ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.7, July 2008, pp.1683−1692
DOI: 10.3724/SP.J.1001.2008.01683

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于*K*-Means局部最优性的高效聚类算法*

雷小锋[1,2+], 谢昆青[1], 林 帆[1], 夏征义[3]

[1](北京大学 信息科学技术学院智能科学系/视觉与听觉国家重点实验室,北京 100871)

[2](中国矿业大学 计算机学院,江苏 徐州 221116)

[3](中国人民解放军总后勤部 后勤科学研究所,北京 100071)

An Efficient Clustering Algorithm Based on Local Optimality of *K*-Means

LEI Xiao-Feng[1,2+], XIE Kun-Qing[1], LIN Fan[1], XIA Zheng-Yi[3]

[1](Department of Intelligence Science/National Laboratory on Machine Perception, Peking University, Beijing 100871, China)

[2](School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

[3](Logistics Science and Technology Institute, P.L.A. Chief Logistics Department, Beijing 100071, China)

+ Corresponding author: E-mail: leiyunhui@gmail.com

Lei XF, Xie KQ, Lin F, Xia ZY. An efficient clustering algorithm based on local optimality of *K*-Means. *Journal of Software*, 2008,19(7):1683−1692. http://www.jos.org.cn/1000-9825/19/1683.htm

Abstract: *K*-Means is the most popular clustering algorithm with the convergence to one of numerous local minima, which results in much sensitivity to initial representatives. Many researches are made to overcome the sensitivity of *K*-Means algorithm. However, this paper proposes a novel clustering algorithm called *K*-MeanSCAN by means of the local optimality and sensitivity of *K*-Means. The core idea is to build the connectivity between sub-clusters based on the multiple clustering results of *K*-Means, where these clustering results are distinct because of local optimality and sensitivity of *K*-Means. Then a weighted connected graph of the sub-clusters is constructed using the connectivity, and the sub-clusters are merged by the graph search algorithm. Theoretic analysis and experimental demonstrations show that *K*-MeanSCAN outperforms existing algorithms in clustering quality and efficiency.

Key words: *K*-MeanSCAN; density-based; *K*-Means; clustering; connectivity

摘 要: *K*-Means聚类算法只能保证收敛到局部最优,从而导致聚类结果对初始代表点的选择非常敏感.许多研究工作都着力于降低这种敏感性.然而,*K*-Means 的局部最优和结果敏感性却构成了 *K*-MeanSCAN 聚类算法的基础.*K*-MeanSCAN 算法对数据集进行多次采样和 *K*-Means 预聚类以产生多组不同的聚类结果,来自不同聚类结果的子簇之间必然会存在交集.算法的核心思想是,利用这些交集构造出关于子簇的加权连通图,并根据连通性合并子簇.理论和实验证明,*K*-MeanScan 算法可以在很大程度上提高聚类结果的质量和算法的效率.

关键词: *K*-MeanSCAN;基于密度;*K*-Means;聚类;连通性

中图法分类号: TP18 **文献标识码**: A

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2006AA12Z217 (国家高技术研究发展计划(863)); the Foundation of China University of Mining and Technology under Grant No.OD080313 (中国矿业大学科技基金)

Received 2006-10-09; Accepted 2007-07-17

聚类分析在统计学、机器学习、数据挖掘、生物学、空间数据库、Web 搜索、分布式网络、市场营销等领域得到广泛的研究和应用,在文献[1]中对聚类方法进行了很好的综述.一般地,假设 d 维数据空间 A 中有 n 个样本,每个样本可以视为 d 维空间的一个点,聚类是将这 n 个点分组,每组就形成一个类簇,要求属于同一类簇的点尽可能相近,不同类簇间的点尽可能远离.

常用的聚类算法主要有划分方法、层次方法、基于密度的方法、基于网格的方法和基于模型的方法.本文提出一种新的聚类算法——*K*-MeanSCAN,集成多种聚类方法的性质,特别是 *K*-Means 算法的局部最优特征和结果敏感性.理论分析和实验结果表明,*K*-MeanSCAN 算法具有如下特点:

(1) 可以发现任意形状的类簇.*K*-MeanSCAN 算法充分借鉴了基于密度聚类方法的思想,能够发现任意形状的类簇,并对异常点和噪声数据不敏感.

(2) 考虑了子簇之间的连通强度.避免了基于密度聚类算法中由于密度连通关系的传递性导致绝大多数的样本点聚集到非常少的几个类簇中(通常是一类).

(3) 算法具有很好的时间性能和伸缩性.算法结合采样技术、*K*-Means 预聚类技术和图搜索算法提供高效的聚类性能,其时间复杂度为 $O(n)$,且对大数据集有很强的伸缩性.

(4) 算法引入了钝参数的概念,减轻了聚类结果对参数阈值的敏感性,使得算法无需过多的先验知识.

此外,算法仅使用一个参数对类簇的密度进行建模,与DBSCAN(density based spatial clustering of application with noise)[2]的Eps和MinPts两个参数相比,易于交互式或自动计算以确定密度阈值.

本文第 1 节对现有的一些聚类算法进行综述.第 2 节详细阐述 *K*-MeanSCAN 算法的基本思想及其形式化定义,并给出具体的算法描述和复杂性分析.第 3 节提供详细的实验以证明 *K*-MeanSCAN 算法的聚类质量和执行效率.最后是结论和下一步工作说明.

1 常用聚类算法综述

划分聚类算法通过迭代重定位策略优化特定的目标函数,尝试确定数据集的一个划分.最常用的目标函数是误差平方和准则,如 *K*-Means 算法.*K*-Means 算法对类球形且大小差别不大的类簇有很好的表现,但不能发现形状任意和大小差别很大的类簇,且聚类结果易受噪声数据影响.此外,*K*-Means 算法仅保证快速收敛到局部最优结果,从而导致聚类结果对初始代表点的选择非常敏感,本文称其为结果敏感性,是许多研究工作所要着力解决的问题.

层次聚类算法以自顶向下(分裂)或自底向上(凝聚)的方式将数据对象划分成一个层次树结构,即类簇树.算法的聚类效果很大程度上依赖于度量类簇之间相异度的距离函数.此外,一般层次聚类算法的伸缩性不强,其时间复杂度通常为$O(n^2)$.BIRCH(balanced iterative reducing and clustering using hierarchies)[3]和CURE(clustering using representatives)[4]算法试图提高层次聚类结果的质量,解决其算法伸缩性问题.BIRCH算法具有$O(n)$的计算复杂度,在有限内存下可以很好地工作.但是,算法使用的相异度度量导致BIRCH只能发现球形类簇.CURE使用固定数量的代表点来定义类簇,可以发现复杂形状和不同大小的类簇,对噪声具有很好的免疫能力.然而,CURE算法的收缩方式隐含地依赖于球形类簇假设,故在处理特殊形状的类簇时比较困难.

基于密度的聚类算法中类簇被定义为连通的稠密子区域.因此,算法能够发现任意形状的类簇,并对异常点和噪声有自然的免疫能力.DBSCAN 是典型的面向低维空间数据聚类的基于密度的算法,其关键概念是由对象最近邻域的局部分布度量的密度及其连通性.DBSCAN 算法本质上只是提供了一个根据密度阈值参数进行聚类结果搜索的过程,聚类结果在用户指定密度阈值那一刻已经唯一地确定,算法本身并不对聚类的结果负责.其主要缺陷包括:

(1) 算法通常对参数值的设置非常敏感,聚类结果的质量很大程度上依赖于密度阈值参数的合理选取,因而对用户的经验和专业素养提出很高的要求;

(2) 真实的数据集合经常分布不均匀,导致这种全局性的密度参数通常不能刻画数据内在的聚类结构;

(3) 算法中通过密度连通性来合并高密子簇,由于密度连通关系的传递性,往往使得绝大多数的样本点

聚集到非常少的几个类簇中(通常是一类).

OPTICS(ordering points to identify the clustering structure)[5]算法针对DBSCAN算法的缺陷进行了改进.

2 K-MeanSCAN 算法

2.1 算法思想

2.1.1 从搜索到构造

许多聚类算法的基本框架是搜索与合并.如在层次方法中需要搜索两个距离最近的类簇然后合并;而基于密度的聚类算法则不断地搜索高密子区域,然后利用连通性将其合并到当前聚类结果中.很明显,搜索过程需要面对整个样本集合,通常会导致算法低效.如DBSCAN需要测试每个对象是否是核心对象,并对每个核心对象搜索其直接密度可达的对象,如果没有空间索引的辅助,DBSCAN算法的复杂度为$O(n^2)$.实际上,现有的很多聚类算法已经关注到这个问题,如CURE算法利用采样方法减小搜索空间,而Chameleon[6]算法则通过图划分算法将样本对象聚类为大量相对较小的子簇.具体到本文,我们采用了随机采样和*K*-Means算法高效地聚出大量的高密子簇,后续处理都基于这些构造出的高密子簇进行,无须直接面对所有的样本.我们称这种算法框架为构造与合并.因此,*K*-MeanSCAN算法的处理流程为:随机采样、预聚类、合并和后处理.聚类算法的本质是密度估计问题,*K*-MeanSCAN算法的核心思想是,增加*K*-Means算法中高斯混合模型的高斯分量数目以提升密度估计的精度,并利用*K*-Means聚类的局部最优和结果敏感性的特征进行高斯分量(即高密子簇)的合并剪枝,克服过拟合问题,最终实现有效的聚类.

2.1.2 *K*-Means 算法的概率模型

K-Means 算法实质上是一种将聚类视为密度估计问题的概率方法.在概率方法中,假设样本来自于如下形式的混合模型:

$$p(\boldsymbol{x}\mid\Theta)=\sum_{j=1:k}P(C_j)p_j(\boldsymbol{x}\mid\theta_j,C_j),$$

式中,$\Theta=(\theta_1,\ldots,\theta_k)$是待估计的参数向量;条件概率密度$p(\boldsymbol{x}|\theta_j,C_j)$称为分量密度,表示类别$j$的概率密度形式,且参数向量$\theta_j$未知;先验概率$P(C_j)$称为混合因子.为了简化问题,*K*-Means算法进一步假设:(1) 每个类别的概率密度形式为球形高斯分布,即$\theta_j=(\mu_j,\Sigma)$且$\Sigma_1=\ldots=\Sigma_k=\sigma^2I,\mu_j,\sigma^2$未知;(2) 每个样本唯一地属于一个类别;(3) 假设所有类别的混合因子相等.于是,混合模型简化为

$$p(\boldsymbol{x}\mid\Theta)=\max_{j=1:k}\phi(\boldsymbol{x}\mid\mu_j,\Sigma,C_j)=\max_{j=1:k}\left\{\frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x}-\mu_j\|^2\right)\right\} \tag{1}$$

该简化模型可以通过最大似然方法求解,对于观测样本$X=(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$,相应的对数似然函数为

$$l(\Theta\mid X)=\sum_{i=1}^{n}\ln p(\boldsymbol{x}_i\mid\Theta)=-n\ln[(2\pi\sigma^2)^{d/2}]-\frac{1}{2\sigma^2}\sum_{j=1}^{k}\sum_{\boldsymbol{x}_i\in C_j}\|\boldsymbol{x}_i-\mu_j\|^2 \tag{2}$$

最大化该对数似然函数等价于最小化上式的欧氏距离平方项,即得到 *K*-Means 的误差平方和准则:

$$J_e=\sum_{j=1}^{k}\sum_{\boldsymbol{x}_i\in C_j}\|\boldsymbol{x}_i-\mu_j\|^2 \tag{3}$$

通过迭代优化上述的误差平方和准则,*K*-Means算法最终可以估计出每个高斯分量的均值向量和协方差矩阵$\Sigma=\sigma^2I$.式中,n_j是类簇C_j的样本数目.

$$\mu_j=\frac{1}{n_j}\sum_{\boldsymbol{x}_i\in C_j}\boldsymbol{x}_i,\ \sigma^2=\frac{1}{nd}\sum_{j=1}^{k}\sum_{\boldsymbol{x}_i\in C_j}\|\boldsymbol{x}_i-\mu_j\|^2=\frac{J_e}{nd} \tag{4}$$

2.1.3 *K*-MeanSCAN 的过拟合-合并剪枝策略

要获得更有效的样本密度估计,J_e的值自然是越小越好.但是,J_e的值不仅取决于样本的分类情况,而且与类别数目k有关.当类别数目k给定时,J_e的值由样本的分类情况所决定,且存在一个最小值$J_{k\text{-min}}$对应于最优的样本类别划分.如果类别数目k和高斯混合模型假设与实际问题相匹配时,最小值$J_{k\text{-min}}$必定很小,从而可以很好地近似样本密度;而如果模型假设不合理,则最小值$J_{k\text{-min}}$可能依然很大,对样本分布的近似效果较差.

对于任意形状的类簇,很明显不能直接要求数据分布满足高斯混合模型的假设,否则会导致最小的误差平方和$J_{k\text{-min}}$很大.实际上,高斯混合模型具有很强的表达能力,如果高斯分量密度的数目k足够大,则高斯混合模型几乎可以近似任意一种概率分布.换言之,随着类别数目k的增加,相应的$J_{k\text{-min}}$会减小.简单证明如下:

类别数目k的增加,必然导致最终每个类簇的形状缩小,对应于高斯分量的σ^2减小;而从公式(3)和公式(4)可以得出$J_{k\text{-min}}=nd\sigma^2$,即最小的误差平方和正比于$\sigma^2$;因此,$k$的增加会导致最小的误差平方和$J_{k\text{-min}}$的减小.

极端情况下,$k=n$,则每个样本点都是一个类簇,即$J_{k\text{-min}}=0$,说明此时的经验误差为 0,但是此时,模型的推广能力极差.根据统计学习理论,经验误差最小并不等于期望误差最小,经验风险只有在样本数无穷大时才趋近于期望风险.因此,经验误差最小不能保证分类器的推广能力,需要找到经验风险最小和推广能力最大的平衡点.同样,利用足够多的高斯分量组成的混合模型来描述数据会导致过拟合的问题,影响模型的推广能力.因此,在*K*-MeanSCAN算法中,我们采用过拟合-剪枝的策略进行聚类,即首先使用分量足够多的高斯混合模型来较好地近似样本分布,然后通过合并一些高斯分量的剪枝策略来处理过拟合问题.

除了提高对样本分布的近似精度以外,增加类别数目还有其他好处.已知 *K*-Means 算法通过迭代重定位技术最小化误差平方和准则,最终快速收敛到一个局部最优的解.形象地,可以认为 *K*-Means 算法是在对数似然函数空间上的随机爬山算法,对于具有 k 个全局最高点和大量局部极高点的数据集,*K*-Means 聚类算法很可能陷入局部极高点,从而导致对于不同的初始类簇代表点,算法会收敛到不同的局部极值点,即聚类结果对初始代表点的选择非常敏感,本文称其为"结果敏感性".类别数目 k 的增大类似于开辟更多的爬山路径,如果 $k'>>k$ 个路径,则到达全局最高点的可能性自然会增大,当然过拟合问题不可避免.

2.1.4 类簇的合并剪枝

在*K*-MeanSCAN算法中,利用*K*-Means的结果敏感性对类簇进行合并剪枝来处理过拟合问题.假设在样本集上进行两次*K*-Means聚类,得到两组结果分别是$R_1=\{C_{11},C_{12},\ldots,C_{1k'}\}$和$R_2=\{C_{21},C_{22},\ldots,C_{2k'}\}$,且因结果敏感性使得$R_1\neq R_2$,则至少存在一个属于$R_1$(或$R_2$)的类簇$C_{1m}$与$R_2$(或$R_1$)中至少两个类簇之间都存在交集(证明略).

实际中,*K*-Means 算法的结果敏感性导致不同聚类结果中大量的类簇之间存在交集.如图 1 所示,对于图 1(a)的样本集合,图 1(b)、图 1(c)分别是两次 *K*-Means 聚类的结果,图 1(c)说明两次聚类结果之间存在交集.如果不同聚类结果中两个类簇之间交集的基数足够大,则我们宁愿相信这两个类簇中的样本不应该分为两类,而应合并为一个类簇.通过这种方式,可以建立如图 2 所示的类簇间的连通关系,并根据连通关系将类簇合并.

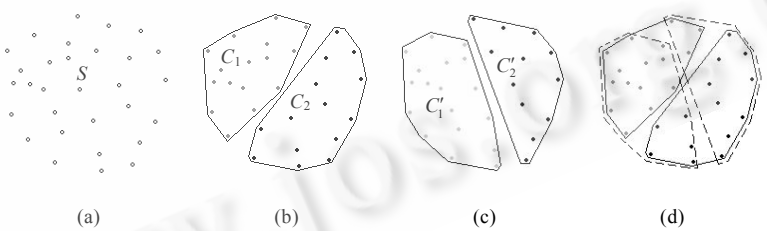


Fig.1 A demo of the sensitivity of *K*-Means algorithm

图 1 *K*-Means 算法结果敏感性的示意

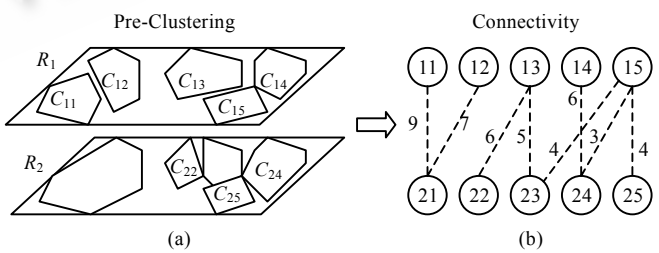


Fig.2 Building the connection relationships between clusters

图 2 建立类簇之间的连通关系

上述思路构成了 K-MeanSCAN 算法实现类簇合并剪枝的基础.基于 K-Means 算法预聚类的结果,K-MeanSCAN 算法的主要步骤包括:(1) 在多次预聚类结果上构造类簇之间的加权连通图;(2) 利用图搜索算法合并连通的类簇;(3) 算法最后还包括一个后处理阶段,利用凝聚层次聚类算法对步骤(2)生成的合并类簇再进行一次聚类,以进一步保证聚类结果的质量.此外,后处理步骤还需要对所有在合并剪枝中剔除的样本进行重新计算,确定其最终的类别标记或者标记为噪声.

在描述具体算法之前,我们首先引入一些形式化的概念和定义.以下用 k 表示实际的类簇数目,用 k'表示预聚类的类簇数目.

2.2 概念和定义

设$P=(R_1,R_2,\ldots,R_h)$表示h次K-Means聚类结果,且$R_i=\{C_{i1},C_{i2},\ldots,C_{ik'}\}\in P$,并称$C_{ij}$为聚类结果$R_i$的一个子簇.

定义 1(子簇密度和高密子簇). 假设C_{ij}为一个子簇,其密度应正比于子簇中的样本数量,反比于子簇的空间范围,形式化定义如下:

$$den(C_{ij})=\|C_{ij}\|/r^d,\ r=\max_{x_l\in C_{ij}}(dist(\boldsymbol{x}_l,C_{ij}.mean)).$$

式中,$\|*\|$表示集合*的基数,r为所有样本到子簇均值中心的最大距离.如果$den(C_{ij})$大于给定阈值$denh$,则称该子簇为高密子簇.在K-MeanSCAN算法中,只有高密子簇才参与合并.

定义 2(直接连通). 假设$C_{ij}\in R_i$和$C_{st}\in R_s$是两个高密子簇,且二者的交集非空,则称C_{ij}与C_{st}是直接连通的,并定义C_{ij}与C_{st}之间的直接连通强度为两个子簇交集的基数与其并集基数的比值,

$$dcn(C_{ij},C_{st})=\|C_{ij},\cap C_{st}\|/\|C_{ij},\cup C_{st}\|.$$

直接连通关系具有自反、对称和传递性,是一种等价关系.可以看到,来自同一聚类结果的任意两个子簇之间都不存在交集,不可能直接连通或者连通强度为 0.如果两个直接连通的子簇的直接连通强度大于给定的阈值 $dcnh$,则称两个子簇是强直接连通的.

定义 3(间接连通). 假设C_i和C_j是任意两个高密子簇,且二者的交集为空,如果存在一个子簇的序列$C_1,C_2,\ldots,C_s$,且$C_1=C_i,C_s=C_j$,使得C_{t+1}与C_t之间是强直接连通的,则称C_i与C_j是强间接连通的,并且间接连通的连通强度为所有强直接连通强度的最小值.

定义 4(合并簇). 基于上述的连通关系,可以构造关于所有高密子簇的加权连通图(weighted connected graph,简称WCG),其中,一个高密子簇对应于图中的一个顶点,并且为每条边分配一个连通强度作为权值.于是,一个合并簇C就定义为所有强连通(包括直接连通和间接连通)的子簇的并集,最终产生的合并簇满足两个条件:(1) C中任意两个子簇必定是强连通的;(2) 如果C_i属于C,且与C_j是强连通的,则C_j也必定属于C.

2.3 算法描述

K-MeanSCAN 算法在构造与合并的算法框架下实现.在构造过程中,算法首先从样本数据集中随机抽取一组样本.然后,算法利用 K-Means 对随机样本进行预聚类,预聚类的类簇数目为 $k'>>k$.重复采样和预聚类过程 h 次,得到 h 组在随机样本上的聚类结果,每组聚类结果包括 k'个类簇中心.此时,对整个样本集进行单遍扫描,将每个样本对象分配给相应的距离最近的 h 个类簇中心.至此,产生 h 组关于数据集的划分.

基于这些数据集的划分,K-MeanSCAN算法计算来自不同划分的子簇之间的连通性,并构造加权连通图,然后利用广度优先的图搜索算法求得最大连通的子图,从而将相应的高密子簇合并,即上文定义的合并簇.最后,利用凝聚的层次聚类算法对合并簇进一步处理,得到最终聚类结构.在层次聚类过程中使用的距离度量定义为两个待合并簇中所有子簇中心的最小距离,假设MC_1和MC_2是两个合并簇,则

$$dist(MC_1,MC_2)=\min_{C_1\in MC_1,C_2\in MC_2}dist(C_1.mean,C_2.mean).$$

最后,根据最终的聚类结构对所有未参与合并剪枝的样本进行重新标记,确定其类别或者标记为噪声.以下通过伪代码给出 K-MeanSCAN 算法的流程说明.

```
Algorithm program K-MeanSCAN(SetOfPoints,h=2,k')
    var  RandomSample: rndsmp;
```

```
    Clusters: cluss, mergedcluss, finalcluss;
    Cluster: clus; Connection: conns;
    Integer: iterations; Float: denh, dcnh;
begin
    //pre-clustering
    repeat
        rndsmp=DrawSample();
        clus=K-Means(rndsmp,k');
        cluss.add(clus); iterations++;
    until iterations=h
    AssignEachObject(SetOfPoints,cluss);
    denh=ComputeDenh(cluss);
    conns=BuildWCG(cluss);
    dcnh=ComputeDcnh(conns);
    mergedcluss=MergedByBFS(cluss,conns);
    finalcluss=MergedByAgglom(mergedcluss);
    return finalcluss;
end. //K-MeanSCAN
```

算法中,参数 h 是采样和预聚类的次数,也是构造过程产生的划分数.参数 k' 是 K-Means 预聚类时的类簇数目.在 K-MeanSCAN 算法中,局部变量 *denh* 和 *dcnh* 分别是密度阈值和连通强度阈值.函数 *AssignEachObject*()在样本集上执行单遍扫描,将每个样本对象分配给相应的子簇;函数 *BuildWCG*()计算子簇之间的连通关系,并构造加权连通图;函数 *MergedByBFS*()通过广度优先的图搜索算法合并子簇;最后,函数 *MergedByAgglom*()通过层次聚类算法产生最终的聚类结构,并对被剔除的样本重新标记.限于篇幅,本文略去关于这些函数更为详细的描述.

2.4 算法参数设置

2.4.1 采样大小

在文献[4]中已经阐明合理的随机采样大小能够相当精确地反映类簇的几何结构,并通过如下定理给出了相应的采样精度保证.假设对于类簇 u,如果样本集包含至少 $f|u|$ 个来自类簇 u 的样本点,$0\le f\le 1$,则在聚类中丢失该类簇的概率很小.

定理 1. 对于一个类簇u,如果采样大小s满足如下约束:则随机样本集中来自类簇u的样本数小于$f|u|$的概率小于δ,$0\le\delta\le 1$.

$$s \ge fn + \frac{n}{|u|}\log\left(\frac{1}{\delta}\right) + \frac{n}{|u|}\sqrt{\left(\log\left(\frac{1}{\delta}\right)\right)^2 + 2f|u|\log\left(\frac{1}{\delta}\right)}.$$

假设$u_{\min}$是需要关注的最小类簇,根据定理 1 可以得到相应的$s_{\min}$.对于k个类簇,在样本大小为$s_{\min}$时,从任意类簇u中采样数目小于$f|u|$的概率具有上界$k\delta$.

2.4.2 预聚类次数 h 和类簇数 k'

参数 h 是预聚类或采样的次数,很明显,$h\ge 2$.根据 K-MeanSCAN 算法的原理,h 较大,则属于同一类簇的子簇之间被合并的几率就较大,从而可以产生更高质量的聚类结果,不过同时也会增加计算代价.本文的实验测试表明,h=3~10 通常已经足够.

预聚类的另一个重要参数是类簇数目k'.对于该参数的约束是要求k'远远大于实际的类簇数目k,小于样本集的大小n.为了能够捕捉到感兴趣的最小类簇$u_{\min}$,k'必须大于$\lceil n/|u_{\min}|\rceil$.在文献[7]中研究了$k$近邻分类算法中参

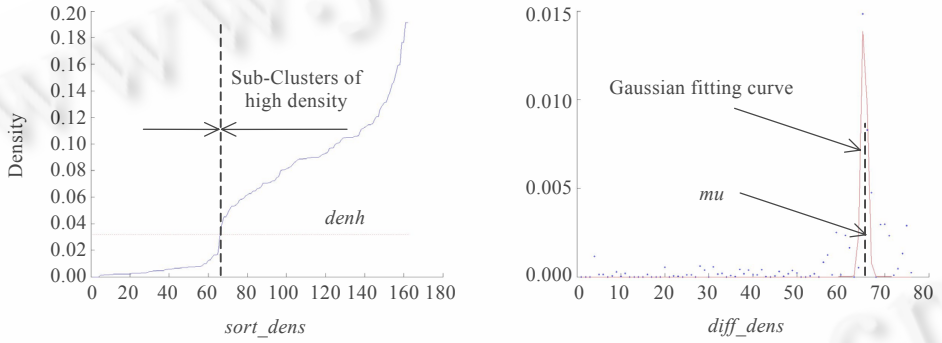
数k的设置,给出一个法则:$k\approx n^{3/8}$.参考这一结论,我们推荐k'值的选取满足如下约束:$\min(\alpha\lceil n/|u_{\min}|\rceil, n^{5/8})$,$\alpha$取值一般小于等于 10.需要说明的是,在满足上述约束条件时,k'的取值对于聚类结果不敏感,我们称这种对结果不敏感的参数为“钝参数”,可以降低对先验知识和用户专业素养的要求,下文的实验将对这一事实进行说明.

2.4.3 子簇密度阈值 *denh* 和连通强度阈值 *dcnh*

参数*denh*是子簇密度阈值,只有密度大于*denh*的子簇才参与*K*-MeanSCAN算法的合并过程,而密度小于*denh*的子簇必包含大量的噪声.直观地,对于包含噪声或完全由噪声样本形成的子簇,其样本数目少,而空间分布范围却较大,故其密度很小;而对实际存在的类簇,则子簇比较紧缩,密度较大.假设h次预聚类形成的hk'个子簇的密度集合为$dens=\{den_i|i=1,\ldots,hk'\}$,以升序排序得到*sort_dens*,其曲线模式如图 3(a)所示.可以看出,从低密区域到高密区域存在一个急剧的跃迁,即可确定密度阈值.

为了自动求解密度阈值,首先对 *sort_dens* 进行差分运算得到 *diff_dens*.绘制差分曲线,与密度曲线的跃迁处相对应,在差分曲线上存在一个较大的峰值.取前一半的差分运算值,并利用高斯函数形式对其进行曲线拟合,差分曲线的峰值处就对应于高斯函数的均值 *mu*,是密度阈值在 *sort_dens* 中对应的索引值,如图 3(b)所示.于是,密度阈值为 $denh=sort\_dens[round(mu)]$.

实验测试表明,这种自动密度阈值计算方法效果很好.而连通强度阈值 *dcnh* 的设置取决于实际用户的主观倾向.一般情况下,可直接设置 *dcnh*=0,或者取所有的连通强度的均值.



(a) Pattern of *sort_dens* (b) Finding *denh* on *diff_dens*

(a) 子簇密度变化的曲线模式 (b) 差分曲线上的密度阈值求解

Fig.3 Density threshold *denh*

图 3 密度阈值 *denh*

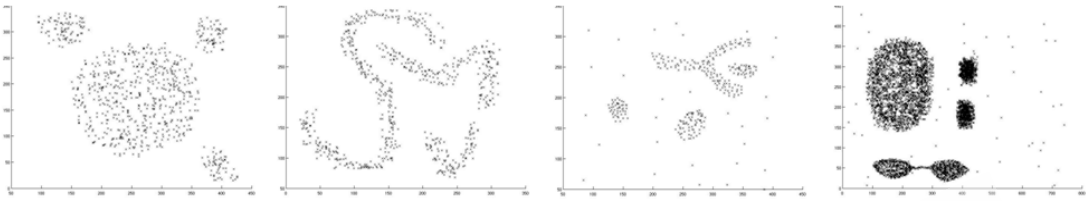
2.5 算法复杂性分析

K-Means算法是相当高效的,其计算复杂度为$O(nkdt)$,其中,n为所有样本对象的数目,k是设定的类簇数目,d是样本数据的维度,t是迭代的次数.通常,$k<<n$且$t<<n$.h次预聚类的时间复杂度为$O(hn_{smp}kdt)$,这里,n_{smp}为随机样本的大小.单遍扫描样本集需要的时间代价为$O(hn)$.此后,所有的运算都基于*K*-Means的预聚类结果,其中不涉及任何代价较高的空间几何计算.函数*BuildWCG*()需要$(hk)^2$次集合求交运算,假定每个子簇的平均样本数目为n/k,则整个集合求交运算的时间代价为$O(h^2kn)$.函数*MergedByBFS*()的时间复杂度为$O(k^2)$,即广度优先搜索的代价.最后,函数*MergedByAgglom*()的时间代价为$O(k^2)$.于是,*K*-MeanSCAN算法的总代价为$O(hkn_{smp}dt)+O(hn)+O(h^2kn)+O(k^2)+O(k^2)$,假定$n$很大,$h$为常数,$k<<n$,则算法的时间复杂度约为$O(n)$.

3 实验评估

本节我们通过实验比较来研究*K*-MeanSCAN算法的性能和有效性.实验采用的数据包括在各种聚类算法评估中常用的模拟数据(如图 4 所示),以及DBSCAN算法中使用的来自真实地球科学领域的SEQUOIA 2000 基准测试数据[8].实验的硬件环境为P4 1.86Ghz的CPU和 512MB的内存,软件环境为Microsoft Windows XP

(professional)操作系统,所有代码均用Visual C++ 6.0 实现.



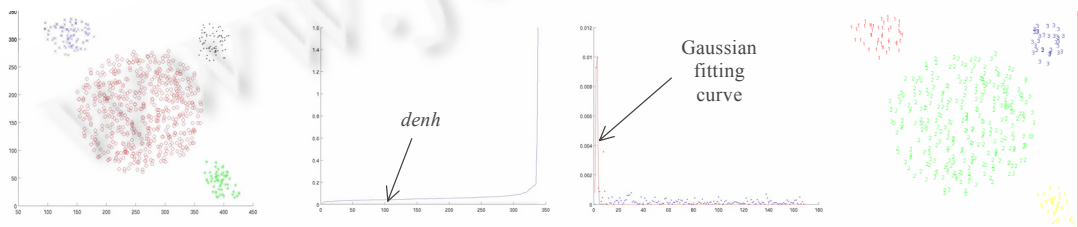
(a) Data 1 (b) Data 2 (c) Data 3 (d) Data 4

Fig.4 Synthetic datasets for evaluating clustering algorithms

图 4 常用的评估聚类算法的模拟数据

3.1 聚类结果

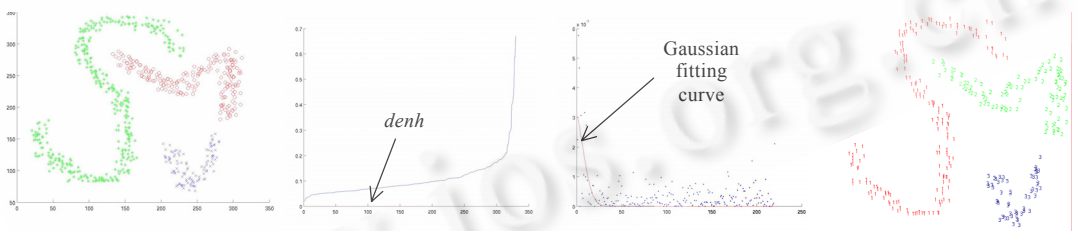
图 5~图 8 分别给出了上述 4 组数据经过预聚类、合并剪枝后的结果,以及相应的DBSCAN聚类结果.实验中,预聚类的类簇数目均设置为$k'\approx n^{5/8}$,预聚类次数h=5,连通强度阈值$dcnh$=0.在每个图中,图(a)是K-MeanSCAN聚类结果,图(b)是相应的密度曲线和密度阈值,图(c)是密度差分序列和高斯函数拟合曲线,图(d)是DBSCAN聚类结果.



(a) (b) *sort_dens* (c) *diff_dens* (d) *Eps*=21, Min*Pts*=4

Fig.5 Clustering result of data 1 and its threshold *denh*

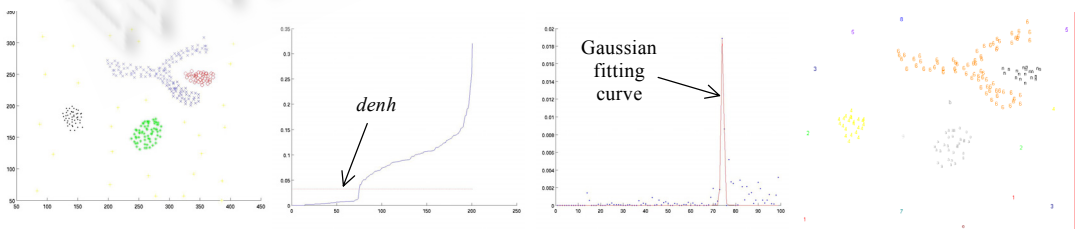
图 5 Data 1 聚类结果及其密度阈值



(a) (b) *sort_dens* (c) *diff_dens* (d) *Eps*=18.64, Min*Pts*=4

Fig.6 Clustering result of data 2 and its threshold *denh*

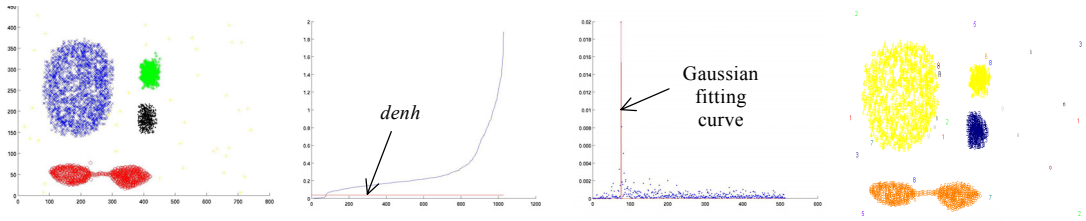
图 6 Data 2 聚类结果及其密度阈值



(a) (b) *sort_dens* (c) *diff_dens* (d) *Eps*=15.38, Min*Pts*=4

Fig.7 Clustering result of data 3 and its threshold *denh*

图 7 Data 3 聚类结果及其密度阈值



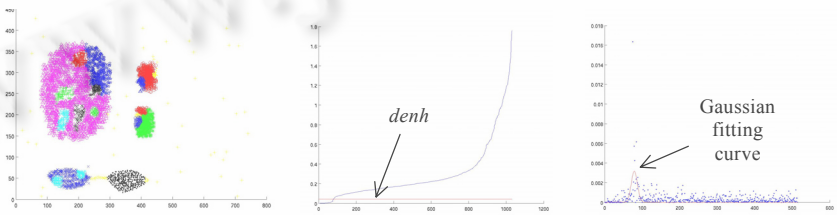
(a) (b) *sort_dens* (c) *diff_dens* (d) *Eps*=10.58, Min*Pts*=4

Fig.8 Clustering result of data 4 and its threshold *denh* (*dcnh*=0)

图 8 Data 4 聚类结果及其密度阈值(*dcnh*=0)

可以看出,对于上述 4 组数据,*K*-MeanSCAN 算法无需后处理步骤即可得到高质量的聚类结果.对于数据集 Data 4 而言,当将其连通阈值设置为所有连通强度的均值时,无后处理步骤的 *K*-MeanSCAN 聚类结果如图 9 所示.此时,下方的两个子簇得到区分,经后处理过程即可得到高质量的聚类效果.此外,DBSCAN 算法亦可取得不错的结果,但其密度参数设置需要多次调试,且对于 Data 4 而言,无法区分下方的两个子簇.

同时可以看出密度曲线的变化模式,即对于有噪声的数据集,其密度曲线的跃迁区域远离 y 轴;而对于无噪声的数据集,则其密度曲线的跃迁区域非常靠近 y 轴.



(a) (b) *sort_dens* (c) *diff_dens*

Fig.9 Clustering result of data 4 and its threshold *denh* (*dcnh*=average connectivity)

图 9 Data 4 聚类结果及其密度阈值(*dcnh*=平均连通强度)

3.2 参数敏感性和算法伸缩性评估

在 *K*-MeanSCAN 算法中,一个重要的参数是预聚类时的子簇数目 k'.前文已经说明该参数对聚类结果不敏感,是“钝参数”.这里,我们在 Data 4 上对参数 k'进行敏感性分析.实验中,k'≈206,h=5,*dcnh*=0.最终结果中,簇数的变化在图 10 中给出.可以看出,在 k'>30 之后,*K*-MeanSCAN 算法均可获得满意的聚类结果.

在 SEQUOIA 2000 的点数据集上执行 DBSCAN,CURE 和 *K*-MeanSCAN 这 3 种算法的时间代价在图 11 中给出.其中,DBACAN 算法在 R*树索引基础上实现.在样本数量较少时(<10000),DBSCAN 算法性能较高;当样本数量继续增加时,采样技术使得 CURE 和 *K*-MeanSCAN 算法具有更好的伸缩性.

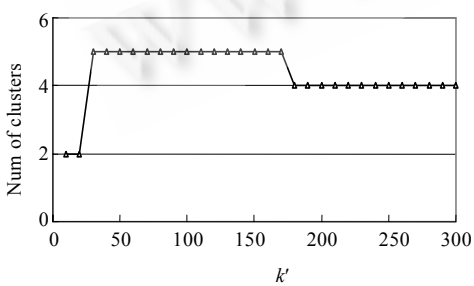


Fig.10 Sensitivity analysis of k'

图 10 参数 k'敏感性分析

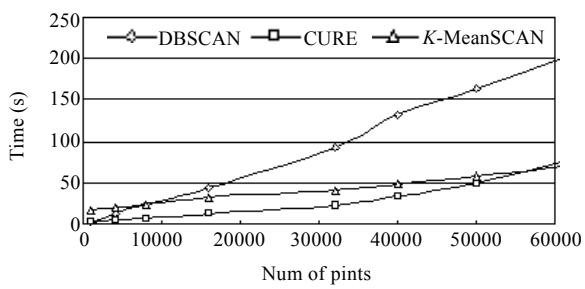


Fig.11 Comparison of algorithm scalability

图 11 算法的伸缩性比较实验

4 结论和下一步工作

本文简要综述了几种主要类型的聚类算法,特别是 *K*-Means 算法的概率方法本质和局部最优及结果敏感性特征.集成多种聚类方法的性质,提出了一种新的聚类算法——*K*-MeanSCAN,可以发现任意形状的类簇,并对异常点和噪声数据不敏感.同时,引入了连通强度的概念,避免由于密度连通关系的传递性导致绝大多数的样本点聚集到非常少的几个类簇中(通常是一类)的问题.此外,算法引入了钝参数的概念,减轻了聚类结果对参数阈值的敏感性;算法的时间复杂度约为 $O(n)$.

References:

[1] Han JW, Kamber M. Data Mining: Concepts and Techniques. 2nd ed., San Francisco: Morgan Kaufmann Publishers, 2001. 223−250.

[2] Ester M, Kriegel HP, Sander J, Xu XW. A density-based algorithm for discovering clusters in large spatial database with noise. In: Simoudis E, Han J, Fayyad UM, eds. Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996. 226−231.

[3] Zhang T, Ramakrishnan R, Linvy M. BIRCH: An efficient data clustering method for very large databases. In: Jagadish HV, Mumick IS, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM Press, 1996. 103−114.

[4] Guha S, RastogiR, Shim K. CURE: An efficient clustering algorithm for large databases. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 1998. 73−84.

[5] Ankerst M, Breuning M, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. In: Delis A, Faloutsos C, Ghandeharizadeh S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia: ACM Press, 1999. 49−60.

[6] Karypis G, Han EH, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. Computer, 1999,32(8): 68−75.

[7] Hand DJ, Vinciotti V. Choosing *k* for two-class nearest neighbour classifiers with unbalanced classes. Pattern Recognition Letters, 2003,24(9):1555−1562.

[8] Stonebraker M, Frew J, Gardels K, Meredith J. The SEQUOIA 2000 storage benchmark. In: Buneman P, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Washington: ACM Press, 1993. 2−11.

雷小锋(1975－),男,陕西合阳人,博士,主要研究领域为数据库,数据挖掘,机器学习.

林帆(1983－),男,硕士生,主要研究领域为时空数据挖掘.

谢昆青(1957－),男,博士,教授,博士生导师,主要研究领域为智能信息处理,时空数据库,数据挖掘.

夏征义(1974－),男,工程师,主要研究领域为计算机应用,信息系统分析设计.