文章编号: 1000−8152(2011)12−1791−06

密度分布函数在聚类算法中的应用

谭建豪, 章 兢, 李伟雄

(湖南大学 电气与信息工程学院, 湖南 长沙 410082)

摘要: 深入分析了传统的基于密度的聚类方法的特点和存在的问题及讨论了基于密度聚类算法研究现状, 提出了一种改进的基于密度分布函数的聚类算法. 使用K最近邻(KNN)的思想度量密度以寻找当前密度最大点, 即中心点. 并使用区域比例, 将类从中心点开始扩展, 每次扩展的同时引入半径比例因子以发现核心点. 再从该核心点的KNN扩展类, 直至密度下降到中心点密度的给定比率时结束. 给出了数个算法实例并与基于网格的共享近邻聚类(GNN)算法在聚类准确率和效率上进行了试验比较, 试验表明该算法极大降低了基于密度聚类算法对参数的敏感性、改善了对高维密度分布不均数据集的聚类效果、提高了聚类准确率和效率.

关键词: 聚类算法; KNN; GNN; 密度分布函数; OPTICS; DENCLUE; 区域比例; 半径比例因子

中图分类号: TP181, TP182 **文献标识码**: A

Application of density distribution function in clustering algorithms

TAN Jian-hao, ZHANG Jing, LI Wei-xiong

(Electrical and Information Engineering College, Hunan University, Changsha Hunan 410082, China)

Abstract: Characteristics and disadvantages of traditional density-based clustering algorithms are deeply investigated; the present research status of density-based clustering algorithms is discussed; an improved clustering algorithm based on density distribution function is put forward. K nearest neighbor (KNN) is used to measure the density of each point; a local maximum density point is defined as the center point. By means of local scale, classification is extended from the center point. For each point there is a procedure to determine whether it is a core point by a radius scale factor. The classification is extended once again from the core point until the density descends to the given ratio of the density of the center point. Several algorithm examples are given and the algorithm is experimentally compared with the grid-shared nearest neighbor (GNN) clustering algorithm, on the clustering accuracy ratio and efficiency. The tests show that the improved algorithm greatly reduces the sensitivity of density-based clustering algorithms to parameters, improves the clustering effect of the high-dimensional data sets with uneven density distribution, and enhances the clustering accuracy and efficiency.

Key words: clustering algorithms; KNN; GNN; density distribution function; OPTICS(ordering points to identify the clustering structure); DENCLUE(density-based clustering); local scale; radius scale factor

1 引言(Introduction)

聚类分析是数据挖掘领域中一个非常活跃的研究课题. 聚类分析在模式识别、图像处理、市场营销等领域都有广泛的应用. 一些算法只能发现球状簇, 在发现任意形状簇时遇到了困难. 基于密度的聚类方法的提出, 为解决任意形状数据集的聚类提供了一种参考的思路. 在基于密度的聚类方法中, 典型的算法有: DBSCAN(density based spatial clustering of applications with noise), OPTICS(ordering points to identify the clustering structure)和DENCLUE(density-based clustering). DBSCAN将密度超过某一阈值的区域划分为簇, 具有在含有“噪声”的空间数据库中发现任意形状的簇的能力. 但该算法需要由用户根据经验确定输入参数, 这在真实的高维数据集合中, 变得不太现实. 此外, 由于这类算法对参数值非常敏感, 参数值的微小变化往往会导致差异很大的聚类结果[1~3]. OPTICS是一种自动交互式的聚类分析方法, 该方法用簇次序代表基于密度聚类的数据结构, 该结构包含了从一个宽广的参数设置范围进行基于密度聚类所需的信息. 由于OPTICS与DBSCAN在结构上是等价的, 该算法具有的时间复杂度与DBSCAN相同. DENCLUE聚类分析方法是一个基于一组密度分布函数的聚类算法, 能更形式化地定义中心定义的簇和任意形状的簇, 其运算速度比DBSCAN和OPTICS都快. 和DBSCAN一样, 由于该方法要求仔细选择密度参数和噪声阈值, 因而具有参数敏感性问题[4]. 本文结合OPTICS和DENCLUE的特点, 提出一种改进的基于密度分布

收稿日期: 2010−02−04; 收修改稿日期: 2011−02−27.

基金项目: 国家自然科学基金资助项目(60634020); 湖南省自然科学基金资助项目(08JJ3132); 中央高校基本科研业务费资助项目.

函数的聚类方法. 使用K最近邻的思想度量密度以寻找当前密度最大点为中心点, 并使用区域比例, 将类扩展到由比例因子决定的密度边缘, 每次扩展的同时引入半径比例因子以发现核心点. 实例证明, 该算法是一种自动和交互的快速聚类方法.

2 基于密度的聚类算法研究现状(Present research status of density-based clustering algorithms)

2.1 自动交互式的聚类分析方法(OPTICS, ordering points to identify the clustering structure)

OPTICS是一种自动交互式的聚类分析方法, 它没有显式地产生一个数据集合簇. 它用簇次序代表基于密度聚类的数据结构, 该结构包含了从一个宽广的参数设置范围进行基于密度聚类所需的信息[5].

为了生成基于密度聚类的集合或次序, 通过扩展DBSCAN算法来同时处理一组距离参数值. 为了同时构建不同的聚类, 该算法以特定顺序处理核心对象. 为首先完成高密度区域的聚类, 算法依据该次序选择具有最小的ε值密度可达的对象. 核心对象包含两个重要参数—核心距离(core-distance)和可达距离(reachability-distance).

1) 核心距离. 使对象p成为核心对象的最小ε. 如果p不是核心对象, 则不包含该参数.

2) 可达距离. 对象q关于对象p的可达距离是p的核心距离与p和q之间的欧几里的距离相比之较大值. 如果p不是核心对象, 则不包含该参数.

OPTICS算法创建了数据库中对象的一个次序, 额外存储了每个对象的核心距离和一个适当的可达距离, OPTICS依据这个次序信息来抽取聚类[1~4].

2.2 基于密度分布函数的聚类算法(DENCLUE, density-based clustering)

DENCLUE是基于密度分布函数的聚类算法. 该算法主要基于下列思想[6]: 1) 每个数据点在邻域内的影响可以用一个被称为影响函数(influence function)的数学函数来形式化地模拟; 2) 数据空间的整体密度可以被模型化为所有数据点的影响函数的总和; 3) 可以通过确定密度吸引点(density attractor)来实现聚类, 密度吸引点是全局密度函数的局部最大.

对一个连续可微的密度函数, 可导出其梯度函数. 而一个用梯度指导的爬山算法能用来计算一组数据点的密度吸引点. 一个点x是被一个密度吸引点x^*密度吸引的, 如果存在一组点$x_0, x_1, \cdots, x_k$, $x_0 = x, x_k = x^*$, 对$x_{i+1}(0 < i < k)$的梯度是在x_i的方向上.

由此, 可引出中心定义的簇(center-defined cluster)和任意形状的簇(arbitrary-shape cluster)两个概念. 当x^*的密度函数不小于阈值ξ时, 密度吸引点x^*的中心定义的簇是一个被x^*密度吸引的子集C; 否则, 它被认为是孤立点. 而一个任意形状的簇是子集C的集合. 从一个域到另一个域都存在一条路径P, 该路径上每个点的密度函数值都不小于ξ[1~4].

2.3 几种较新的密度聚类算法分析(Analysis of several newer density-based clustering algorithms)

1) 经典的共享近邻聚类(SNN)算法.

基于时空聚类分析的应用需求, Levent Ertoz等人提出了一个经典的基于共享型最近k邻居的聚类算法, 即SNN(shared neatest neighbor clustering algorithm). 共享近邻聚类算法适合于均匀密度的数据集, 但是对不同密度的数据集进行聚类时精度并不理想.

共享近邻SNN算法的主要思想是: 对于数据集中每个点, 找出距离其最近的k个邻近点, 形成一个集合. 然后考虑数据集中的任意两个点, 若对应于这两个点的k个邻近点集合交集部分的点数超过一个阈值, 则将这两个点归于一类.

2) 基于网格的共享近邻聚类(GNN)算法.

GNN聚类算法的基本思想[7]: 首先将数据空间S划分为网格单元, 将每一维M等分, 并将数据集V映射到网格单元中, 计算出非空的网格单元数GridNum和网格单元中点数的最大值Max_Grid, 再利用密度阈值处理方法算出密度阈值MinPts. 根据密度阈值判断每个网格单元是否为高密度单元, 对高密度单元利用网格中心点技术计算其中心点, 而对低密度单元中的数据点进行边界点提取或作为噪声数据处理, 根据用户输入的共享点数sharedpoints, 近邻数k和聚类阈值Min对高密单元的中心点使用共享近邻算法进行聚类, 并对聚类结果进行检查, 对类中数据点个数小于Min的聚类作为噪声数据处理.

3) CLIQUE算法.

CLIQUE聚类算法综合了基于密度和基于网格的聚类方法, 是一种基于密度网格的聚类方法, 它对于处理大数据库中的高维数据非常有效. 其基本思想如下:

a) 给定一个非均匀分布的大规模多维数据点集, CLIQUE能够识别其中的稀疏和拥挤空间区域, 从而获得数据集的全局分布模式;

b) 如果一个单元中包含的数据点数超过某个输入模型参数, 则该单元是密集的. CLIQUE将相连的

密集单元组合起来实现聚类.

CLIQUE主要包括以下两个步骤:

Step 1　将数据空间划分为互不相交的长方形单元, 记录每个单元里的对象数.

Step 2　用先验性质识别包含簇的子空间.

c) 识别簇. 在符合兴趣度的子空间中先找出密集单元, 再找出相连的密集单元.

d) 为每个簇生成最小化的描述.

CLIQUE采用的先验知识是: 给定一个K维的候选密集单元, 如果检查它的$K-1$维投影空间, 发现任何一个不是密集的, 那么第K维的单元也不可能是密集的. CLIQUE能够自动确定高密度单元所在子空间的最高维数, 并在此子空间内对高密度单元实行聚类[8]. 它具有对数据输入顺序不敏感、可对任意形状高密度单元聚类、数据维数增加时具有良好的可伸缩性等特点. 该算法的不足之处是聚类精度不是很高.

4) LSD算法.

王晓峰等人提出的水平集密度算法(level set density, LSD)[9], 通过分析数据集获得数据集边界信息, 由给定数据集定义网格边界, 然后使用给定的步长构建网格结构. 将数据集映射到网格之后, 网格反映了数据集的分布状况. 为了提高聚类质量, 使用了门限处理方法进行网格分割. 网格聚类从任意非零网格点开始, 以各维步长为单位, 以该网格点为核心点, 不断搜索网格点的邻域, 挖掘其中的直接密度可达对象, 循环这个过程, 直至所有密度可达网格点获得类标记, 完成一个类的聚类. 循环这个过程, 直至网格结构中所有的网格单元都获得类标记.

3　改进的基于密度分布函数的聚类算法 (Improved clustering algorithm based on density distribution function)

针对以上提出的问题, 在改进的基于密度分布函数的聚类算法中引入区域比例的思想[10,11], 即, 借助数据集的区域统计实现聚类. 所谓区域比例指的是将数据集中某个点的k个最近邻点的平均距离与中心点的k个最近邻点的平均距离的比值. 依据区域比例可以找到不同密度的类之间的比例因子, 并由此创建数据集的相似关系(矩阵), 它是自反的、对称的. 在此基础上, 根据传递闭包的概念, 将其变换为等价关系. 然后, 选择适当的比例阈值ratio, 取等价关系的ratio水平集, 根据水平集确定数据点所属类别. 该算法用到以下定义[12~14]:

定义 1　点的密度分布函数[6,15].

给定数据对象p和i, 且i对p的影响函数是$\mathrm{Density}_B(p,i)$. 原则上, 该影响函数是由i和p之间的距离决定的一个任意函数. 距离函数$d(p,i)$应当是自反的和对称的, 例如欧几里德距离函数. 两种典型的影响函数是方波函数(square wave function)和高斯函数(Gause function). 相应地, 定义如下:

$$\mathrm{Density}_{\mathrm{Square}}(p,i)=\begin{cases}0, & d(p,i)=0,\\ 1, & \text{其他},\end{cases} \tag{1}$$

$$\mathrm{Density}_{\mathrm{Gause}}(p,i)=\mathrm{e}^{\frac{-d(p,i)^2}{2\sigma^2}}. \tag{2}$$

在数据对象p上的密度函数被定义为所有数据点的影响函数的和. 给定k个数据对象$D=(x_1,x_2,\cdots,x_k)$, 在p上的密度函数定义如下:

$$\mathrm{Density}_B^D(p)=\sum_{i=1}^{k}\mathrm{Density}_B^i(p). \tag{3}$$

如影响函数采用高斯函数, 则点P的密度分布函数是

$$\mathrm{Density}(p)=\mathrm{Density}_B^D(p)=\sum_{i=1}^{k}\mathrm{e}^{\frac{-d(p,i)^2}{2\sigma^2}}, \tag{4}$$

其中$d(p,i)$为点p和第i个最近邻之间的欧氏距离, 该距离越小, 点的邻域越紧凑, 点的密度越大.

定义 2　核心点.

以点p与其第K个最近邻点之间的距离的某个给定比例ratio, 称为区域比例, 作为半径radius, 若以该点为中心, radius为半径的圆内最近邻的点的个数大于等于某个给定的阈值MinPts, 则该点为核心点. 类从核心点扩展.

定义 3　中心点.

中心点为当前密度最大的核心点. 定义当前密度最大的核心点为中心点p_core.

算法思想　以p_core为中心点向其KNN扩展, 将KNN加入候选队列, 如果候选队列中某点q密度大于中心点密度和密度阈值a的积, 称为半径比例因子, 即

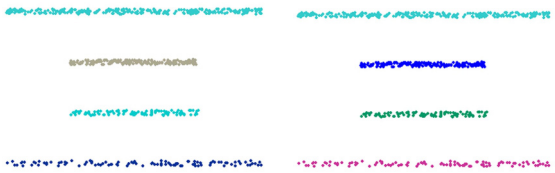
$$\mathrm{density}(p)\geqslant\alpha\times\mathrm{density}(p\_\mathrm{core}). \tag{5}$$

且该点为核心点, 则从该点的K最近邻扩展类, 将该点的KNN加入候选队列, 否则给该点类标号, 从种子队列删除. 当密度下降到中心点密度的给定的比率a时类延伸结束. 该过程被循环直至聚类完成.

4　算法实例(Algorithm examples)

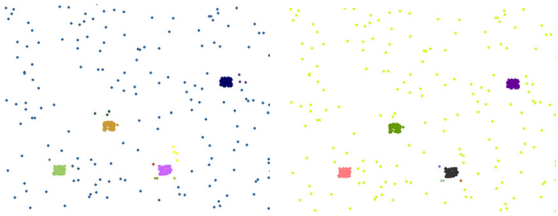
DENCLUE聚类结果对参数非常敏感, 参数微小变化可能导致差别极大的聚类结果[16,17]. 为考察改进的基于密度分布函数的聚类算法的有效性和对参数的敏感性, 笔者进行了大量实验. 本文对参数MinPts, k, a, ratio设置两组不同的值, 分别对3个数据集进行了聚类, 试验结果如图1所示.

从图1可以看出, 本文的算法能有效进行聚类并降低了基于密度聚类方法对参数的敏感性.



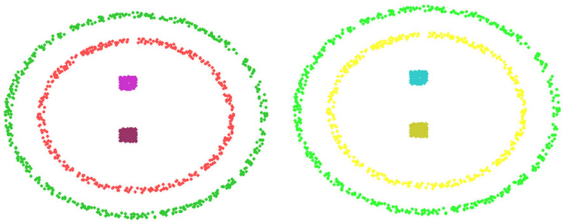
MinPts=4,k =15,a =0.9,ratio=0.1(左)
MinPts=7,k =20,a =0.9,ratio=0.1(右)

(a)



MinPts=4,k =15,a =0.9,ratio=0.1(左)
MinPts=7,k =20,a =0.9,ratio=0.1(右)

(b)



MinPts=4,k =15,a =0.9,ratio=0.1(左)
MinPts=7,k =20,a =0.8,ratio=0.2(右)

(c)

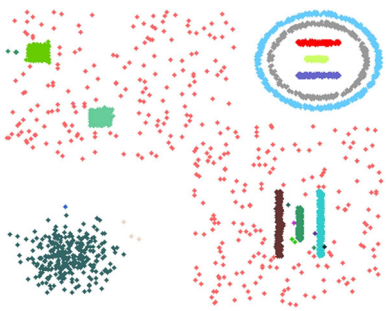
图 1 改进算法聚类结果

Fig. 1 Clustering results of the improved algorithm

5 改进算法与其它算法的对比分析 (Contrast and analysis of the improved algorithm and other algorithms)

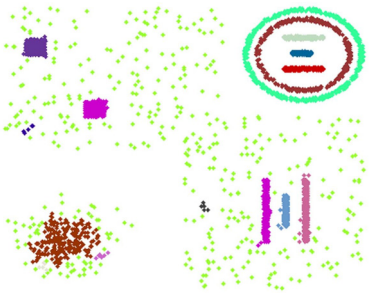
1) 与DENCLUE聚类效果对比分析.

为比较本文的算法与DENCLUE的聚类效果, 采用两种算法对同一数据集进行聚类, 聚类结果如图2所示.



MinPts=4, k =12, a =0.9, ratio=0.1

(a) 改进算法聚类结果



MinPts=4, ε = 0.015

(b) DENCLUE聚类结果

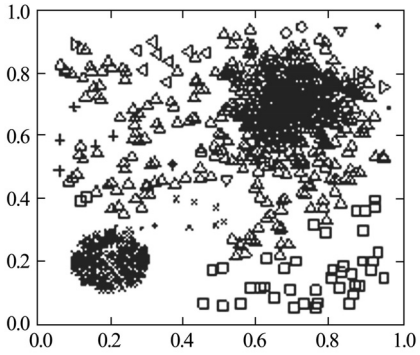
图 2 改进算法与DENCLUE聚类结果之比较

Fig. 2 Comparison of the clustering results of the improved algorithm and DENCLUE

从图2可以看出, DENCLUE在发现密度均匀区域的类时效果较好, 而在发现密度不均匀的类时, 聚类的结果中吸收了附近不少的孤立点, 效果不理想. 本文的算法没有吸收孤立点, 明显改善了这一不足.

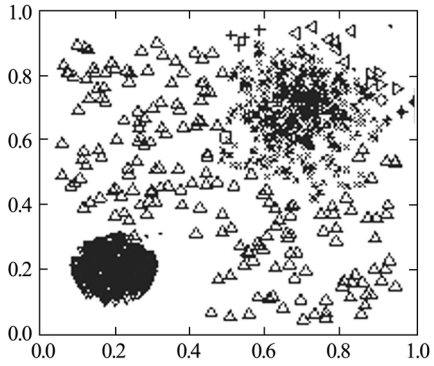
2) 与GNN聚类准确率及效率对比分析.

为了使用改进算法与GNN进行对比试验, 使用了更为复杂的实际数据集D, 数据集D有3个簇, 如图3所示, 图3(a)为改进算法聚类结果, 参数设置为k =4, Minpts = 7, α = 0.9, β = 0.1, 如图所示, 低密度区域被分离为两个簇, 高斯分布的簇与低密度区域的簇被合并, 原因是密度边缘距密度中心太远. 图3(b)所示为改进算法参数设置为k = 4, Minpts = 7, α = 0.9, β = 0.15时的聚类结果, 其中β = 0.15, 即, 收缩了密度边缘. 从图3中可以看出收缩了密度边缘之后, 低密度区域聚类为一个簇, 总体上比较完整的辨识了3个簇. 图3(c)所示为GNN参数设置为密度阈值Minpts= 4, 半径eps = 0.05时的聚类结果, GNN将低密度区域分割为多个区域, 然而从另两个簇的聚类结果来看, 除了离簇中心较远的区域没有识别外, 其他数据点都能很好的聚类. 图3(d)为GNN参数设置为Minpts = 4, eps = 0.03时的聚类结果, GNN只聚类了高密度区域. 结果表明改进算法在处理密度分布不均数据集时比GNN有更好的聚类效果, 并且有很好的接口.



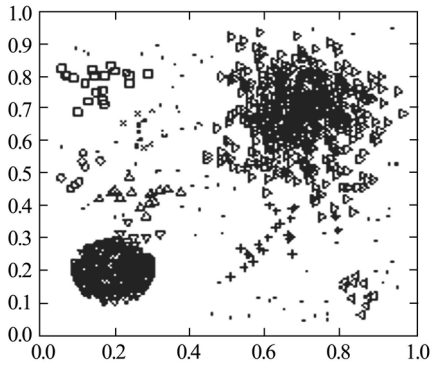
k = 4, Minpts = 7, α = 0.9, β = 0.15

(a) 改进算法聚类结果



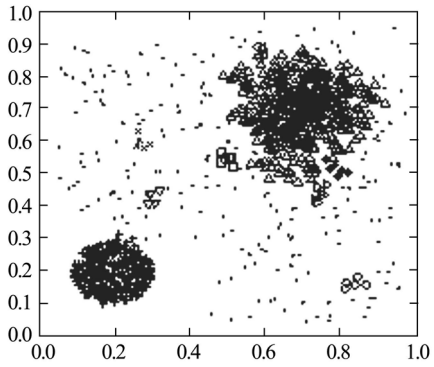
$k=4, \text{Minpts}=7,\ \alpha=0.9,\ \beta=0.1$

(b) 改进算法聚类结果



Minpts = 4, eps = 0.05

(c) GNN聚类结果



Minpts = 4, eps = 0.03

(d) GNN聚类结果

图 3 改进算法与GNN对比试验

Fig. 3 Comparative tests of the improved algorithm with GNN

聚类结果使用准确率来评价, 其定义为

$$\text{准确率} = \frac{\text{正确识别的模式类样本数}}{\text{模式类样本总数}} \times 100\%. \quad (6)$$

改进算法与GNN聚类准确率之比较如表1所示. 从表1可看出, 改进算法聚类准确率比GNN略高.

使用9组均匀分布的数据集对改进算法与GNN进行时间性能对比.

每个数据集分为3类. 测试时将数据规范化在[0, 1], 参数设置为: $K=15$, Minpts = 7, $\alpha=0.2$, $\beta=0.9$, 并使用本算法与GNN算法进行比较. 改进算法和GNN的运行时间如图4所示. 从图4可见, 在处理小数据集时, 改进算法效率不如GNN, 但是在处理大数据集时, 改进算法比GNN的效率高, 原因是改进算法为数据集维护了一个密度列表, 使得改进算法在处理大数据集时有更高的效率.

表 1 改进算法与GNN聚类准确率之比较

Table 1 Contrast of the clustering accuracy of the improved algorithm and GNN

模式类	准确率	
	改进算法/%	GNN/%
◁	97.9	92.3
△	95.9	93.6
□	94.4	91.3

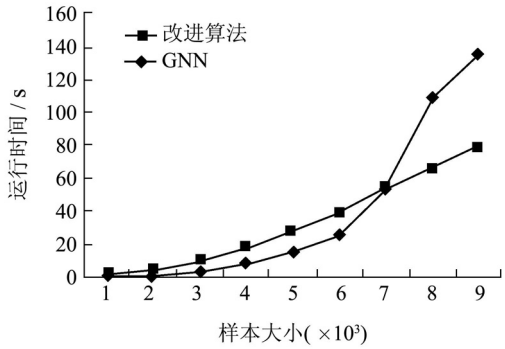


图 4 改进算法和DBSCAN效率之比较

Fig. 4 Comparison of the clustering efficiency of the improved algorithm and GNN

6 结论(Conclusion)

本文利用区域比例和边界阈值的思想构建了一种改进的基于密度分布函数的聚类算法. 其主要特点有: 该算法具有坚实的数学基础, 融合了基于划分的、层次的、密度的、网格的及基于模型的等算法的特点; 对含有大量“噪声”的任意形状高维数据集合, 具有良好的聚类特性; 用当前密度最大点作为中心点, 并从中心点进行类的扩展, 直至密度边界阈值. 本文旨在极大降低基于密度聚类算法对参数的敏感性、改善对高维密度分布不均数据集的聚类效果、提高聚类准确率和效率. 试验结果表明本文算法基本达到了上述目标.

参考文献(References):

[1] 陈燕, 耿国华, 郑建国. 一种改进的基于密度的聚类算法[J]. 微机发展, 2005, 3(15): 12 – 16.
(CHEN Yan, GENG Guohua, ZHENG Jianguo. An improved density-based clustering algorithm[J]. *Microcomputer Development*, 2005, 3(15): 12 – 16.)

[2] 冯少荣, 肖文俊. 一种提高DBSCAN聚类算法质量的新方法[J]. 西安电子科技大学学报(自然科学版), 2008, 35(3): 24 – 27.

(FENG Shaorong, XIAO Wenjun. A new method improving the quality of DBSCAN[J]. *Journal of XIDIAN University(Science and Technology)*, 2008, 35(3): 24 – 27.)

[3] 马帅. 一种基于参考点和密度的快速聚类算法[J]. 软件学报, 2003, 11(6): 34 – 37.
(MA Shuai. An fast clustering algorithm based on reference points and density[J]. *Journal of Software*, 2003, 11(6): 34 – 37.)

[4] 容秋生, 严君彪, 郭国强. 基于DBSCAN聚类算法的研究与实现[J]. 计算机应用, 2004, 4(24): 12 – 16.
(RONG Qiusheng, YAN Junbiao, GUO Guoqiang. Research and realization based on DBSCAN[J]. *Journal of Computer Applications*, 2004, 4(24): 12 – 16.)

[5] 周妍, 孔晓玲, 张然. 数据挖掘中的聚类算法研究[J]. 福建电脑, 2007, (8): 9 – 10.
(ZHOU Yan, KONG Xiaoling, ZHANG Ran. Research of clustering algorithms in data mining[J]. *Fujian Computer*, 2007, (8): 9 – 10.)

[6] 余小高, 余小鹏. 基于距离和密度的无监督算法的研究[J]. 计算机应用与软件, 2010, 27(7): 122 – 125.
(YU Xiaogao, YU Xiaopeng. Research of un-supervisory algorithms based on distance and density[J]. *Computer Applications and Software*, 2010, 27(7): 122 – 125.)

[7] NASIBOV E N, ULUTAGAY G. Robustness of density-based clustering methods with various neighborhood relations[J]. *Fuzzy Sets and Systems*, 2009, 160(24): 3601 – 3615.

[8] AGRAWAL R, GEHRKE J, GUNOPULOS D, et al. Automatic subspace clustering of high dimensional data for data mining applications[C] *//1998 ACM SIGMOD International Conference on Management of Data Seattle*, USA: ACE: 1998, 28(2): 94 – 105.

[9] WANG X F, HUANG D S. A novel density-based clustering framework by using level set method[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2009, 21(11): 1515 – 1531.

[10] BICICI E, YURET D. Locally scaled density based clustering[C] *//The 8th International Conference on Adaptive and Natural Computing Algorithms*. Berlin: Springer-Verlag, 2007: 739 – 748.

[11] ZHOU S G, ZHOU A Y, JIN W. FDBSCAN: a fast DBSCAN algorithm[J]. *Journal of Software*, 2000, 11(6): 735 – 744.

[12] 李飞, 薛彬, 黄亚楼. 初始中心优化的K-Means聚类算法[J]. 计算机科学, 2002, 29(7): 94 – 96.
(LI Fei, XIE Bin, HUANG Yalou. K-Means clustering algorithm based on initial center optimization[J]. *Computer Science*, 2002, 29(7): 94 – 96.)

[13] 周水庚, 周傲英, 曹晶. 基于数据分区的DBSCAN算法[J]. 计算机研究与发展, 2000, 37(10): 1153 – 1159.
(ZHOU Shuigeng, ZHOU Aoying, CAO Jing. DBSCAN based on data partition[J]. *Journal of Computer Research and Development*, 2000, 37(10): 1153 – 1159.)

[14] CHEN N, CHEN A, ZHOU L X. An incremental grid density based clustering algorithm[J]. *Journal of Software*, 2002, 13(1): 1 – 7.

[15] 李清峰, 周鲜成, 王莉等. 一种不精确数据的聚类挖掘算法[J]. 计算机应用研究, 2009, 26(3): 887 – 889.
(LI Qingfeng, ZHOU Xiancheng, WANG Li, et al. A clutering mining algorithm of unexact data[J]. *Application Research of Computers*, 2009, 26(3): 887 – 889.)

[16] 张莉, 周伟达, 焦李成. 核聚类算法[J]. 计算机科学, 2002, 25(6): 587 – 590.
(ZHANG Li, ZHOU Weida, JIAO Lichen. Nuclear clustering algorithm[J]. *Computer Science*, 2002, 25(6): 587 – 590.)

[17] 张伟, 廖晓峰, 吴中福. 一种基于遗传算法的聚类新方法[J]. 计算机科学, 2002, 29(6): 114 – 116.
(ZHANG Wei, LIAO Xiaofeng, WU Zhongfu. A new clustering method based on genetic algorithms[J]. *Computer Science*, 2002, 25(6): 587 – 590.)

作者简介:

谭建豪 (1962—), 男, 教授, 博士, 目前研究方向为人工智能和数据挖掘, E-mail: tanjianhao96@sina.com.cn;

章 兢 (1957—), 男, 教授, 博士生导师, 目前研究方向为复杂系统的计算机控制, E-mail: zhangj@hnu.cn;

李伟雄 (1985—), 男, 硕士研究生, 目前研究方向为人工智能和数据挖掘, E-mail: lwx1633@yahoo.com.cn.