第 32 卷第 8 期
2015 年 8 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 32 No. 8
Aug. 2015

DOI: 10.7641/CTA.2015.41122

基于密度和混合距离度量方法的混合属性数据聚类研究

陈晋音†, 何辉豪

(浙江工业大学 信息工程学院，浙江 杭州 310023)

摘要: 针对基于密度的传统算法不能处理混合属性数据, 以及目前的混合属性聚类算法大多数聚类质量不高等问题, 提出了基于密度和混合距离度量方法的混合属性聚类算法. 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3类, 分析不同情况的特征选取相应的距离度量方式, 通过预设参数能够发现数据密集区域, 确定核心点, 再利用核心点确定密度相连的对象实现聚类, 获得最终的聚类结果. 将算法应用于多种数据集上的实验结果表明, 该算法具有较高的聚类质量, 能够有效处理混合属性数据.

关键词: 数据挖掘; 混合属性; 聚类; 密度; 混合距离度量

中图分类号: TP391　　**文献标识码**: A

Density-based clustering algorithm for numerical and categorical data with mixed distance measure methods

CHEN Jin-yin†, HE Hui-hao

(College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310023, China)

Abstract: Traditional density-based clustering algorithm cannot deal with the mixed data, and the accuracy of most existing clustering algorithms for mixed data is not high enough as desired. To solve the problem, a density-based clustering algorithm for mixed data with mixed distance measure is proposed. Firstly, the characteristics of the mixed attribute data are analyzed, and then the data is divided into three parts: numerical dominant data, categorical dominant data and balanced data. According to the situation of dominance, corresponding distance measure method is selected. Distance between objects is calculated for finding the dense regions, and core objects are defined by preset parameters. Then, by making use of the core points to determine the objects with neighboring densities to form clusters, we obtain the final clustering result. Experiments on real data sets show that the algorithm can achieve better clustering results, and can deal with the numerical and categorical data efficiently.

Key words: data mining; mixed attributes; cluster; density; mixed distance measure methods

1 引言(Introduction)

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, 在同一个聚类形成的簇中的对象具有较高相似度, 不同簇中的对象具有较低相似度[1]. 聚类分析技术在数据挖掘、模式识别、统计等诸多领域有着广泛的应用前景, 当今人类社会在各个方面比如医疗卫生教育, 社交网站, 商场和购物网站等领域每时每刻都产生大量的数据, 这些数据大多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[2−3]. 目前的聚类算法大多用于处理单重属性的数据, 比如K−means[4], BRICH[5], DBSCAN[6], 改进DBSCAN[7], MST[8]等只针对处理数值属性数据, 而K−modes[9], COOLCAT[10]等只针对处理分类属性数据. 在处理混合属性数据时, 上述的算法都得不到期望的聚类结果.

由于混合属性数据的广泛存在, 到目前为止, 也有一些研究工作直接处理这种类型的数据. 黄哲学结合K−means和K−modes算法的思想提出了k−prototypes算法[11]来解决这个问题. 考虑到数据对象在簇归属上的不确定性, S. P. Chatzis提出了KL−FCM−GM[12]算法来扩展k-prototypes 算法, KL−FCM−GM算法是Gath−Geva算法[13]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等人引入了进化算法框架, 提出了EKP[14]算法, 该算法具有全局搜索能力. Li和Biswas等人提出了基于相似度的凝聚层次聚类算法SBAC算法[15], 该算法采用Goodall[16]提出的相似度量方法来测量数据对象间的相似性. Hsu等人提出了基于方差和熵的CAVE

收稿日期: 2014−12−03; 录用日期: 2015−05−15.

†通信作者. E-mail: chenjinyin@zjut.edu.cn.

浙江省自然科学基金项目(Y14F020092), 宁波市自然科学基金项目(2013A610070)资助.

Supported by Zhejiang Provincial Natural Science Foundation (Y14F020092) and Natural Science Foundation of Ningbo City(2013A610070).

算法[17], 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等人提出了一个K–means类型的算法[18]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. 冀进朝等人提出了IWKM[19] 和WFK–prototypes[20]算法, 考虑数据对象在簇归属上的不确定性的同时, 采用Ahmad[18]提出的属性重要性概念, 一定程度上提高了聚类精度. Hsu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[3].

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, 而使用基于密度来处理混合属性数据的方法较少. 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.

针对传统基于密度的算法不能处理混合属性数据, 以及目前的混合属性聚类算法大多数聚类质量不高的问题, 本文在传统基于密度算法的基础上提出了一种基于混合距离的混合属性密度聚类算法(density-based clustering algorithm for mixed data with mixed distance measure methods, MDCDen), 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3类, 针对不同情况, 选择相应的距离计算方法, 通过预设参数能够发现数据密集区域, 确定核心点, 再利用核心点将密度相连的对象聚成一类, 获得最终的聚类结果. 实验结果表明, MDCDen算法具有较高的聚类质量, 能够有效地处理混合属性数据聚类问题.

2 基于密度的算法及其相关定义(Density-based algorithm and definitions)

传统基于密度的聚类算法(DBSCAN)[6]是一种基于密度搜索密集区域的聚类算法, 该算法的目的通过参数ε(邻域)和μ(密度阈值)找到核心点, 从核心点出发将密度相连的数据对象聚成一类. 该算法使用数据对象间的欧式距离来度量数据对象间的相似性, 距离公式如下:

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

其中n代表数据的维数.

算法中相关概念定义如下:

定义 1 邻域. 给定数据对象的阈值参数ε内的邻域称为该数据对象的ε邻域.

定义 2 核心点. 如果一个数据对象P_i的ε邻域内的数据对象数目超过密度阈值μ, 则认为数据对象P_i为核心点.

定义 3 边界点. 如果一个数据对象P_i不是核心点, 但落在某个核心点的邻域内, 则认为数据对象P_i为边界点.

定义 4 噪声点. 如果一个数据对象P_i既不是核心点, 也不是边界点, 则认为数据对象P_i为噪声点.

定义 5 直接密度可达. 如果一个数据对象P_i是另一个数据对象P_j的ε邻域中的元素, 且P_j是核心点, 则认为数据对象P_i从数据对象P_j出发直接密度可达.

定义 6 密度可达. 如果存在一个数据链$p^1, p^2, \cdots, p^n, p^1 = q, p^n = p$, 对 于$p^i \in (1 \leqslant i \leqslant n)$, p^{i+1}是从p^i关于参数ε和μ 直接密度可达的, 则认为数据对象p是从数据对象q关于参数ε和μ密度可达的.

定义 7 密度相连. 如果存在数据对象$o \in D$, 使得数据对象p和q都是从o关于ε和μ密度可达的, 则认为数据对象p和q是关于ε和μ密度相连的.

算法根据式(1)依次地计算数据样本中的数据对象X_i与X_j之间的距离$d(X_i, X_j)$, 统计ε邻域的数据对象数量, 将X_i标记为核心点、边界点或噪声点, 直到所有的数据对象都被标记, 算法从核心点出发, 将密度可达的核心点聚成一类, 同时将其余的边界点分配到与之关联的核心点的簇中.

3 基于密度和混合计算方法的混合属性聚类算法(Density-based clustering algorithm for numerical and categorical data with mixed distance measure methods)

3.1 占优分析(Dominance analysis)

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(University of California Irvine)数据及其学习库中有56个混合属性数据集, 通过对UCI数据库中多个数据集进行测试, 如表1所示, 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.

表 1 部分UCI混合属性数据集

Table 1 Part of the UCI mixed attribute data set

UCI数据集名称	维数d	分类属性数量q	数值属性数量r	数据量	q/d	r/d	占优分析结果
Annealing	38	32	6	798	0.842	0.158	分类占优
C MethodChoice	9	7	2	1473	0.778	0.222	分类占优
Pittsburgh bridges	13	10	3	108	0.769	0.231	分类占优
Cover type	54	47	7	581012	0.87	0.13	分类占优
Adult	14	8	6	48842	0.571	0.429	均衡型
Automobile	26	11	15	205	0.423	0.578	均衡型
Credit approval	16	10	6	690	0.625	0.375	均衡型
Cylinder bands	37	16	21	512	0.432	0.568	均衡型
Flags	30	20	10	194	0.667	0.333	均衡型
Statlog heart	13	8	5	270	0.615	0.385	均衡型
Abalone	8	1	7	4177	0.125	0.875	数值占优
KDD−99	42	9	33	494031	0.214	0.786	数值占优

3.2 数据对象间距离计算方式(Distance calculation between data objects)

传统基于密度的方法只能处理数值属性数据, 欧式距离无法度量既包含数值属性数据, 又包含分类属性数据的混合属性数据, 因此本文设计了一种面向混合属性数据的距离计算方法.

3.2.1 混合属性数据的属性间关联性分析(Association analysis of mixed data attributes)

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[16].

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: $p_i(z^{\mathrm{c}}|y)$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 8　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:

$$\max d^{ij}(x,y) = P_i(z/x) + P_i(z^{\mathrm{c}}/y), \qquad (2)$$

其中z为A_j取值的子集, 这个子集能最大化式(2)的值. 注意到$p_1(z|x)$和$p_i(z^{\mathrm{c}}|y)$的取值都在[0, 1]范围内, 因此进一步修正$\max d^{ij}(x,y)$的计算为

$$\max d^{ij}(x,y) = P_i(z/x) + P_i(z^{\mathrm{c}}/y) - 1.0, \quad (3)$$

使得$\max d^{ij}(x,y)$的取值在[0, 1]范围内.

公式(3)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 9　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

$$d^i(x,y) = \frac{\sum\limits_{j=1, i\neq j}^{d} d^{ij}(x,y)}{d-1}, \qquad (4)$$

其中$d^i(x,y)$具有下述的3个属性:

1) $0 \leqslant d^i(x,y) \leqslant 1$.

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

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

本文定义的对于x和y的距离度量是基于这样的考虑: 如果某个属性的每一对属性值都能针对其他属性很好的分开, 即对所有的属性值对x, y都有一个高的距离值$d^i(x,y)$, 那么这个属性在聚类过程中的作用就很重要. 换句话说, 当属性A_i取值为x和y的时候, 对应到属性A_i上, 如果相同的公共值越多, 因为在$d^i(x,y)$的计算公式中, 相同的属性值只会出现在一个概率中, 所以$d^i(x,y)$越小, 直观来看, 相同的值越多, 必然使x 和y对应的两个对象出现在同一类的几率越高, 所以x 和y 的距离就越小.

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, 利用公式(4)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(见第3.2.2节).

对于混合属性数据的处理, 需综合考虑每一维属性对于数据聚类的重要性, 因此结合以上分析, 综合得到混合属性数据距离计算方法如下定义.

定义 10 假设待处理数据集为$D=(X_1,X_2,\cdots,X_i,\cdots,X_n)$, 每一个样本具有$d$维属性$X_i=(A_i^1,A_i^2,\cdots,A_i^r,B_i^1,B_i^2,\cdots,B_i^q)$, 其中有$r$维数值属性和$q$维分类属性, 即$d=r+q$. 则任意两个数据对象间的距离$D(X_i,X_j)$由两部分组成为

$$D\left(X_i,X_j\right)=d(X_i,X_j)_n+d(X_i,X_j)_c, \tag{5}$$

其中: $d(X_i,X_j)_n$表示数据对象X_i和X_j之间的数值型属性距离分量; $d(X_i,X_j)_c$表示数据对象X_i和X_j之间的分类型属性距离分量. 两者之和就是最终的距离.

根据第3.2.1节的分析, 不同占优类型的混合属性数据对象内部, 属性之间的相关性不同, 本文分别计算3种占优情况对应的距离计算方法.

3.2.2 数值占优混合属性数据距离计算方法(Distance calculation between numerical dominant data objects)

根据第3.1节的占优分析已知数值占优数据中数值属性维数r大于分类属性维数q, 即$r>q$; 而数据集D中的所有数据对象$X_i(i\in d)$, 如公式(4)所示, 其任意一维上对象间距离$d^i(x,y)$由$\sum\limits_{j=1,i\neq j}^{r}d^{ij}\left(x,y\right)$和$\sum\limits_{j=1,i\neq j}^{q}d^{ij}\left(x,y\right)$组成. 由于数值占优混合属性数据中$r>q$, 在整体距离计算时, 数值属性维数$r$占主导地位. 为了弱化非占优属性对整体距离计算的影响, 提高距离计算的速度, 根据数值占优混合属性数据的距离计算公式为公式(5), 其中$d(X_i,X_j)_{\rm n}$和$d(X_i,X_j)_{\rm c}$计算方法如下.

定义 11 任意两个对象X_i,X_j的数值属性部分的距离为

$$d(X_i,X_j)_{\rm n}=\sqrt{\sum_{p=1}^{m}\left(X_i^p-X_j^p\right)^2}. \tag{6}$$

定义 12 任意两个对象X_i,X_j的分类属性部分每一维的距离则采用二元化的方法, 如X_i,X_j的第p维之间的距离为

$$d(X_i^p,X_j^p)=\begin{cases}0,X_i^p=X_j^p,\\1,X_i^p\neq X_j^p,\end{cases} \tag{7}$$

则分类属性部分的距离为

$$d(X_i,X_j)_{\rm c}=\sum_{p=1}^{q}d(X_i^p,X_j^p). \tag{8}$$

3.2.3 分类占优混合属性数据距离计算方法(Distance calculation between categorical dominant data objects)

根据第3.1节的占优分析已知分类占优数据中分类属性维数q大于数值属性维数r, 即$q>r$; 而数据集D中的所有数据对象$X_i(i\in d)$, 如公式(4)所示, 其任意一维上对象间距离$d^i(x,y)$由$\sum\limits_{j=1,i\neq j}^{r}d^{ij}\left(x,y\right)$和$\sum\limits_{j=1,i\neq j}^{q}d^{ij}(x,y)$组成. 由于分类占优混合属性数据中$q>r$, 在整体距离计算时, 分类属性维数$q$占主导地位. 为了弱化非占优属性对整体距离计算的影响, 提高距离计算的速度, 根据公式(5)得到分类占优混合属性数据的距离计算方法, 其中$d(X_i,X_j)_{\rm n}$和$d(X_i,X_j)_{\rm c}$计算方法如下. 对任意数据对象X_i的数值属性部分的每一维进行标准化处理, 即X_i的第p维的值为

$$d(X_i^p)_{\rm n}=\frac{X_i^p-X_{i\_\min}^p}{X_{i\_\max}^p-X_{i\_\min}^p}, \tag{9}$$

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 13 任意两个对象X_i,X_j的数值属性部分的距离为

$$d(X_i,X_j)_{\rm n}=\sum_{p=1}^{r}(d(X_i^p)_{\rm n}-d(X_j^p)_{\rm n}). \tag{10}$$

对于分类占优数据, 其分类部分的距离计算与定义12一致.

3.2.4 均衡型混合属性数据距离计算方法(Distance calculation between balanced data objects)

根据第3.1节的占优分析已知均衡型数据中数值属性维数r和分类属性维数q的比例相对均衡. 而数据集D中的所有数据对象$X_i(i\in d)$, 其数据对象间距离由每一维属性上距离$d^i(x,y)$累加得到, 其具体定义如下:

定义 14 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

$$D\left(X_i,X_j\right)=\sum_{p=1}^{d}d^p\left(X_i,X_j\right). \tag{11}$$

3.3 共享最近邻相似度(Shared nearest neighbor similarity)

考虑到传统基于密度算法使用全局单一密度阈值, 密度阈值设置过小会产生很多簇及离群点, 密度阈值设置过大则会使得不同的簇合并, 使得聚类质量不高. 通过引入共享最近邻(shared nearest neighbors)概念来改进这一问题, 共享最近邻概念最早是由文献[21]

提出的, 由于共享最近邻相似性度反映了数据空间的局部结构, 可以根据数据分布的密度变化进行相应的伸缩, 因此它对密度的变化和空间的维度是相对不太敏感的.

本文采用不同的共享最近邻相似度计算方式, 若对象p和q相互是对方的k最近邻, 则认为对象p和q是相似的, 而两个对象之间的相似度则可以由两个对象共享的k最近邻数目来确定, 如图1所示, 两个黑点都有8个最近邻, 而其中4个是相同的, 则认为这两个点之间的共享最近邻相似度为4.

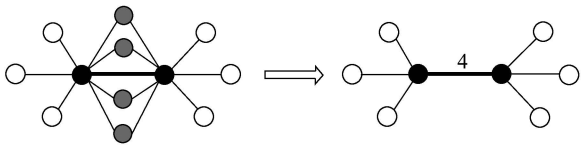

图 1 两对象之间的SNN相似度

Fig. 1 Shared nearest neighbor similarity between objects

算法 1　计算共享最近邻相似度.

For (找出所有对象的k–最近邻)

If (两个对象x和y不是相互在对方的k–最近邻中)

Similarity $(x, y) = 0$;

Else

Similarity $(x, y) =$ 共享的近邻个数;

End if

3.4 MDCDen算法详细描述(Description of MDCDen algorithm)

MDCDen算法执行时需要3个参数, 最近邻数目k, SNN相似度阈值m和密度阈值minPts, 来搜索核心点. 如果任意一个数据对象p的k最近邻中, 数据对象p与其k 最近邻中对象的SNN相似度超过阈值m的对象数目超过了密度阈值minPts, 则创建一个以数据对象p为核心点的簇, 然后通过广度搜索方式, 聚集从这些核心点出发直接密度可达的数据对象, 得到所有从数据对象p出发密度可达的数据对象, 将这些数据对象归属为一个类. 若p是核心点, 则从p出发密度可达的所有数据对象被标记为当前类, 并从他们进一步扩展. 如果p不是一个核心点, 则算法提取下一个数据对象继续处理, 依次进行, 直到找到一个完整的聚类, 然后再选择一个未被处理的核心点开始扩展, 得到下一个聚类, 依次进行, 直到所有的数据对象都被标记为止.

MDCDen算法在DBSCAN算法基础上, 结合共享最近邻相似度, 将DBSCAN中相关概念修正定义如下:

定义 15　邻域. 给定数据对象P_i, P_i与其第k个最近邻距离内的邻域称为该数据对象的邻域, 该邻域根据数据分布的密度变化进行相应的伸缩.

定义 16　核心点. 如果在一个数据对象P_i的k最近邻中, 数据对象P_i与其k最近邻中对象的SNN相似度超过阈值m的对象数目超过了密度阈值minPts, 则认为数据对象P_i 为核心点.

定义 17　边界点. 如果一个数据对象P_i不是核心点, 但与某个核心点的SNN相似度超过阈值m, 则认为数据对象P_i为边界点.

定义 18　直接密度可达. 如果一个数据对象P_i是另一个数据对象P_j的k–最近邻中的元素, P_j是核心点, 且P_i与P_j的SNN相似度超过阈值m, 则称数据对象P_i从数据对象P_j出发时直接密度可达.

定义 19　密度可达. 如果存在一个数据链p^1, $p^2, \cdots, p^n$, $p^1 = q$, $p^n = p$, 对于$p^i \in (1 \leqslant i \leqslant n)$, p^{i+1}是从p^i关于参数k, m和minPts直接密度可达的, 则认为数据对象p是从数据对象q关于参数k, m和minPts密度可达的.

定义 20　密度相连. 如果存在数据对象$o \in D$, 使得数据对象p和q都是从o关于参数k, m和minPts密度可达的, 则数据对象p和q是关于参数k, m和minPts密度相连的.

其他定义与DBSCAN算法相同.

3.4.1 预处理过程(Pretreatment)

通过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3类, 针对不同情况, 选择不同的距离计算方式, 所以本文预先对混合属性数据进行占优分析, 并对相应属性进行处理, 执行操作见算法2.

算法 2　Pretreatment ( )

//给定数据集D, 每一个样本具有d维属性, 其中有r 维数值属性与q维分类属性.

If $(r/d \in [0.75, 1])$

For (数据点的每一维)

数据点信息根据定义8, 定义9处理, 并将新结果保存;

Else if $(q/d \in [0.75, 1])$

For (数据点的每一维)

数据点信息根据定义8, 定义10处理, 并将新结果保存;

Else if $(r/d \in (0.25, 0.75)$ or $q/d \in (0.25, 0.75))$

For (数据点的每一维)

数据点信息根据定义12, 定义13处理, 并将新结果保存;

End

3.4.2 确定核心点(Determine the core points)

对于任意数据点X_i, 根据相应距离计算方式确定数据点X_i的k–最近邻, 如果其k–最近邻中对象与数据点X_i的SNN相似度超过阈值m, 将该数据点放入X_i 邻域内. 执行操作见算法3.

在确定核心点时实现参数的选取, 方法如下:

1) 对于数值占优数据和分类占优数据的聚类情况, 算法设计时根据第3.2节的分析, 其计算距离时可以不使用SNN, 仅采用DBSCAN算法实现聚类, 因此参数ε和minPts的选取影响聚类效果. 本文采用自适应DBSCAN算法[22–23], 解决了ε和minPts参数选取的问题. 假定minPts 值的情况下, minPts 值的选取与DBSCAN算法一致, 即$\mathrm{minPts}=4$, 而使用多个不同的ε参数进行试聚类, 然后评估各次聚类的有效性, 从中选择最优的ε参数值.

2) 对于均衡型混合属性数据的聚类情况, 算法需要3个参数最近邻数目k, SNN相似度阈值m和密度阈值 minPts. 参数 k 和 m 的作用与参数 ε 相对应, 但k和m参数的加入使得数据对象的邻域能够根据数据分布的密度变化进行相应的伸缩, 其具体的参数设置方法与处理数值占优数据和分类占优数据时一致.

算法 3 SetNearPoints( )

For (任意数据点X_i)
  找到X_i的k个最近邻.
For (任意数据点X_i)
  For (任意数据点X_j)
    If (Similarity$(X_i, X_j) > m$)
      把X_j放入到X_i的邻域内;
    End if
For (任意数据点X_i)
  If (X_i邻域内数量$\geqslant \mathrm{minPts}$)
    X_i是核心点;
  End if
End

3.4.3 最终聚类(Final cluster)

从任意核心点出发遍历找到与其密度相连的数据点, 将其聚成一类. 直至所有数据点都被处理. 具体步骤见算法4.

算法 4 Do-cluster( )

输入: 核心点及其邻域信息.
输出: 数据对象及其类标号.
do
  得到一个未处理的数据点P;
  If (p是核心点) then
  找出所有到p密度相连的数据点;
  利用它们形成一个自然簇;
  Else
    Break;
End if;
Until所有的数据点都被处理.

3.4.4 算法复杂度分析(Complexity analysis)

假设聚类对象数据集规模是m个数据(样本), 则基本DBSCAN算法的时间复杂度为$\mathrm{O}(m2)$, 主要消耗在寻找ε邻域内中的点. 而本文提出的MDCDen算法的时间复杂性主要由计算数据点的k近邻及其邻域中的点构成的, 该过程的计算代价为$\mathrm{O}(m2 + m * k3)$, 其中k是数据点的最近邻数目.

一般基于划分的聚类算法的时间复杂度是$\mathrm{O}(t*k*m)$, 通常层次聚类算法的时间复杂度为$\mathrm{O}(m2)$, 其中: t为迭代次数, k 为聚类个数, m为数据对象个数. 从以上理论分析得到结论相比于基于划分和基于层次聚类, 基于密度的聚类算法复杂度要高, 但是其优势在于单次扫描无序回溯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高的缺陷. 而本文提出的MDCDen算法与DBSCAN相比, 增加了k近邻的计算代价, 这种情况只出现在均衡型混合数据聚类的计算上, 对于另外两种占优情况(数值型占优和分类型占优)都不会额外增加时间复杂度, 因此时间复杂度增加也在可以接受的范围内.

在空间复杂度计算上, 因为对每个聚对象点, MDCDen算法与DBSCAN算法只需要维持少量数据, 即簇标号和数据点的标识. 所以不论是高维还是低维数据, 其空间复杂度均为$\mathrm{O}(m)$.

4 实验结果与分析(Experimental results and analysis)

实验中的操作系统为Windows7, 集成开发环境为Microsoft Visual C++ 2010. 硬件条件为: CPU为Intel Core I5 2.6 GHz, 内存为4 GB.

为了验证新算法MDCDen的性能, 本文使用6个真实的数据集, 这6个数据集均来自UCI及其学习库(machine learning repository), 具体信息如表2所示.

表 2 6个真实数据集信息

Table 2 Description of six real datasets

数据集	维数	数值型维数	分类型维数	类属性数	数据量
Iris	4	4	0	4	150
Soybean	35	0	35	4	47
Zoo	15	1	14	7	101
Acute	7	1	6	2	120
Heart	13	5	8	2	270
KDD–99	41	32	9	不定	1000

4.1 聚类结果评价(Evaluation of clustering results)

1) 本文采用由Huang和Ng提出的[9]聚类准确率作为评价标准, 聚类准确率r的定义如下:

$$r=\frac{\sum_{i=1}^{k}a_i}{n},\tag{12}$$

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:

$$\mathrm{Pur}=\sum_{i=1}^{k}\frac{\left|C_i^d\right|}{\left|C_i\right|}/K,\tag{13}$$

其中: K表示为簇的个数, $\left|C_i^d\right|$表示在簇i中具有该簇最主要类标号的数据点数, $\left|C_i\right|$表示簇i中包含的所有数据点的个数. 平均聚类纯度越高, 算法的聚类效果越好.

4.2 实验结果分析(Clustering results analysis)

实验 1　数值占优数据.

Iris数据集包含150个数据对象, 每个数据对象由4个数值属性描述. Iris数据集有3个类属性: Iris-Setosa, Iris-Versicolour和Iris-Virginica. 在所有的数据集中, 类属性不参与聚类过程, 只用来评估算法的聚类结果. 图2给出了MDCDen算法在聚类Iris数据集, 设置$\varepsilon=0.4$时, 密度阈值minPts对算法聚类准确率的影响.

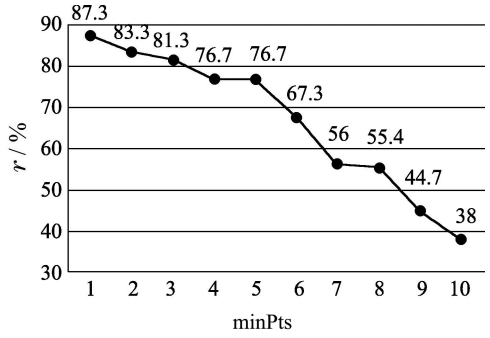

图 2 密度阈值minPts对MDCDen算法在Iris数据集上聚类准确率的影响

Fig. 2 The impact of the density threshold minPts on the accuracy of our proposed algorithm for clustering Iris data

MDCDen算法, IWKM算法, SBAC算法, KP算法和KL–FCM–GM算法的聚类准确率(r)在表3中列出.

从表3和图2中可以看出, 算法KP, SBAC, IWKM的聚类准确率分别为0.819, 0.426和0.822; 而MDCDen算法在$\varepsilon=0.4$, $\mathrm{minPts}=1$时聚类准确率最高为0.873, 而KL–FCM–GM算法在模糊系数为1.1时聚类准确率最高为0.335. 表3中的聚类结果表明, MDCDen算法的聚类准确率比KP, SBAC, KL–FCM–GM和IWKM算法分别高出了5.4%, 44.7%, 53.8%和5.1%. 因此MDCDen算法的性能更好.

表 3 5种算法在Iris数据集上的聚类准确率

Table 3 Clustering accuracy for clustering Iris data of five algorithms

算法	r
K–prototypes	0.819
SBAC	0.426
KL–FCM–GM	0.335 ($\alpha=1.1$)
IWKM	0.822
MDCDen	0.873

KDD–CUP 99网络入侵数据集包含494031条记录. 每条记录含有41维属性描述, 其中有34维数值属性和7维分类属性, 每条记录已经被甄别为5个大类, 24个小类, 包括正常的链接和各种不同的入侵和攻击.

以每次间隔1000条记录为样本集, 本文选择了一些代表性的数据进行试验, 如当t为150时, 出现正常情况normal373次, 出现Satan, Bufoverflow, teardrop和Smurf的频率分别为380, 5, 99和143次; 当t为350时, 出现正常情况normal381 次, 出现Neptune的频率为618次.

表4中的聚类结果表明, MDCDen算法在数据集KDD–Cup 99的样本集上有较高的聚类质量.

表 4 MDCDen算法在KDD–Cup 99样本数据集上的聚类准确率

Table 4 Clustering accuracy for clustering KDD–CUP 99 sample data of MDCDen

各攻击类型数量	入侵时间戳			
	150	250	350	450
Normal	373		381	215
Satan	380			
Bufoverflow	5			
Teardrop	99			
Smurf	143	1000		785
Neptune			618	
Land			1	
记录总数	1000	1000	1000	1000
r	0.962	1	0.972	0.981

实验 2　分类占优数据.

Soybean数据集是分类属性数据集, 由47个数据对象构成, 每个数据对象由35个分类属性描述. Soybean数据集有4个类属性值. 图3给出了MDCDen算法在聚类Soybean数据集, 设置$\varepsilon=5$时, 密度阈值minPts对算法聚类准确率的影响.

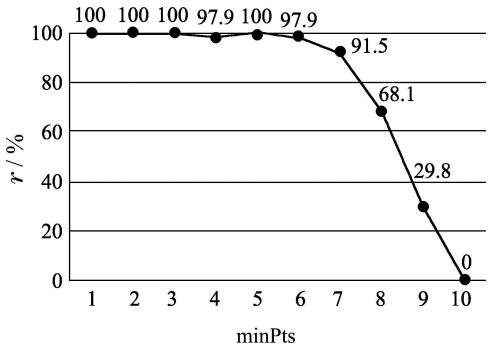

图 3 密度阈值minPts对MDCDen算法在Soybean数据集上聚类准确率的影响

Fig. 3 The impact of the density threshold minPts on the accuracy of our proposed algorithm for clustering Soybean data

MDCDen算法, IWKM算法, SBAC算法, KP算法和KL–FCM–GM算法的聚类准确率在表5中列出.

表 5 5种算法在Soybean数据集上的聚类准确率

Table 5 Clustering accuracy for clustering Soybean data of five algorithms

算法	r
K–prototypes	0.856
SBAC	0.617
KL–FCM–GM	0.903 ($\alpha = 1.8$)
IWKM	0.908
MDCDen	1

从表5和图3中可以看出, 算法KP, SBAC, IWKM的聚类准确率分别为0.856, 0.617和0.908; 而MDCDen算法在$\varepsilon = 5$, $\mathrm{minPts} = 3$时聚类准确率最高为1.0, KL–FCM–GM在模糊系数为1.8时聚类准确率最高为0.903. 表5中的聚类结果表明, MDCDen算法的聚类准确率比KP, SBAC, KL–FCM–GM和IWKM算法分别高出了14.6%, 38.3%, 9.7%和9.2%. 因此MDCDen算法的性能更好.

Zoo数据集包含101个数据对象, 每个数据对象由一个数值属性和15个分类属性描述. Zoo数据集有7个类属性值. 图4给出了MDCDen算法在聚类Zoo数据集, 设置$\varepsilon = 2.6$时, 密度阈值minPts对算法聚类准确率的影响.

MDCDen算法, WFK–P算法, EKP算法, SBAC算法, KP算法和KL–FCM–GM算法的聚类准确率在表6中列出.

从表6和图4中可以看出, 算法KP, SBAC, EKP的聚类准确率分别为0.806, 0.426和0.629; 而MDCDen算法在$\varepsilon = 2.6$, $\mathrm{minPts} = 6$时聚类准确率最高为0.931, KL–FCM–GM在模糊系数为1.3时聚类准确率最高为0.864. WFK–P在模糊系数为2.1时聚类准确率最高为0.908. 表6中的聚类结果表明, MDCDen算法的聚类准确率比KP, SBAC, KL–FCM–GM, EKP和WFK–P算法分别高出了12.5%, 50.5%, 6.7%, 30.2%和2.3%. 因此MDCDen算法的性能更好.

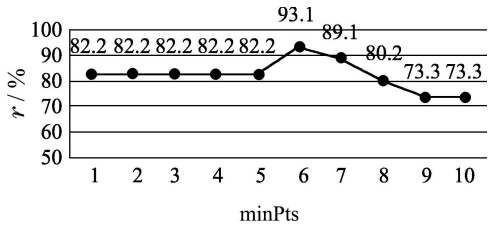

图 4 密度阈值minPts对MDCDen算法在Zoo数据集上聚类准确率的影响

Fig. 4 The impact of the density threshold minPts on the accuracy of our proposed algorithm for clustering Zoo data

表 6 6种算法在Zoo数据集上的聚类准确率

Table 6 Clustering accuracy for clustering Zoo data of six algorithms

算法	r
K–prototypes	0.806
SBAC	0.426
KL–FCM–GM	0.864 ($\alpha = 1.3$)
EKP	0.629
WFK–prototypes	0.908 ($\alpha = 2.1$)
MDCDen	0.931

Acute Inflammatins数据集包含120个数据对象, 每个数据对象由1个数值属性和6个分类属性描述. Acute数据集有2个类属性值. 图5给出了MDCDen算法在聚类Acute Inflammations数据集, 设置$\varepsilon = 0.8$时, 密度阈值minPts对算法聚类准确率的影响.

MDCDen算法, WFK–P算法, EKP算法, SBAC算法, KP算法和KL–FCM–GM算法的聚类准确率在表7中列出.

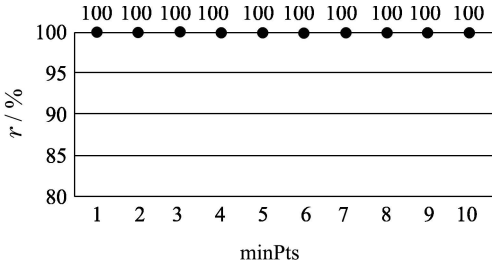

图 5 密度阈值minPts对MDCDen算法在Acute数据集上聚类准确率的影响

Fig. 5 The impact of the density threshold minPts on the accuracy of our proposed algorithm for clustering Acute data

表 7 6种算法在Acute数据集上的聚类准确率

Table 7 Clustering accuracy for clustering Acute data of six algorithms

算法	r
K–prototypes	0.61
SBAC	0.508
KL–FCM–GM	0.682 ($\alpha = 1.1$)
EKP	0.508
WFK–prototypes	0.710 ($\alpha = 1.1$)
MDCDen	1

从表7和图5中可以看出, 算法KP, SBAC, EKP的聚类准确率分别为0.610, 0.508和0.508; 而MDCDen算法在$\varepsilon = 0.8$, minPts取值为1–10内聚类准确率最高为1.0, KL–FCM–GM在模糊系数为1.1时聚类准确率最高为0.682.WFK–P在模糊系数为1.1时聚类准确率最高为0.710.表7中的聚类结果表明, MDCDen算法的聚类准确率比KP, SBAC, KL–FCM–GM, EKP和WFK–P算法分别高出了39%, 49.2%, 31.8%, 49.2%和29%. 因此MDCDen算法的性能更好.

实验 3　均衡型混合属性数据. Statlog Heart数据集包含270个数据对象, 每个数据对象由5个数值属性和9个分类属性描述. Statlog Heart有2个类属性值. MDCDen 算法, WFK–P算法, EKP算法, SBAC算法, KP 算法和KL–FCM–GM算法的聚类准确率在表8中列出.

表 8 6种算法在Heart数据集上的聚类准确率

Table 8 Clustering accuracy for clustering Heart data of six algorithms

算法	r
K–prototypes	0.577
SBAC	0.752
KL–FCM–GM	0.758 ($\alpha = 1.7$)
EKP	0.545
WFK–prototypes	0.835 ($\alpha = 1.3$)
MDCDen	0.729

MDCDen算法在$\text{minPts} = 2, m = 9, k = 20$时, 聚类准确率最高为0.729.从表8中可以看出, 算法KP, SBAC, EKP的聚类准确率分别为0.577, 0.752和0.545; 而KL–FCM–GM在模糊系数为1.7时聚类准确率最高为0.758. WFK–P在模糊系数为1.3时聚类准确率最高为0.835. 算法MDCDen的准确率比KP, EKP算法的聚类准确率高, 略低于KL–FCM–GM, SBAC算法, 但比WFK–P算法的准确率低了10.6%. 考虑到基于密度的算法会排除一些可能代表噪声, 离群点以及没有很强连接的对象, 属于不完全聚类, 所以在表9 中给出MDCDen算法在平均聚类纯度指标下的聚类质量和离群点比例.

表 9 不同参数下MDCDen算法的Purity及离群点比例

Table 9 Purity and outlier ratio of MDCDen with different parameters

minPts	m	k	Purity/%	离群点比例/%
3	8	18	86.3	16.7
3	**8**	**19**	**91.7**	**13.7**
2	9	19	86.7	19.3
2	**9**	**20**	**88.2**	**12.6**
1	9	20	81	10.4
1	**10**	**22**	**85.3**	**10.4**

从表9中可以看出, 在MDCDen算法除离群点以外的数据对象上的聚类结果大多数有超过85%的平均聚类纯度, 说明簇内的聚类质量较好. 使用聚类准确率作为评价标准时, 由于离群点的数量均被当作未被正确分类的数据对象处理, 使得MDCDen算法的聚类准确率略低.

密度阈值minPts决定了核心点在数据集中的比例, 在确定最优ε参数后, 将minPts逐步增大来观察聚类质量的变化情况, 图2–5显示了实验结果. 针对不同的数据集, minPts的值设定有不同的结果. 然而, minPts不宜被设置的过大, 随着minPts的增大, 聚类质量显著降低, 这是因为密度阈值minPts设置过高导致核心点的数量急剧减少, 部分相对密集的区域未被发现, 同时大量数据对象被当作噪声点处理, 使得聚类质量有所下降.

表2–9中的实验结果表明, 和其他算法相比, 算法MDCDen能够取得较高的聚类准确率, 因此算法的性能更好. MDCDen算法具有较好聚类效果的原因在于, MDCDen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3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MDCDen算法能够针对混合属性数据的特点, 获得较好的聚类质量.

4.3 算法执行时间(Execution time)

表10列出了本文算法在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与数据量相关. Iris, Soybean, Zoo 和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, 因此算法执行消耗时间较长. Statlog Heart数据由于使用SNN密度, 在计算k近邻时需要消耗更多的时间, 因此算法执行消耗时间更多.

表 10 MDCDen算法对各个数据集聚类时间复杂度统计

Table 10 Execution time of MDCDen on different datasets

数据集名称	平均执行时间/ms
Iris	94
KDD–CUP 99	6414
Soybean	56
Zoo	78
Acute	63
Statlog Heart	2776

5 结语(Conclusions)

本文提出基于混合距离的混合属性密度聚类算法, 该算法在传统基于密度算法的基础上进行扩展, 无需预设聚类个数、能够处理噪声点, 能够发现任意形状的簇. 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, 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对海量数据实现高质量的聚类, 进一步探讨如何对混合属性数据流进行高效聚类.

参考文献(References):

[1] HAN J, KAMBER M. *Data Mining Concepts and Techniques* [M]. San Francisco: Morgan KaufmannJ, 2001.

[2] HSU C C, CHEN C L, SU Y W. Hierarchical clustering of mixed data based on distance hierarchy [J]. *Information Sciences*, 2007, 177(20): 4474 – 4492.

[3] HSU C C, HUANG Y P. Incremental clustering of mixed data based on distance hierarchy [J]. *Expert Systems with Applications*, 2008, 35(3): 1177 – 1185.

[4] LLOYD S P. Least square quantization in PCM [J]. *IEEE Transactions on Information Theory*, 1982, 28(2): 129 – 137.

[5] ZHANG T, RAMAKRISHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases [C] //*Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*. Montreal: ACM Press, 1996: 103 – 114.

[6] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C] //*Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon: AAAI Press, 1996: 226 – 231.

[7] 李春生, 王耀南. 聚类中心初始化的新方法 [J]. 控制理论与应用, 2010, 27(10): 1435 – 1440.
(LI Chunsheng, WANG Yaonan. New initialization method for cluster center [J]. *Control Theory & Applications*, 2010, 27(10): 1435 – 1440.)

[8] 谭建豪, 章兢, 李伟雄. 密度分布函数在聚类算法中的应用 [J]. 控制理论与应用, 2011, 28(12): 1791 – 1796.
(TAN Jianhao, ZHANG Jing, LI Weixiong. Application of density distribution function in clustering algorithms [J]. *Control Theory & Applications*, 2011, 28(12): 1791 – 1796.)

[9] HUANG Z. A fast clustering algorithm to cluster very large categorical data sets in data mining [C] //*Research Issues on Data Mining and Knowledge Discovery*. Arizona: ACM Press, 1997: 1 – 8.

[10] BARBARA D, COUTO J, LI Y. COOLCAT: an entropy-based algorithm for categorical clustering [C] //*Proceedings of the 11th International Conference on Information and Knowledge Management*. Virginia: ACM Press, 2002: 582 – 589.

[11] HUANG Z. Clustering large data sets with mixed numeric and categorical values [C] //*The first Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Singapore: World Scientific Publishing, 1997: 21 – 34.

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

[13] GATH I, GEVA A B. Unsupervised optimal fuzzy clustering [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, 711(7): 73 – 781.

[14] ZHENG Z, GONG M, MA J, et al. Unsupervised evolutionary clustering algorithm for mixed type data [C] //*Proceedings of the 2010 IEEE Congress on Evolutionary Computation*. Barcelona: CEC, 2010: 1 – 8.

[15] LI C, BISWAS G. Unsupervised learning with mixed numeric and nominal data [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2002, 14(4): 673 – 690.

[16] GOODALL D W, A new similarity index based on probability [J]. *Biometrics*, 1966, 22(4): 882 – 907.

[17] HSU C C, CHEN Y C. Mining of mixed data with application to catalog marketing [J]. *Expert Systems with Applications*, 2007, 32(1): 12 – 23.

[18] AHMAD A, DEY L. A k-mean clustering algorithm for mixed numeric and categorical data [J]. *Data & Knowledge Engineering*, 2007, 63(2): 503 – 527.

[19] CHAO J, PANG W, ZHOU C G, et al. An improved k-prototypes clustering algorithm for mixed numeric and categorical data [J]. *Neurocomputing*, 2013, 120(1): 590 – 596.

[20] CHAO J, PANG W, ZHOU C G, et al. A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data [J]. *Knowledge-Based Systems*, 2012, 30(1): 129 – 135.

[21] JARVIS R A, PATRICK E A. Clustering using a similarity measure based on shared nearest neighbors [J]. *IEEE Transactions on Computers*, 1973, 22(11): 1025 – 1034.

[22] HALKIDI M, VAZIRGIANNIS M. Clustering validity assessment: finding the optimal partitioning of a data set [C] //*Proceedings of the 2001 IEEE International Conference on Data Mining*. Califomia: IEEE, 2001: 187 – 194.

[23] FENG P J, GE L D. Adaptive DBSCAN based algorithm for constellation reconstruction and modulation identification [C] //*Proceedings of Radio Science Conference*. Beijing: Publishing House of Electronics Industry, 2004: 177 – 180.

作者简介:

陈晋音 (1982–), 女, 博士，副教授, 研究领域包括智能计算、优化计算、网络安全等, E-mail: chenjinyin@zjut.edu.cn;

何辉豪 (1990–), 男, 硕士研究生, 研究领域包括数据挖掘与应用、聚类分析, E-mail: hhh_zjut@163.com.