5

用表 1 中算法 K-prototypes、EKP 进行聚类, 以对象 deer 为例, 分别计算距离 d(deer, dolphin), d(deer, frog), 如表 3 所示.

表 3 动物数据集中对象间距离

Table 3 The distance between objects of zoo data set

对象间距离	分类距离	数值距离	总距离
d(deer, dolphin)	4	16	20
d(deer, frog)	8	0	8

如表 3 所示, 对象 deer 与 frog 的距离相对 deer 与 dolphin 较小, 因此对象 deer 与 dolphin 相比更偏向与对象 frog 聚成一类. 而实际上, 对象 deer 与 dolphin 均属于类标为 1 的类, 而与对象 frog 属于不同的类.

使用算法 K-prototypes、EKP 中距离计算方法时, 存在不同类中对象不可分的情况.

再例如: 网络入侵数据集中每条记录均有特征: 协议类型 (分类型)、连接时间 (数值型)、数据字节数 (数值型) 等共 34 维数值, 7 维分类属性. 在计算对象间距离时, 使得少数数值属性或分类属性对整体的距离产生较大影响, 而影响最终的聚类质量.

因此本文针对此问题, 引入了占优因子, 提出了占优分析方法, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 不同类型选择不同的距离计算公式.

1.1 占优分析

假设待处理数据为数据集 $D = (X_1, X_2, \cdots, X_i, \cdots, X_n)$, 每一个样本具有 d 维属性 $X_i = (X_i^1, X_i^2, \cdots, X_i^d)$, 其中有 r 维数值属性与 q 维分类属性, $d = r + q$. 引入占优因子 α, 将 r 与 d 的比值和 q 与 d 的比值作为占优分析的评判标准.

1) 若 $r/d > \alpha$, 则数据集 D 是数值占优数据集.

2) 若 $q/d > \alpha$, 则数据集 D 是分类占优数据集.

3) 若 $1 - \alpha < r/d > \alpha$ 或 $1 - \alpha < q/d > \alpha$, 则数据集 D 是均衡型混合属性数据.

UCI 数据及其学习库中有 56 个混合属性数据集, 通过对 UCI 中多个数据进行测试, 得到通用占优因子 α 为 0.75, 即:

1) 若 $r/d \in [0.75,1]$, 则数据集 D 是数值占优数据集.

2) 若 $q/d \in [0.75,1]$, 则数据集 D 是分类占优数据集.

3) 若 $r/d \in (0.25, 0.75)$ 或 $q/d \in (0.25,0.75)$, 则数据集 D 是均衡型混合属性数据.

1.2 数据对象间距离计算方式

考虑到混合属性数据包括数值属性和分类属性, 对混合属性数据进行占优分析, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 针对不同情况, 选择相应的距离计算方法, 数值占优和分类占优的距离计算方式不同是为了降低非占优属性对数据对象整体相似性的影响, 而均衡型混合属性数据需要综合考虑每一维属性的重要性.

对于一些特殊的情况不能用以上三种占优分析解决的, 例如: 数据集属于数值占优型数据 (分类占优数据), 虽然分类属性 (数值属性) 维度很少, 但却对聚类结果起着决定性作用, 这样的特殊情况, 本文算法采用半监督聚类方法, 在预处理阶段从数据集中随机提取部分已知类标的数据进行训练, 从而获取数据对象的各维属性对聚类结果的影响权重, 具体操作如下:

假设待处理数据为数据集 $D = (X_1, X_2, \cdots, X_i, \cdots, X_n)$, 每一个样本具有 d 维属性 $X_i = (X_i^1, X_i^2, \cdots, X_i^d)$, 其中有 r 维数值属性与 q 维分类属性. 设置权重向量 $\omega = (\omega_r^1, \omega_r^2, \cdots, \omega_r^r, \omega_q^1, \omega_q^2, \cdots, \omega_q^q)$ 来描述各维属性的重要程度, 设置其初始值为 [0, 1] 的一个随机数. 通过粒子群算法 (PSO) 进化学习, 根据分簇的簇内对象到聚类中心的平均距离得到一个聚类质量评价, 不断更新权重向量 $\omega = (\omega_r^1, \omega_r^2, \cdots, \omega_r^r, \omega_q^1, \omega_q^2, \cdots, \omega_q^q)$, 直至聚类质量评价不再变化时, 获得最优的一组权重向量. 而对于一般数据集, 大部分均能利用占优分析准确分类其各维属性对聚类效果的影响, 因此 $\omega = (1, 1, \cdots, 1)$.

1.2.1 数值占优和分类占优

本文分别用 $d(X_i, X_j)_n$ 和 $d(X_i, X_j)_c$ 代表数值属性部分的距离和分类属性部分的距离, 距离定义为 (以数据集 D 为例):

1) 若数据集 D 是数值属性占优的数据, 则对于

数据集中数据对象之间距离定义如下:

定义 1. 任意两个对象 X_i, X_j 的数值属性部分的距离为

$$d(X_i, X_j)_n = \sqrt{\sum_{p=1}^{m} \left(X_i^p - X_j^p\right)^2} \tag{1}$$

定义 2. 任意两个对象 X_i, X_j 的分类属性部分每一维的距离采用二元化的方法, 如 X_i, X_j 的第 p 维之间的距离为

$$d(X_i^p, X_j^p) = \begin{cases} 0, & X_i^p = X_j^p \\ 1, & X_i^p \neq X_j^p \end{cases} \tag{2}$$

则分类属性部分的距离为

$$d(X_i, X_j)_c = \sum_{p=1}^{q} d(X_i^p, X_j^p) \tag{3}$$

2) 若数据集 D 是分类属性占优的数据, 则对任意数据对象 X_i 的数值属性部分的每一维均进行标准化处理, 即 X_i 的第 p 维的值为

$$d(X_i^p)_n = \frac{X_i^p - X_{i\_\min}^p}{X_{i\_\max}^p - X_{i\_\min}^p} \tag{4}$$

其中, $X_{i\_\max}^p$ 为该维样本数据的最大值, $X_{i\_\min}^p$ 为该维样本数据的最小值. 则对于数据集 D 中数据对象间数值部分距离定义如下:

定义 3. 任意两个对象 X_i, X_j 的数值属性部分的距离为

$$d(X_i, X_j)_n = \sum_{p=1}^{r} \left(d(X_i^p)_n - d\left(X_j^p\right)_n\right) \tag{5}$$

对于分类占优数据, 其分类部分的距离计算与定义 2 一致.

数值占优或分类占优数据集 D 中数据对象任意两个对象间距离定义如下:

定义 4. 数值占优或分类占优数据集 D 中数据对象任意两个对象 X_i, X_j, 则 X_i 与 X_j 的距离为

$$D\left(X_i, X_j\right) = d(X_i, X_j)_n + d(X_i, X_j)_c \tag{6}$$

对于数值占优和分类占优数据的处理, 主要是基于突出占优属性对数据整体相似性的重要性, 降低非占优属性对数据整体相似性的影响.

存在样本数据集 DataSet1, 每个对象包含 1 维数值属性和 1 维分类属性, 如 X_1 (1, 10), X_2 (2, 40), X_3 (3, 70) 等共 30 个点, 其二维空间内数据分布如图 1 所示.

若按数值属性占优数据处理, 则其分布图与图 1 一致, 但计算距离时, 由于分类部分距离远小于数值部分距离, 使得数值属性占优数据中分类属性数据对整体距离的计算影响较小.

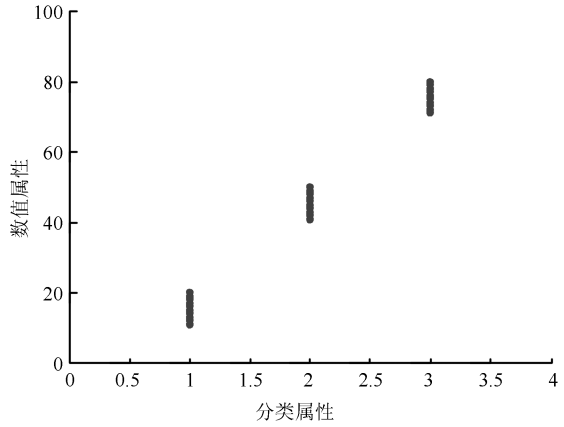


图 1　DataSet1 数据二维分布图

Fig. 1　The 2-dimensional distribution of DataSet1

若按分类属性占优数据处理, 则其分布图如图 2 所示. 数值部分经过了归一化处理, 但仍保留数据原始的分布结构.

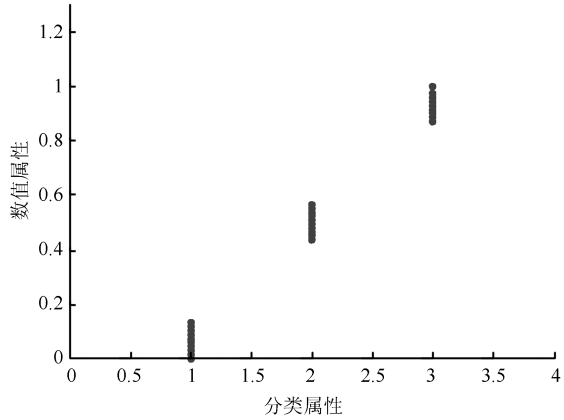


图 2　数值归一化后样本 DataSet1 数据分布图

Fig. 2　The 2-dimensional distribution of DataSet1 after normalization process of numerical attributes

计算距离时, 由于数值部分被映射到 [0, 1] 区间内, 使得计算数值部分距离时, 远小于大多数分类属性产生的分类部分距离, 使得分类属性占优数据中数值属性对整体距离的影响减小.

1.2.2　均衡型混合属性数据

传统的距离度量会独立看待每个属性, 在计算差异性的时候, 处理每个属性都只是简单地比较同一个属性的取值关系, 可是实际上, 对于一个样本集, 每个属性都不是孤立的, 它同其他属性取值之间存在着某种关联, 而这种关联也体现出了样本集所蕴含的内在类属结构[26].

假定数据集 D, A_i 表示一个分类属性, 假设 x 和 y 是这个属性的两个不同的属性值. 用 A_j 表示另一个分类属性, z 表示值域 $\mathrm{Dom}(A_j)$ 的子集, z^c 表示集合 z 的补集. $p_1(z|x)$ 表示属性 i 的值为 x 的数据对象在属性 j 上的值属于集合 z 的条件概率, $p_i(z^c|y)$ 表示属性 i 的值为 y 的数据对象在属性 j 上的值属于集合 z_c 的条件概率.

定义 5. 相对分类属性 A_j, 属性 i 的两个值 x 和 y 之间的最大距离 $\max\ d^{ij}(x,y)$ 就可以由以下公式衡量:

$$\max\ d^{ij}(x,y) = P_i\left(\frac{z}{x}\right) + P_i\left(\frac{z^c}{y}\right) \tag{7}$$

其中, z 为 A_j 取值的子集, 这个子集能最大化式 (7) 的值. 当集合 z 为空集或为 A_j 取值的集合本身时, 式 (7) 的取值为 1, 即说明最大化式 (7) 的取值大于等于 1. 注意到 $p_1(z/x)$ 和 $p_i(z^c/y)$ 的取值都在 [0, 1] 范围内, 因此式 (7) 的取值范围为 [1, 2], 进一步修正 $\max\ d^{ij}(x,y)$ 的计算为:

$$\max\ d^{ij}(x,y) = P_i(z/x) + P_i(z^c/y) - 1.0 \tag{8}$$

使得 $\max\ d^{ij}(x,y)$ 的取值在 [0, 1] 范围内.

式 (8) 把属性 i 的两个值 x 和 y 之间的距离表示为这两个值和另一个属性 j 的属性值集的共现概率. 当出现多个分类属性时, 属性值 x 和 y 相对于这些属性的距离可以用类似的方法计算得到. 当出现数值属性时, 通过离散化数值属性, 属性值 x 和 y 相对于数值属性的距离可以用类似的方法计算得到.

定义 6. 对于混合属性数据集 D, 每一个样本 d 维属性, 其中有 q 维分类属性, r 维离散化的数值属性, 任意分类属性 A_i 的取值 x 和 y 之间的距离为

$$d^i(x,y) = \frac{\sum\limits_{j=1,i\neq j}^{d} d^{ij}(x,y)}{d-1} \tag{9}$$

其中, $d^i(x,y)$ 具有下述的三个属性:

1) $0 \le d^i(x,y) \le 1$;

2) $d^i(x,y) = d^i(y,x)$;

3) $d^i(x,x) = 0$.

要计算数值属性同一维中不同取值间的距离, 数值属性通常需要离散化, 因此首先对数值属性进行了离散化, 并对所有的数值属性设定相同的离散间隔 T, 每个间隔指定一个分类属性 $u[1]$, $u[2]$, $\cdots$, $u[T]$. 对离散化的数值属性, 利用式 (9) 计算每一对分类属性值的距离, 计算的方法与计算分类属性值的方法相同.

定义 7. 均衡型混合属性数据集 D 中任意两个对象 X_i, X_j 之间的距离为

$$D\left(X_i, X_j\right) = \sum_{p=1}^{d} d^p\left(X_i, X_j\right) \tag{10}$$

2 聚类中心自动确定方法 (ACC)

2.1 ACC 主要思想

聚类中心自动确定方法 (Automatically determining the cluster centers, ACC) 基于以下思想:

1) 簇类中心被具有较低局部密度的邻居点包围, 且与具有更高局部密度的其他数据对象有相对较大的距离.

2) 噪声点具有较大的距离 δ 和相对较小的局部密度 ρ.

对于任意一个数据对象 i, 需要计算两个量: 数据对象的局部密度 ρ_i 和到具有更高局部密度的其他点的最小距离 δ_i. 局部密度和最小距离的计算依赖于预设的截断距离参数 d_c.

定义 8. 对于任意数据对象 i, 其局部密度 ρ_i 的计算方式如下:

$$\rho_i = \sum_j f\left(d_{ij} - d_c\right) \tag{11}$$

$$f(x) = \begin{cases} 1, & x = d_{ij} - d_c < 0 \\ 0, & \text{否则} \end{cases} \tag{12}$$

局部密度等价于数据对象 i 的 d_c 领域内的数据对象个数.

定义 9. 对于任意数据对象 i, 其到具有更高局部密度的其他数据对象的最小距离 δ_i 定义如下:

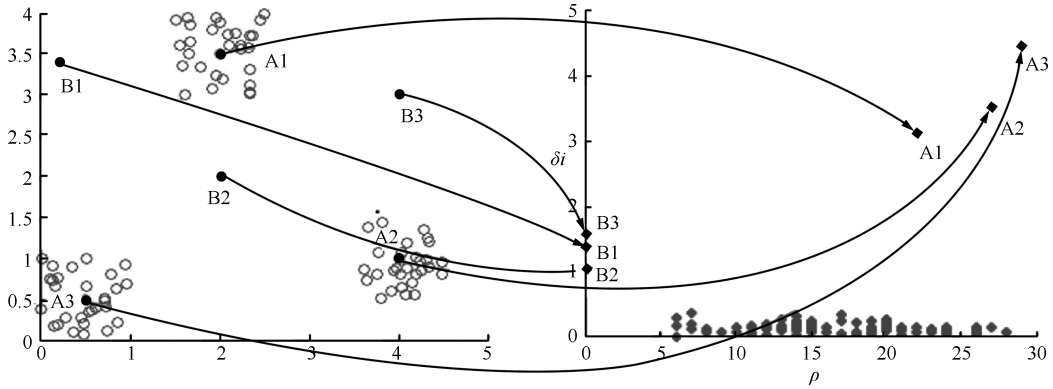
$$\delta_i = \min_{j:\rho_j > \rho_i}\left(d_{ij}\right) \tag{13}$$

其中, 对于最优最高局部密度的数据点, 定义 $\delta_i = \max_j\left(d_{ij}\right)$.

存在样本数据集 DataSet2, 其二维空间内数据分布如图 3 (a) 所示. 计算样本数据集中每个数据对象 i 的局部密度 ρ_i 和到具有更高局部密度的其他点的最小距离 δ_i, 作出 ρ 和 δ 的分布图如图 3 (b) 所示.

数据集数据分布与数据对象 ρ 和 δ 分布存在如下映射关系:

图 3 (a) 中 3 个点 A1、A2、A3 是原始数据分布中的三个簇的簇类中心, 其在图 3 (b) 中分布具有较大的密度 ρ 和较大的距离 δ. 图 3 (a) 中三个点 B1、B2、B3 是远离簇的数据点, 即离群点, 其在图 3 (b) 中分布具有较大的距离 δ 和较小的密度 ρ. 而其他点称为边界点, 均属于某个簇类, 具有较小距离 δ 的性质.



(a) Dataset 2 数据分布图　　(b) 数据对象 ρ 和 δ 分布图

(a) Point distribution of Dataset 2　　(b) ρ and δ distribution of data objects

图 3　样本数据分布与 ρ 和 δ 的分布图映射关系图

Fig. 3　Mapping relationship between point distribution and ρ and δ distribution

根据上述映射关系, 算法采用非线性函数 $y = b_0 + b_1/x$ 转换为线性函数去拟合, 令 $x' = 1/x$, 则 $y = b_0 + b_1 \times x'$, 利用线性函数模型拟合所有数据局部密度 ρ_i 和距离 δ_i 的函数关系 $\delta_i^{\times} = f(\rho_i)$. 使用残差分析确定 ρ_i 和 δ_i 的分布图中奇异点信息, 其中奇异点为远离拟合曲线的点, 即是聚类的簇中心, 奇异点个数是聚类的簇个数.

ACC 算法整体框架如图 4 所示.

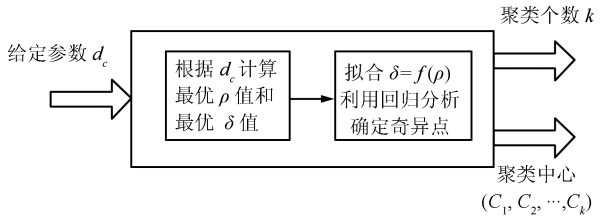


图 4　ACC 算法整体框架

Fig. 4　The framework of ACC algorithm

2.2　回归分析确定聚类中心

回归分析是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法. 线性回归模型建立基于以下前提假设:

1) 随机误差项是一个期望值或平均值为 0 的随机变量;

2) 对于解释变量的所有观测值, 随机误差项有相同的方差;

3) 随机误差项彼此不相关;

4) 随机误差项服从正态分布.

线性回归模型的前提假设符合高斯 – 马尔科夫定理, 即求得的线性回归模型回归系数的最佳线性无偏估计就是最小方差估计.

高斯 – 马尔科夫定理: 在误差零均值、同方差且互不相关的线性回归模型中, 回归系数的最佳线性无偏估计 (BLUE) 就是最小方差估计.

推论 1. 令 $\rho' = 1/\rho$, 线性模型 $\delta^* = b_0 + b_1 \times \rho'$ 的残差 $\varepsilon_i = \delta_i^* - \delta_i$ 服从 $\mathrm{N}(0, \sigma^2)$ 正态分布.

推论 2. 标准化残差 $ZRE_i = \varepsilon_i/\sigma$ 服从 $\mathrm{N}(0, 1)$ 标准正态分布.

定理 1. 对于任意残差 ε_i, 均有一个置信度为 $1-\alpha$ 的置信区间 $[\varepsilon_i - \sigma \times Z_{\alpha/2}, \varepsilon_i + \sigma \times Z_{\alpha/2}]$, 若残差 ε_i 在置信区间外, 则对应的数据对象为奇异点, 即为算法期望聚类中心.

证明. 设残差为 ε_i, 其服从 $\mathrm{N}(0, \varepsilon_i)$ 正态分布. 令

$$P = \left\{ \left| \frac{\bar{X} - \mu}{\sqrt{\frac{\sigma^2}{n}}} \right| \le Z_{\frac{\alpha}{2}} \right\} = 1 - \alpha$$

$$P\left\{ -Z_{\frac{\alpha}{2}} \le \frac{\bar{X} - \mu}{\sqrt{\frac{\sigma^2}{n}}} \le Z_{\frac{\alpha}{2}} \right\} = 1 - \alpha$$

$$P\left\{ -\frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \le \bar{X} - \mu \le \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \right\} = 1 - \alpha$$

$$P\left\{ \bar{X} - \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \le \mu \le \bar{X} + \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \right\} = 1 - \alpha$$

对于任意一个普通残差 ε_i, 则 $\bar{X} = \varepsilon_i$ 且 $n = 1$, 得到:

$$P\left\{ \varepsilon_i - \sigma \times Z_{\frac{\alpha}{2}} \le \mu \le \varepsilon_i + \sigma \times Z_{\frac{\alpha}{2}} \right\} = 1 - \alpha$$

则对于任意一个残差 ε_i, 认为其落在区间 $[\varepsilon_i - \sigma \times Z_{\alpha/2}, \varepsilon_i + \sigma \times Z_{\alpha/2}]$ 内的可信度为 $(1-\alpha) \times 100$.

若残差不在置信度为 $1-\alpha$ 的置信区间内, 则认为对应的对象点为奇异点, 即为算法期望的簇类中心.

α 的设置影响置信区间的范围大小, α 值越大, 则置信度越小, 置信区间越小; 反之, 则置信区间越大. 由于聚类中心的密度和到更高密度点之间的最小距离相比其他数据点均较大, 参考一般置信区间的精度要求和检验要求[30], 本文置信区间的参数置信因子 α 设置为 0.05 即可满足检验要求, 若聚类中心与其簇中数据点的差异较小, 则可以适当增大 α 的取值来获得准确的聚类中心.

2.2.1 案例验证

以图 2 样本数据为例, 使用非线性函数 $y = b_0 + b_1/x$ 转换为线性 $y = b_0 + b_1 \times x'$, 利用线性函数模型拟合所有数据局部密度 ρ_i 和距离 δ_i 的函数关系 $\delta_i^* = f(\rho_i)$, 得到拟合曲线如图 5 所示.

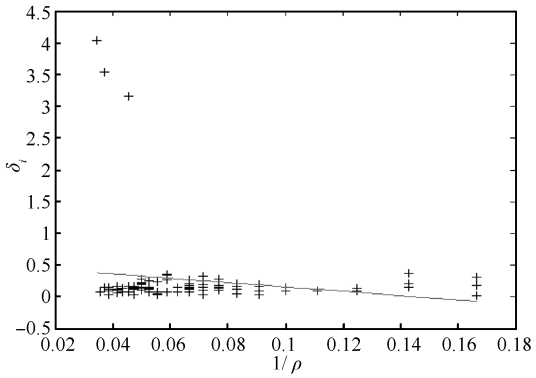


图 5 $\delta_i^* = f(\rho_i)$ 函数关系拟合曲线

Fig. 5 $\delta_i^* = f(\rho_i)$ fitting curve

对 $y = b_0 + b_1 \times x'$ 函数关系拟合曲线进行残差分析, 设定 $\alpha = 0.05$, 得到结果如图 6 所示. 大多数数据点如图 6 中所示, 随机分布在零值左右, 说明利用该线性函数模型 $y = b_0 + b_1 \times x'$ 拟合是合理的. 图中有三个点不在置信区间内, 就是拟合产生的奇异点, 即为算法期望的簇类中心.

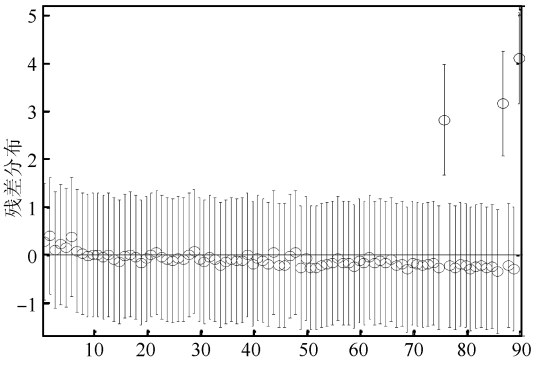


图 6 残差分布图

Fig. 6 Residual distribution

2.3 ACC 算法流程与步骤描述

ACC 算法流程如图 7 所示, 其步骤为:

步骤 1. 根据混合属性占有分析结果对数据 D 确定相应的距离计算方式, 以式 (11) 和式 (13) 计算每个数据对象 i 的 ρ_i 和 δ_i.

步骤 2. 得到 ρ_i 和 δ_i 的函数关系 $\delta_i^* = f(\rho_i)$, 根据回归分析中逆函数 $y = b_0 + b_1/x$ 来拟合此函数关系, 将其转化为线性模型 $y = b_0 + b_1 \times x'$, 则可以利用线性回归模型得到拟合 $\delta_i^* = f(\rho_i)$ 曲线.

步骤 3. 采用残差分析计算拟合函数的各残差分布特征, 并求得 k 个奇异点集合 $(C_1, C_2, \cdots, C_k)$.

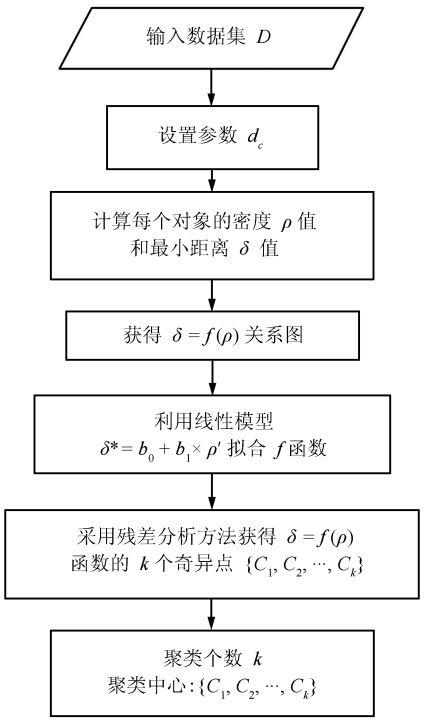


图 7 ACC 算法流程图

Fig. 7 The flowchart of ACC algorithm

3 DC-MDACC 算法

3.1 DC-MDACC 算法主要思想

根据上文证明, 在给定 d_c 的前提下, 通过本文提出的聚类中心自动确定方法可以计算得到最终的聚类类别数和聚类中心, 如何给定最优 d_c 尤为重要. DC-MDACC 算法的思路是首先计算 d_c 的取值区间, 根据 *Science* 上 Alex 算法[27] 证明的将 d_c 取值限定为使得数据对象平均局部密度为数据集数量的 1 %～2 % 时, 可以取得较好的聚类结果. 由于 DC-MDACC 算法面向混合属性数据, 因此对多个来自 UCI 数据库的混合属性数据集观测得到, 使数据对象的平均密度 average(ρ) 为数据集数量的 1 %～20 % 时, 在此范围内可以找到最优 d_c 解. 因此根据每个数据集的数量, 给定 d_c 取值区间

$[d_{c\_low}, d_{c\_high}]$, 随机选取一个 d_c, 然后利用 ACC 算法确定该 d_c 取值情况下的聚类中心, 并利用密度的方法将数据集中所有点划分到各个簇中, 根据分簇的簇内对象到中心的平均距离得到一个聚类质量评价, 通过反复选取 d_c, 从而可以得到较优的聚类质量, 整个反复寻求最优的 d_c 过程利用粒子群优化算法实现, 转化成一个最优化问题. 其中, d_c 作为 PSO 的解, 聚类质量作为评价每个 d_c 优劣的适应度函数. 最终找到最优 d_c 值, 并确定相应的聚类中心.

当聚类中心选定后, 一般基于划分聚类算法将其他点按到中心点的最小距离进行划分, 使得划分聚类算法对球形簇能够产生较好的聚类效果, 不能有效地对任意形状的簇聚类. 本文算法中将其他点按到最近邻的更高密度对象的最小距离进行划分, 具体规则如下:

当前对象的类别标签与高于当前对象局部密度的最近邻对象的标签一致, 从而对所有对象的类别进行标定. 如图 8 所示, 编号表示密度高低, 数字越高表示密度越大. 其中 "4" 号为聚类中心, 类标为 1, "3" 号点的类别标签应该与距离其最近的密度高于它的对象一致, 因此 "3" 号点类标为 1, 由于 "1" 号点最近的密度比其高的对象为 "3" 号点, 因此其类别标签与 "3" 号对象相同, 类标也为 1.

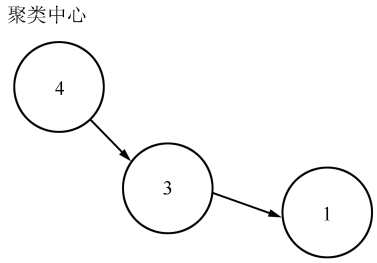


图 8　划分规则图

Fig. 8　Partition rule

该过程首先需要对数据集中所有数据点按密度从大到小排序, 其计算复杂度为: $\mathrm{O}(n \times \log(n))$, 其中 n 为数据对象的数目. 然后除确定的中心点外, 从高密度数据点开始将其按最近邻更高密度对象的最小距离进行类别标定, 从而完成聚类, 该过程的计算复杂度为 $\mathrm{O}(n/2)$.

对于噪声点, 算法无需人为设定噪声点阈值截断的方法去除噪声点, 而是先算出类别之间的边界, 然后找出边界中密度值最高的点的密度作为阈值, 将此密度阈值记为 ρ_b, 只保留此类别中大于或等于此密度值的点.

3.2　DC-MDACC 算法流程和步骤描述

DC-MDACC 算法流程图如图 9 所示, 其步骤为:

步骤 1. 确定截断距离参数 d_c 的取值范围 $[d_{c\_low}, d_{c\_high}]$. 设定粒子群算法的粒子数 m、最大迭代次数 Maxiter. 利用随机函数 rand(·) 在范围 $[d_{c\_low}, d_{c\_high}]$ 内随机生成 m 个粒子, 并初始化当前进化代数 $\mathrm{iter} = 0$.

步骤 2. 根据当前的粒子值, 利用 ACC 算法求得相应的聚类中心集合 (C_1, C_2, $\cdots$, C_n), 并进行准确划分.

步骤 3. 根据适应度计算公式 (14), 求得每个粒子的对应适应度函数值.

$$Fitness = \frac{\sum\limits_{j=1}^{k}\left[\sum_{x_i \in C_j} \frac{d(x_i, C_j)}{|C_j|}\right]}{k} \tag{14}$$

其中, k 表示簇的个数, C_j 表示簇中心, $|C_j|$ 表示该簇的数据对象个数.

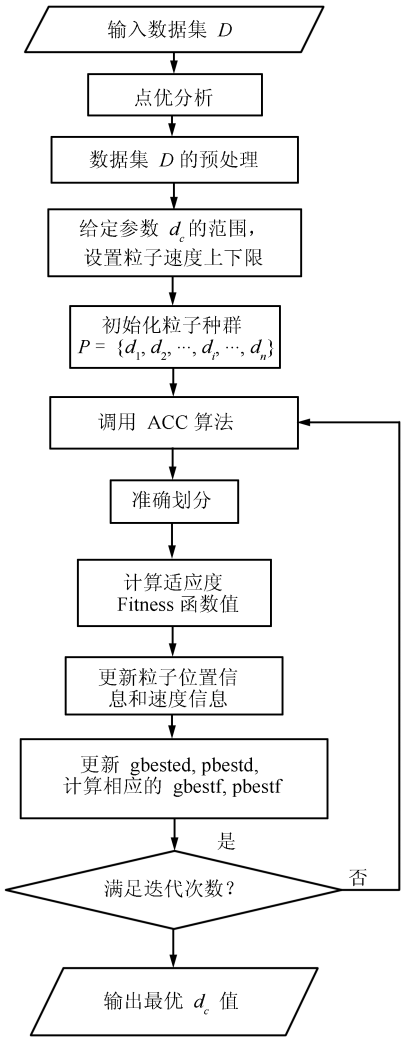


图 9　DC-MDACC 算法流程图

Fig. 9　The flowchart of DC-MDACC

步骤 4. 设置当前适应值为个体极值 pbestf, 当前的位置为个体极值位置 pbestd, 根据各个粒子的个体极值找出全局极值 gbestf 和聚集极值位置 pbestd. 进化代数 iter = iter+1.

步骤 5. 当进化代数 iter $\leq$ Maxiter, 根据式 (15) 和 (16) 更新粒子位置和速度, 然后转向步骤 2; 否则, 转向步骤 7.

$$v_i(t+1) = w \times v_i(t) + c1 \times r1 \times (\text{pbestd} - d_i(t)) + c2 \times r2 \times (\text{gbestd} - d_i(t)) \tag{15}$$

$$d_i(t+1) = d_i(t) + v_i(t+1) \tag{16}$$

步骤 6. 输出全局极值 gbestf 和全局极值位置 gbestd, 记录此时的 gbestd 就是当前最优的 d_c, 对应的聚类中心及聚类分析.

4 仿真实验与性能分析

实验中的操作系统为 Windows 7, 集成开发环境为 Microsoft Visual C++2010. 硬件条件: CPU 为 Intel Core I5 2.6 GHz, 内存为 4 GB.

为了验证新算法 DC-MDACC 的性能, 我们测试了 10 个数据集, 这 10 个数据集均来自 UCI 及其学习库 (Machine learning repository), 具体信息如表 4 所示.

4.1 聚类结果评价

1) 本文采用由 Gan 等[17] 提出的聚类准确率作为评价标准, 聚类准确率 r 的定义如下:

$$r = \frac{\sum_{i=1}^{k} a_i}{n} \tag{17}$$

其中, a_i 表示最终被正确分类的样本数目, k 表示聚类数, n 表示数据集中的样本个数. 聚类准确率越高, 算法的聚类效果越好. 当 r 的值为 1 时, 此时算法在数据集上的聚类结果是完全正确的.

2) 平均聚类纯度 Purity:

$$Pur = \sum_{i=1}^{k} \frac{\frac{|C_i^d|}{|C_i|}}{K} \tag{18}$$

其中, K 表示为簇的个数, $|C_i^d|$ 表示在簇 i 中具有该簇最主要类标号的数据对象数, $|C_i|$ 表示簇 i 中包含的所有数据对象的个数. 平均聚类的范围为 [0, 1], 纯度值越高, 算法的聚类效果越好.

聚类准确率反映数据整体的聚类效果, 聚类纯度能够反映簇内的聚类质量, 两者相互补充.

4.2 实验结果分析

实验中参数除特殊说明外, 粒子群算法 PSO 中学习因子设定为 $c_1 = c_2 = 1.8$, 惯性权重 $w = 0.9$, 粒子数 $n = 5$, 最大迭代次数 Maxiter = 10, 并设定置信区间 $\alpha = 0.05$.

数据集 Aggregation、数据集 Jain、数据集 Spiral、数据集 Flame 均为二维数值型数据, 其中包含各种形状的簇. 算法对这 4 个数据集进行测试, 其结果展示如图 10 所示.

实验结果显示, 算法能够对任意形状、变密度的簇进行聚类, 具有较好的聚类质量.

4.2.1 实验数据集

本文使用表 4 所示的数据集作为实验对象, 包含: Iris、KDD-CUP 99 网络入侵数据集、Soybean、Zoo、Acute Inflammatins 和 Statlog Heart, 其中 Iris 和 KDD-CUP 99 数据集属于数值占优型数据, Soybean、Zoo 和 Acute 数据集属于分类占优型数据, Heart 数据集属于均衡型数据. 在所有的数据集中, 类标属性不参与聚类过程, 只用来评估算法的聚类结果.

表 4 10 个真实数据集信息

Table 4 Ten real data sets

数据集名称	维数	数值型维数	分类型维数	类属性数	数据量
Aggregation	2	2	0	7	788
Spiral	2	2	0	3	312
Jain	2	2	0	2	373
Flame	2	2	0	2	240
Iris	4	4	0	4	150
Soybean	35	0	35	4	47
Zoo	15	1	14	7	101
Acute	7	1	6	2	120
Statlog Heart	13	5	8	2	270
KDD CUP-99	41	34	7	不定	1 000

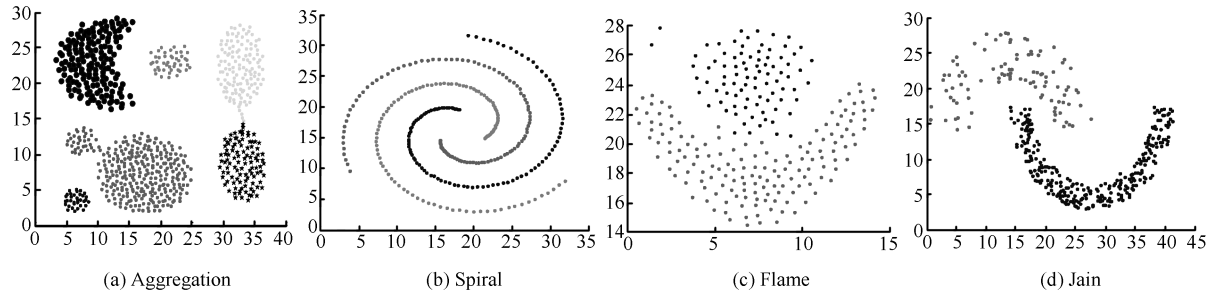


图 10　4 个数据集的聚类结果分布图

Fig. 10　Example data sets

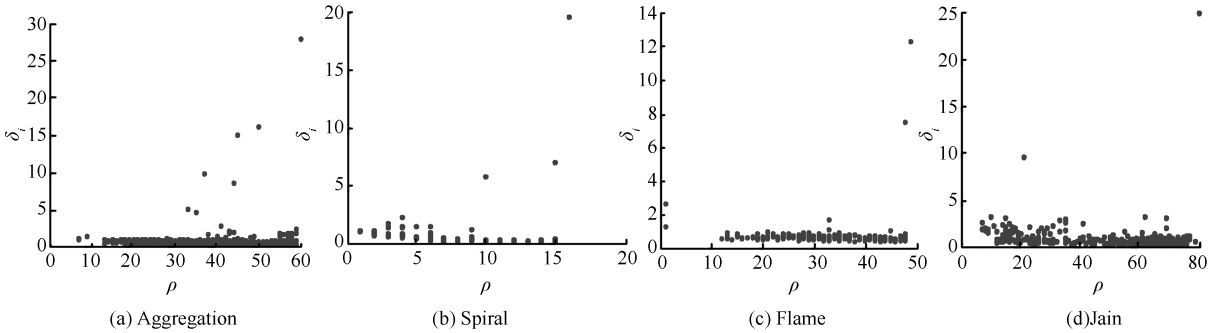


图 11　图 10 的 4 个数据集的 ρ 和 δ 分布图

Fig. 11　Density and distance distribution of example data sets

4.2.2　实验结果与性能分析

DC-MDACC 算法聚类 6 个数据集的 PSO 适应度随时间进化如图 12 所示, 算法经过少次数的迭代就能搜寻到最优解, 证明 PSO 算法的运算速度较快.

利用设计的聚类中心自动确定算法, 采用线性回归模型 $\delta^* = b_0 + b_1 \times \rho'$ 和残差分析方法确定聚类中心, 在 6 个数据集实验, 得到如图 13～图 15 所示的拟合结果和残差分布图, 因此如图 15 (a)～(f) 所示红色圈即算法自动确定的 6 个数据集的聚类中心.

本文提出的 DC-MDACC 与 IWKM 算法、SBAC 算法、K-prototypes 算法、KL-FCM-GM 算法、EKP 算法、WFK-prototypes 算法、K-means 算法和 K-prototypes 算法在相应数据集上的聚类准确率比较, 如表 5～10 所示.

以每次间隔 1 000 条记录为样本集, 我们选择了一些代表性的数据进行实验, 如当 $t = 150$ 时, 出现 “normal” 373 次, 发生了 380 次 “Satan”、5 次 “Bufoverflow”、99 次 “teardrop” 和 143 次 “Smurf” 攻击, 当 $t = 350$ 时, 出现 “normal” 381 次, 发生了 618 次 “Neptune” 攻击. 表 6 中的聚类结果表明, DC-MDACC 算法在聚类数据集 KDD-99 样本集上有较高的聚类质量.

表 5　6 种算法在 Iris 数据集上的聚类准确率

Table 5　Clustering quality evaluation on Iris dataset

算法	聚类准确率 (r)
K-prototypes	0.819
SBAC	0.426
KL-FCM-GM	0.335 ($\alpha = 1.1$)
IWKM	0.822
K-means	0.88
DC-MDACC	0.96

从表 5～表 10 中可以看出, 本文算法在 6 个数据集上的最终聚类准确率均高于 IWKM 算法、SBAC 算法、K-prototypes 算法、KL-FCM-GM 算法、EKP 算法、WFK-prototypes 算法、

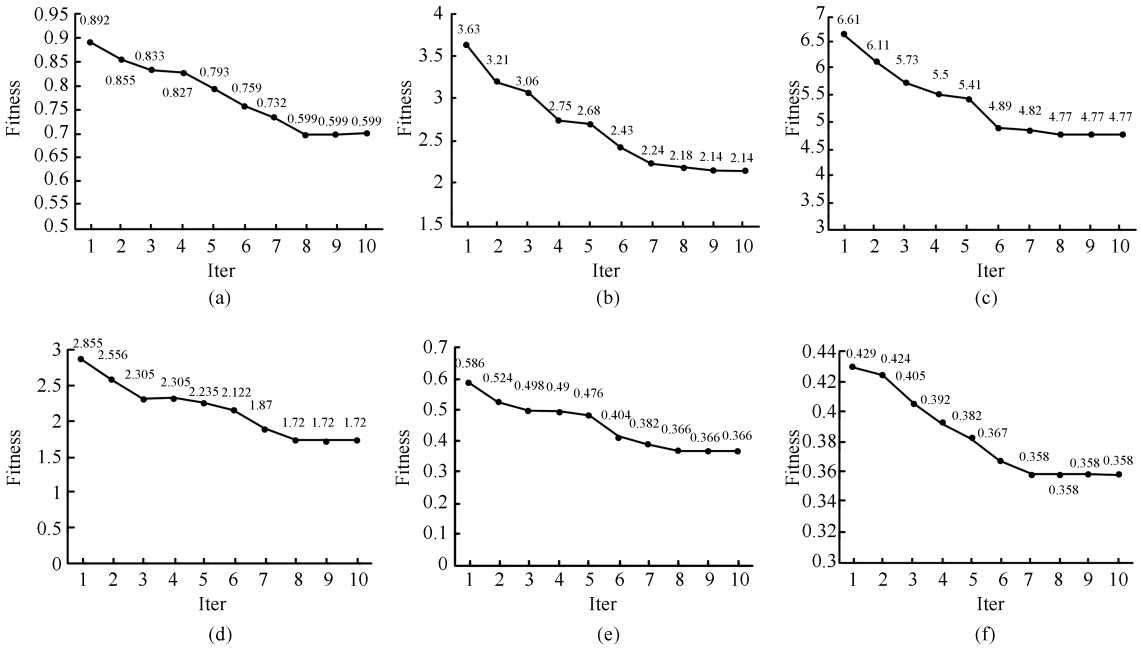


图 12 6 个数据集的算法迭代次数与适应度关系图 ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

Fig. 12 Relationship between iteration and fitness on six data sets ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

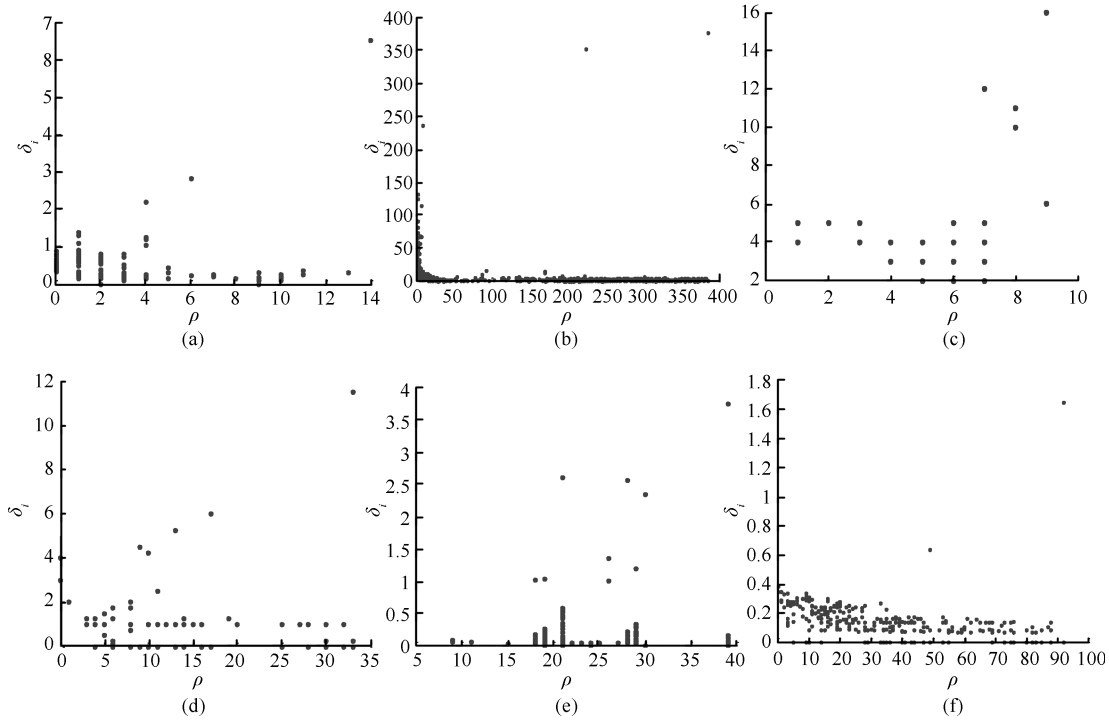


图 13 6 个数据集在最优 d_c 情况下相应 ρ 和 δ 关系图 ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

Fig. 13 Corresponding ρ and δ distribution on six data sets with optimal d_c ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

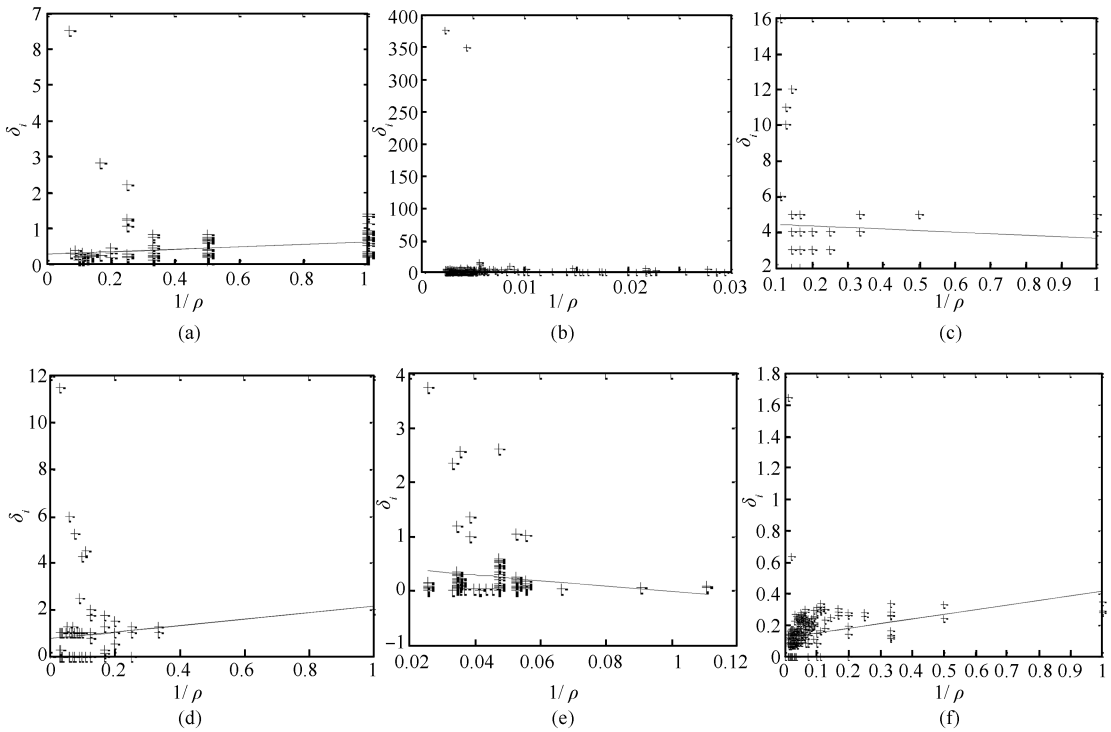


图 14　6 个数据集的线性回归模型拟合图 ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

Fig. 14　Fitting curves on six data sets with optimal d_c ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

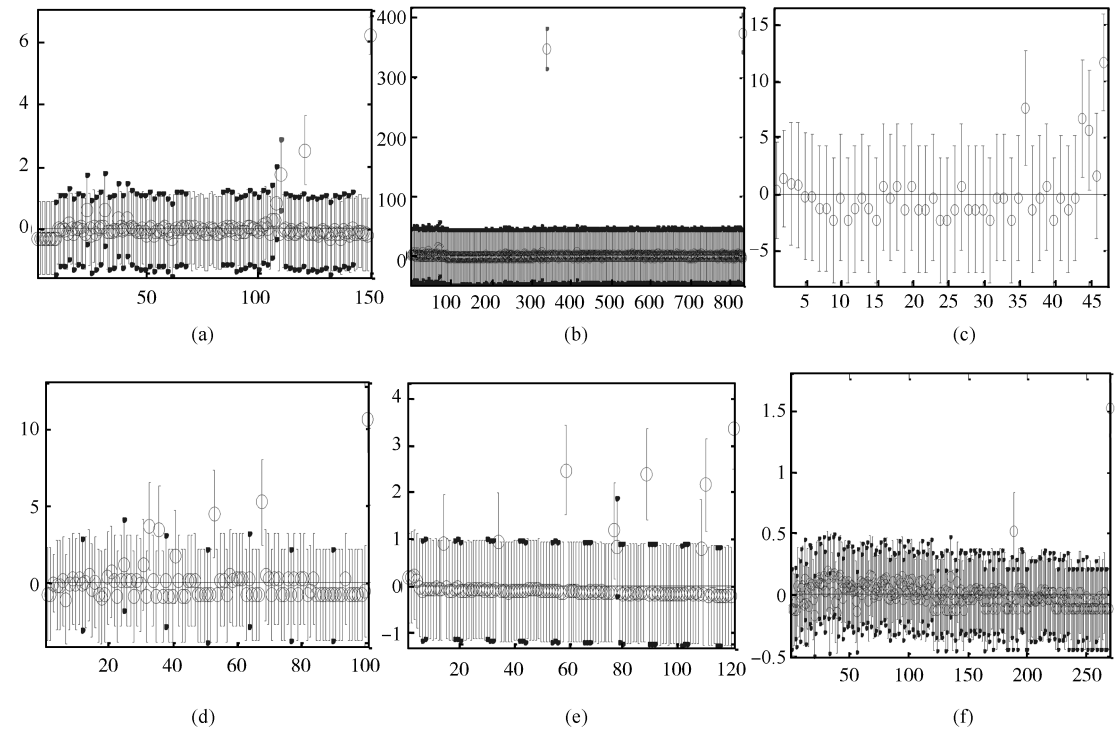


图 15　6 个数据集的残差分布图 ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

Fig. 15　Residuals distribution on six data sets with optimal d_c ((a) Iris; (b) KDD-CUP 99; (c) Soybean; (d) Zoo; (e) Acute; (f) Heart)

表 6 DC-MDACC 算法在 KDD-99 数据集上的聚类准确率
Table 6 Clustering quality evaluation on KDD-99 dataset

各攻击类型	入侵时间戳			
数量	150	250	350	450
Normal	373		381	215
Satan	380			
Bufoverflow	5			
Teardrop	99			
Smurf	143	1 000		785
Neptune			618	
Land			1	
记录总数	1 000	1 000	1 000	1 000
聚类准确率 (r)	0.977	1	0.972	0.997
聚类纯度 (Pur)	0.96	1	0.969	0.996

表 7 5 种算法在 Soybean 数据集上的聚类准确率
Table 7 Clustering quality evaluation on Soybean dataset

算法	聚类准确率 (r)
K-prototypes	0.856
SBAC	0.617
KL-FCM-GM	0.903 ($\alpha=1.8$)
IWKM	0.908
DC-MDACC	0.957

表 8 6 种算法在 Zoo 数据集上的聚类准确率
Table 8 Clustering quality evaluation on Zoo dataset

算法	聚类准确率 (r)
K-prototypes	0.806
SBAC	0.426
KL-FCM-GM	0.864 ($\alpha=1.3$)
EKP	0.629
WFK-prototypes	0.908 ($\alpha=2.1$)
DC-MDACC	0.892

表 9 6 种算法在 Acute 数据集上的聚类准确率
Table 9 Clustering quality evaluation on Acute dataset

算法	聚类准确率 (r)
K-prototypes	0.61
SBAC	0.508
KL-FCM-GM	0.682 ($\alpha=1.1$)
EKP	0.508
WFK-prototypes	0.710 ($\alpha=1.1$)
DC-MDACC	0.917

表 10 6 种算法在 Statlog Heart 数据集上的聚类准确率
Table 10 Clustering quality evaluation on Statlog Heart dataset

算法	聚类准确率 (r)
K-prototypes	0.577
SBAC	0.752
KL-FCM-GM	0.758 ($\alpha=1.7$)
EKP	0.545
WFK-prototypes	0.835 ($\alpha=1.3$)
DC-MDACC	0.848

K-means 算法和 K-prototypes 算法. 算法在 Iris、KDD-CUP 99 样本集、Soybean、Zoo、Acute Inflammations 和 Statlog Heart 的聚类纯度分别为 0.964、0.996、0.985、0.849、0.918 和 0.833, 说明本文算法在相应数据上产生的簇内纯度相对较高. 除在 Zoo 数据集上, 本文算法与 WFK-prototypes 算法相比, WFK-prototypes 算法经多次调整, 最终在 $\alpha=2.1$ 的条件下获得最高聚类准确率 90.3 %, 比本文算法高了 1.6 %, 但在其他数据集上的聚类准确率都低于本文算法.

实验证明: DC-MDACC 与这些现有优秀算法比较, 在对 6 个数据集进行聚类时, 均能够取得较高的聚类准确率. DC-MDACC 算法具有较好聚类效果的原因在于: 1) DC-MDACC 算法通过对混合数据进行占优分析, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 针对不同情况, 选择相应的距离计算方法, 数值占优和分类占优的距离计算方式是基于降低非占优属性对数据对象整体相似性的影响, 而均衡型混合属性数据需要综合考虑每一维属性的重要性, 使得 DC-MDACC 算法能够针对混合属性数据的特点, 获得较好的聚类质量. 2) DC-MDACC 算法通过计算混合属性数据的密度 ρ 及比它密度更高的点的最小距离 δ, 分析 ρ 和 δ 之间的函数关系 $\delta_i^* = f(\rho_i)$, 通过线性回归模型对 $\delta_i^* = f(\rho_i)$ 进行拟合, 利用残差分析自动确定簇类中心及簇类中心数目, 符合混合属性数据原始的数据分布, 能针对数据分布形成更好的划分.

4.3 算法执行时间

图 16 给出了本文算法和对比算法在 6 个真实数据集上的平均执行时间、算法执行时间与数据集的维度、数据量相关及算法 PSO 的迭代次数相关.

从图 16 中可以看出, Iris、Soybean、Zoo、Acute 数据集的数据量较小, 因此算法执行比较快, 而 KDD-CUP 99 数据集数据量和维数相对较大, 因

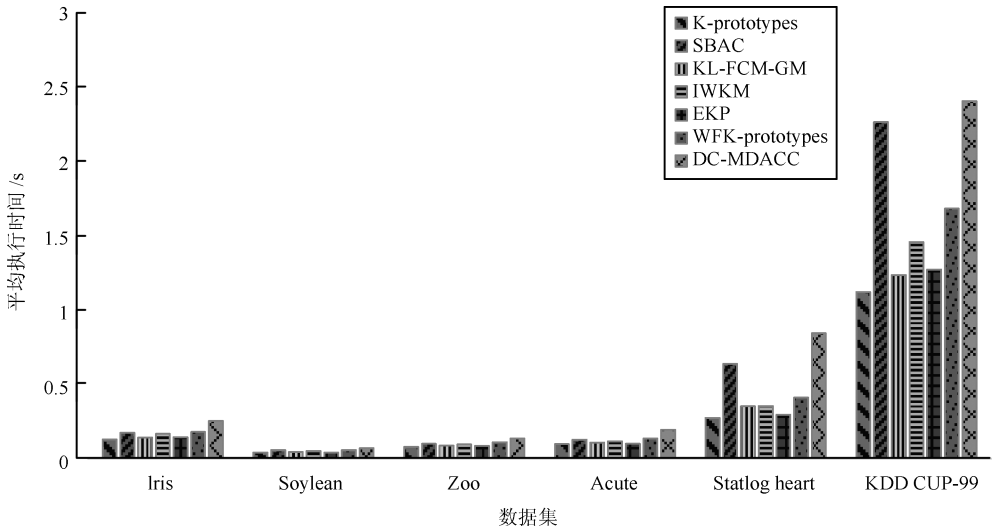


图 16　算法执行时间

Fig. 16　Average execute time

此算法执行消耗时间较长. Statlog Heart 数据由于在预处理阶段采用概率统计的方法, 需要消耗更多的时间, 因此算法执行消耗时间较长. 由于 K-prototypes、EKP 和 IWKM 等算法是基于划分的聚类算法, 其算法执行时间要低于本文聚类算法, 而算法 SBAC 是基于层次的聚类算法, 其算法执行时间与本文算法相当.

4.4　算法复杂度分析

假设聚类对象数据集规模是 n 个数据 (样本), 则 DC-MDACC 算法的时间复杂性主要由计算每个数据对象的密度与距离构成的, 该过程的计算代价分别为 $O(n^2)$ 和 $O((n^2-n)/2)$, 等聚类中心确定后, 算法只需经过一次划分就能完成聚类, 其计算代价为 $O(n\times\log(n)+n/2)$. 由于使用 PSO 算法迭代寻优, 因此算法的时间复杂性为 $O(\text{iter}\times m\times(n^2+(n^2-n)/2+ n\times\log(n)+n/2))$, 其中 iter 为 PSO 算法的迭代次数, m 为粒子群数目.

一般基于划分的聚类算法的时间复杂度是 $O(t\times k\times n)$, 通常层次聚类算法的时间复杂度为 $O(n^2)$, 其中 t 为迭代次数, k 为聚类个数, n 为数据对象个数. 表 12 列出了本文算法和对比算法的计算复杂度. 从表 12 中可以分析得到: 相比其他算法, 本文算法的时间复杂度要高, 主要消耗在迭代寻优解决参数敏感性的过程中, 但是其优势在于能够自动确定聚类中心和对于任意形态分布的数据集均能得到较满意的聚类结果, 因此可以在一定程度上弥补其时间复杂度较高的缺陷.

5　结语

本文提出了基于密度的聚类中心自动确定的混合属性数据聚类算法. 算法具有自适应性, 并且能够处理混合属性数据. 首先, 通过对混合数据进行占优分析, 将混合属性数据分为数值占优、分类占优和均衡型混合属性数据三类, 针对不同情况, 选择相应的距离计算方法, 使得算法能够针对混合属性数据的特点. 再次, 算法计算数据对象的密度及其到更高密度数据点的最小距离, 回归分析拟合密度与距离函数关系, 通过残差分析自动确定簇类中心, 符合数据原始的分布, 能够获得较好的聚类质量, 实验验证了本算法的可行性和有效性. 本文占优因子 α 的设置是通过对 UCI 数据库中混合属性数据集的测试学习得到, 具体 α 的值应通过具体问题分析而设置. 下一步的研究重点是使用本文算法对数据流实现高质量的聚类, 进一步探讨如何对混合属性数据流进行高效聚类.

表 12　算法的时间复杂度统计

Table 12　Time complexity analysis of algorithms

算法	时间复杂度
K-prototypes[19]	$O((s+1)\times k\times n)$
SBAC[23]	$O(n^2+m_c^2\log(m_c^2))$
KL-FCM-GM[[20]	$O(T\times(d\times lk+(c+2)\times n))$
EKP[22]	$O(T\times k\times n)$
IWKM[27]	$O(k\times(m+p+N\times m-N\times p)\times n\times l)$
WFK-prototypes[28]	$O(m^2\times n+m^2\times S^3+k\times(m+p+N\times m-N\times p)\times n\times s)$
DC-MDACC	$O(\text{iter}\times m\times(n^2+(n^2-n)/2+n\times\log(n)+n/2))$

References

1 Huang Z X. Extensions to the K-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 1998, **2**(3): 283−304

2 Jain A K, Dubes R C. *Algorithms for Clustering Data*. New Jersey: Prentice-Hall, 1988.

3 Han J, Kamber M. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann, 2001.

4 Chen W F, Feng G C. Spectral clustering: a semi-supervised approach. *Neurocomputing*, 2012, **77**(1): 229−242

5 Zhang W, Yoshida T, Tang X J, Wang Q. Text clustering using frequent itemsets. *Knowledge-Based Systems*, 2010, **23**(5): 379−388

6 Hsu C C, Chen C L, Su Y W. Hierarchical clustering of mixed data based on distance hierarchy. *Information Sciences*, 2007, **177**(20): 4474−4492

7 Hsu C C, Huang Y P. Incremental clustering of mixed data based on distance hierarchy. *Expert Systems with Applications*, 2008, **35**(3): 1177−1185

8 Lloyd S P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 1982, **28**(2): 129−137

9 Berget I, Mevik B H, Nas T. New modifications and applications of fuzzy C-means methodology. *Computational Statistics & Data Analysis*, 2008, **52**(5): 2403−2418

10 Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for large databases. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. Washington: ACM Press, 1998. 73−84

11 S. H. Cluster Analysis Algorithms. West Sussex: Ellis Horwood Limited, 1980.

12 Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. Montreal: ACM Press, 1996. 103−114

13 Ester M, Kriegel H P, Sander J, Xu X W. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD. 1996. 226−232

14 Bi Kai, Wang Xiao-Dan, Xing Ya-Qiong. Fuzzy clustering ensemble based on fuzzy measure and DS evidence theory. *Control and Decision*, 2015, **30**(5): 823−830
(毕凯, 王晓丹, 邢雅琼. 基于模糊测度和证据理论的模糊聚类集成方法. 控制与决策, 2015, **30**(5): 823−830)

15 Liu Z G, Pan Q, Dezert J, Mercier G. Credal C-means clustering method based on belief functions. *Knowledge-Based Systems*, 2015, **74**: 119−132

16 Huang Z X. A fast clustering algorithm to cluster very large categorical data sets in data mining. In: Research Issues on Data Mining and Knowledge Discovery. Arizona: ACM Press, 1997. 1−8

17 Gan G, Wu J, Yang Z. A genetic fuzzy K-modes algorithm for clustering categorical data. *Expert Systems with Applications*, 2009, **36**(2): 1615−1620

18 Barbara D, Couto J, Li Y. COOLCAT: an entropy-based algorithm for categorical clustering. In: Proceedings of the 11th International Conference on Information and Knowledge Management. Virginia: ACM Press, 2002. 582−589

19 Huang Z X. Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining. Singapore: World Scientific Publishing, 1997. 21−34

20 Chatzis S P. A fuzzy C-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. *Expert Systems with Applications*, 2011, **38**(7): 8684−8689

21 Gath I, Geva A B. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, **711**(7): 773−780

22 Zheng Z, Gong M G, Ma J J, Jiao L C, Wu Q D. Unsupervised evolutionary clustering algorithm for mixed type data. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Barcelona: IEEE, 2010. 1−8

23 Li C, Biswas G. Unsupervised learning with mixed numeric and nominal data. *IEEE Transactions on Knowledge and Data Engineering*, 2002, **14**(4): 673−690

24 Goodall D W. A new similarity index based on probability. *Biometrics*, 1966, **22**(4): 882−907

25 Hsu C C, Chen Y C. Mining of mixed data with application to catalog marketing. *Expert Systems with Applications*, 2007, **32**(1): 12−23

26 Ahmad A, Dey L. A K-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 2007, **63**(2): 503−527

27 Ji J C, Bai T, Zhou C G, Ma C, Wang Z. An improved K-prototypes clustering algorithm for mixed numeric and categorical data. *Neurocomputing*, 2013, **120**: 590−596

28 Ji J C, Pang W, Zhou C G, Han X, Wang Z. A fuzzy K-prototype clustering algorithm for mixed numeric and categorical data. *Knowledge-based Systems*, 2012, **30**: 129−135

29 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. *Science*, 2014, **344**(6191): 1492−1496

30 Wang Song-Gui, Shi Jian-Hong, Yin Su-Ju, Wu Mi-Xia. *Introduction to Linear Models*. Beijing: Science Press, 2004. (王松桂, 史建红, 尹素菊, 吴密霞. 线性模型引论. 北京: 科学出版社, 2004.)

陈晋音 博士, 浙江工业大学信息工程学院副教授. 主要研究方向为智能计算, 优化计算, 网络安全. 本文通信作者.
E-mail: chenjinyin@zjut.edu.cn
(**CHEN Jin-Yin** Ph. D., associate professor at the Institute of Information Engineering, Zhejiang University of Technology. Her research interest covers intelligent computing, optimization, and network security. Corresponding author of this paper.)

何辉豪 浙江工业大学信息学院硕士研究生. 数主要研究方向为据挖掘与应用, 聚类分析.
E-mail: hhh_zjut@163.com
(**HE Hui-Hao** Master student at the Institute of Information Information engineering, Zhejiang University of Technology. He received his bachelor degree from Zhejiang University of Technology in 2013. His research interest covers data mining and applications, and clustering analysis.)