一种适用于不规则分布数据的混合聚类算法

马志民[1,2], 陈汉武[1], 张 军[1]

(1.东南大学 计算机科学与工程系, 江苏 南京 210096
2.江西省信息中心, 江西 南昌 330046)

[摘要] 作为数据挖掘的一项重要技术, 聚类分析具有广泛的应用领域. 同时, 聚类也是数据挖掘领域中一个相对比较困难的问题. 在聚类算法中, 基于模糊划分的 FCM 算法是一种重要的算法. 和其它的算法相比, FCM 算法具有计算简单、运算速度快, 且有比较直观的几何意义的优点, 因此在图像处理、模式识别等领域得到了广泛的应用. 和所有的 c均值算法一样, FCM 算法也是只用类中心来表示类, 这样只是适合球状类型的簇. 本文在目前 FCM 算法研究的基础上, 讨论了传统 FCM 算法在原型初始化上的局限性. 提出一种基于层次凝聚的改进算法, 使之能够适用于不规则分布的数据.

[关键词] 模糊划分, FCM, 层次聚类, 模糊度量
[中图分类号] TP311.12 [文献标识码] A [文章编号] 1672-1292(2006)01-0057-04

A Hybrid Clustering Algorithm for Irregular Distributed Data

MA Zhimin, CHEN Hanwu, ZHANG Jun

(1. Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China;
2. Jiangxi Province Information Center, Nanchang 330046, China)

Abstract Clustering analysis, as an important technology of data mining, has a wide range of application areas, but at the same time, clustering is a rather difficult problem in data mining area. In common clustering algorithms, FCM based on fuzzy division is one of the important algorithms. Compared with other algorithms, FCM has many advantages such as simple computation, rapid speed and an intuitive geometric significance. So it has a wide application in many areas such as image processing and pattern recognition. As many c-means algorithms, FCM denotes class only by class center, which can only fit to sphere-like type of cluster. This dissertation discusses the limitations of traditional FCM algorithm in initialization of prototype. The paper presents a new algorithm based on hierarchical clustering which can be applied to the irregular distributed data.

Key words fuzzy partition, FCM, hierarchical clustering, fuzzy measure

将物理或者抽象对象集合划分成为由类似的对象组成的多个类的过程被称为聚类[1]. 由聚类所生成的簇是一组数据对象的集合, 这些对象与同一个簇中的对象彼此相似, 与其它簇中的对象相异. 在许多应用中, 可以将一个簇中的数据对象作为一个整体来对待.

一个聚类分析系统的输入是一组样板和一个度量两个样本间相似度(或相异度)的标准. 聚类分析的输出是数据集的几个组(类), 这些组构成一个分区或一个分区结构. 聚类分析的一个附加的结果是对每个类的综合描述, 这种结果对于更进一步深入分析数据集的特性是尤其重要的. 一般而言, 传统的聚类分析方法可以分为主要的两类: 基于划分的方法与层次方法.

1 FCM 算法

我们在此主要讨论基于划分的聚类方法. 传统的聚类分析是一种硬划分, 它把每个待分析的的对象严格的划分到某个类中, 一旦一个对象属于某个类那么它就不可能属于另外的一个类, 这种类别的划分界限

收稿日期: 2005-06-01.
基金项目: 国家自然科学基金资助项目(90412014).
作者简介: 马志民(1975-), 讲师, 主要从事数据挖掘等方面的研究. E-mail:mzmseu@sina.com

是分明的. 但在现实生活中很多对象并没有严格的隶属关系, 不能直接认定它们明确的属于那个类, 更多的是相对模糊的定义. 也就是说很多对象具有亦此亦彼的特性, 因此更适合软划分. 模糊理论的提出为这种软划分提供了有力的分析工具, 人们开始用模糊数学处理聚类问题, 并称之为模糊聚类分析. 由于模糊聚类得到了样本属于各个类别的不确定程度, 建立起了样本对于类别的不确定描述, 更能客观的反映现实世界, 从而成为了聚类分析的主流.

在众多的模糊聚类算法中 FCM 是应用最广泛的算法. 它将传统的 c-means算法的隶属度函数区间由 $\{0, 1\}$ 扩展到 $[0, 1]$, 构造目标函数:

$$J_m(u, v) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d(x_k, v_i).$$

在这里 $\sum_{i=1}^{c} u_{ik} = 1$, $u_{ik} \in (0, 1)$ 且 $\forall k$, $d(x_k, v_i) = \| x_k - v_i \|^2$, 通过迭代更新的方式使目标函数获得最小值, 然后根据最大隶属度原则确定每个点属于哪个类.

FCM 算法计算简单而且运算速度快, 具有比较直观的几何意义, 但是与 c均值算法一样只使用类中心来表示类, 这样只适合于发现球形或者类球形的簇. 虽然有不少改进的方法, 比如说对聚类原型模式进行扩展, 形成从特征空间的点到线、面、壳等诸多原型, 提出基于原型的模糊聚类算法, 象模糊 c线、模糊 c面、模糊 c壳以及模糊 c二次曲线等 FCM 类型的算法. 采用这些算法必须首先了解数据的分布特性, 但是在很多情况下数据的分布是未知的, 这也限制了这些基于原型算法的应用.

2 改进的混合聚类算法

在数据呈不规则分布的情况下, 可以考虑先将这些数据划分为若干个小的规则分布的类 (过划分), 然后用层次凝聚的方法将小的类进行合并, 最终得到满足要求的聚类结果[2]. 但是该方法没有指出如何解决两个关键的问题: 首先是代表点的选择, 也就是选择什么样的代表点进行过划分才能合适的将一个不规则的簇分为若干个小的簇; 其次是如何高效地将这些小的簇合并成为大的簇.

2.1 代表点的选择与过划分

在选择代表点的时候, 对于一个大的非规则簇, 我们要考虑如何在其中选择多个代表点将它划分为若干个簇. 考虑到簇与簇之间是用低密度区间隔开的高密度区域, 则我们可以考虑在密度连通域中选择代表点. 虽然 DBScan[3] 是一种相当成熟的寻找密度连通域的算法, 但是该方法没有考虑代表点的选择. Mitra, Murthy, Pal提出根据密度选择代表点的方法[4], 但是该方法只是适用于在数据中抽样而不适用于选择划分聚类的种子. Chaudhuri提出了一种选择代表点的方法[5], 该方法适合过划分的要求.

定义 **1** 当满足下列条件的时候, 点 x 和 y 关于点 z 对称

$$I(x, y)_z = \frac{d(x, y)}{d(x, z) + d(z, y)} \approx 1,\ x,\ y,\ z \in D \text{ 且 } x \neq z,\ y \neq z,$$ 其中 $d(x, y)$ 为两点之间的欧式距离.

定义 **2** 在点 z的 D 邻域内, 定义 I_z 为 $I(x, y)$ 的平均值. 可以看到, 如果 I_z 的值越小, 则 z越可能是边界上的点. 当值 $I_z < 1/2$的时候, z为边界点.

当考察完所有的点之后, 就可以得到所有的边界点集合. 对于球形或者类球形的簇而言只需要将它们的中心设置为代表点就可以了; 而对于拉长的或者非凸的簇而言, 需要在其中设置多个代表点, 每个点各自代表簇的不同部分. 可以看到, 对于一个簇而言, 它的中心一般是靠近高密度区域, 那么在此首先考虑选择密度最大的点作为代表点. Chaudhuri定义算法如下:

(1) 计算点的局部密度. (2) 按前面所述的方法寻找边界点. (3) 选择 D 当中密度最大的点 p 作为代表点. (4) 计算 p到代表点集合的最大距离 f_{max}、最小距离 f_{min}. 当 $f_{max} - f_{min}$ 小于阈值时, 转向 (6); 否则转向 (5). (5) 将到 p距离小于等于 f_{min}的点从 D 当中移除, 当 $|D|$ 很小的时候转向 (6); 否则转向 (3). (6) 用得到的代表点作为原型采用 FCM 算法进行过划分.

如图 1所示的图形, 使用上述算法后可以看到划分为 7个小的子类.

可以看到, Chaudhuri所提出的方法要求考察每个点的邻域来判断该点是不是边界点, 也就是说对于一个点 x来说, 需要判断其它点是否在其领域内并对在其领域半径内的所有点进行计算来决定 x是否为边

界点, 很显然这样做的计算量很大. 我们在此将空间划分为若干个大小相等的超立方体, 将判断一个点是否为边界点的问题改为判断一个立方体是否为边界立方体, 虽然该方法在精度上有所损失, 但是在计算时间上却有很大的减少.

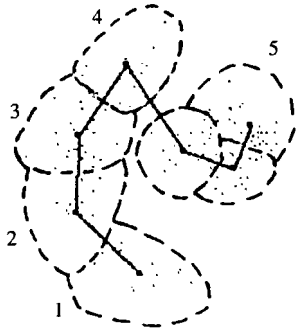

图 1 将不规则簇过划分后的结果

定义 **3** 当满足 $I(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}, B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d}) \approx 1$时, 超立方体 $B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}$, $B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d}$ 关于 $B_{i_1,\cdots,i_{j-1},i_j,i_{j+1},\cdots,i_d}$ 对称. (*d* 为空间的维数)

其中, 当 $\mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}) < \mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d})$时,

$$I(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}, B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d}) = \frac{\mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d})}{\mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d})};$$

反之则

$$I(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}, B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d}) = \frac{\mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d})}{\mathrm{number}(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d})}$$

(当两个立方体当中的点都为 0的时候函数值为 1)

定义 **4** 在超立方体 $B_{i_1 i_2\cdots i_j\cdots i_d}$ 的所有维上的相邻单元中, 定义 $I_{B_{i_1 i_2\cdots i_j\cdots i_d}}$ 为 $I(B_{i_1,\cdots,i_{j-1},i_j-1,i_{j+1},\cdots,i_d}, B_{i_1,\cdots,i_{j-1},i_j+1,i_{j+1},\cdots,i_d})$ 平均值. 当 $I_{B_{i_1 i_2\cdots i_j\cdots i_d}} < 1/2$, 称 $B_{i_1 i_2\cdots i_j\cdots i_d}$ 为边界区域.

改进算法如下:

(1) 将空间划分为若干个大小相等的超立方体集合 *B*.

(2) 计算每个立方体点的个数, 用该个数作为当中每个点的密度.

(3) 按新的定义考察每个立方体寻找边界区域.

(4) 选择 *B* 当中密度最大的立方体中心点 *p* 作为代表点.

(5) 计算 *p* 到边界区域的最大距离 f_{max}、最小距离 f_{min}. 当 $f_{max} - f_{min}$ 小于阈值时, 转向 (7); 否则转向 (6).

(6) 将到 *p* 距离小于等于 f_{min} 的立方体从 *B* 当中移除, 当 $|B|$很小的时候转向 (7); 否则转向 (4).

(7) 用得到的代表点作为原型采用 FCM 算法进行过划分.

在此我们用点的集合 (超立方体) 代替了一个一个的点, 将寻找边界点的计算改为寻找边界区域的计算. 如图 2所示是使用数据生成程序 DataGen随机生成 1万 ~ 10万个数据, 在其中寻找边界所需要的时间, 我们可以看到随着数据量的增长, 原算法的所需要的计算时间快速增多, 而改进算法的所需要的计算时间增长速度明显小于原算法. 而且当我们选择合适的立方体的时候, 得到的边界点和实际的结果相差并不大, 最后可以将数据集进行正确的划分. 当然将数据立方体代替单个的数据点, 在寻找边界的过程中会有精度的损失, 如何选择合适的立方体大小在精度和计算时间上达到平衡是一个难以解决的问题, 也是我们下一步研究的重点. 在目前一般是根据经验选取, 在取值比较多的维上划分更多的区间, 而在取值比较少的维上划分较少的区间, 并且在实验中调整区间的划分大小. 另外, 我们在此并不是直接用该边界作为簇的划分, 快速找到一个合适的代表点才是主要的目的. 而代表点和实际的簇中心之间的偏差可以通过 FCM 算法的迭代来消除.

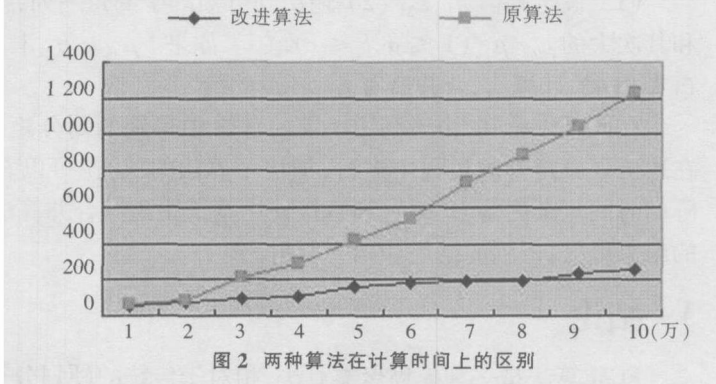

图 2 两种算法在计算时间上的区别

2.2 合并子类

在此算法中, 首先要考虑根据什么样的标准进行聚类, 在普通的层次聚类中最常用的是根据类间距离

的大小决定是否进行凝聚. 但是在最坏的情况下, 计算两个簇 A、B 间距离的时间复杂度是 $O(m_1 m_2)$ ($m_1 = |A|$, $m_2 = |B|$), 那么合并所有簇的时间复杂度在最坏的情况下是 $O(n^2)$. 考虑到在过划分过程中已经进行过大量计算, 我们应尽可能利用前面的计算结果来减少合并过程的计算量.

在第一阶段的算法完成之后, 可以得到每个点到簇中心的划分矩阵 $\boldsymbol{U}$, 在此可以考虑利用该划分矩阵进行聚类. 在评价聚类质量的时候, 一个好的聚类是各个簇内部成员之间相似度大而簇与簇之间的相似度小. 那么当把一个大的簇划分为若干个小的簇时, 这些小的簇之间必然具有较大的相似度. 对于模糊聚类而言, 评价类与类之间的模糊相关度还可以采用子集测度的方法[2].

定义 **5** 如果 f 满足以下性质, 则实函数 $f: F(X) \times F(X) \to [0, 1]$ 叫做 $F(X)$ 上的一个子集测度.

(1) 如果 $A \subset B$, 则 $f(A, B) = 1$ (2) $f(X, \varnothing) = 0$. (3) 如果 $A \subset B \subset C$, 则 $f(C, A) \leqslant f(B, A)$, $f(C, A) \leqslant f(C, B)$.

定义 **6** Kosko子集测度公式

$$f_K(A, B) = \begin{cases} 1, A = \varnothing \\ \dfrac{M(A \cap B)}{M(A)}, A \neq \varnothing \end{cases} \quad \text{其中} \quad M(X) = |X|.$$

我们知道, 在聚类分析中, 一个好的分类应该使相似的样本尽可能分在同一类中. 一个模糊聚类的结果是对数据集进行了模糊划分. 当我们用 Kosko子集测度公式对其中两个类进行判断时, $f_K(A, B)$ 的值越小说明它们的边界越清晰, 当它的值越大的时候它们的边界越模糊. 也就是说对于一个好的分类两个簇应该尽可能分离, 即 A 包含在 B 的程度尽可能小. 那么当我们把一个大的簇划分为若干个小的簇的时候, 很显然他们之间的联系非常紧密, 也就是说它们相互包含的程度很大. 根据以上分析, 我们认为当 $f_K(A, B)$ 的值超过一定的值以后可以将两个类进行合并.

从前面定义可知, 对于模糊聚类而言, 是根据划分矩阵来确定每一个样本隶属于那个类. 很显然对于一个样本 x_k 而言, 如果它越靠近某一个类 X_i ($1 \leqslant i \leqslant c$) 的中心其隶属度函数 μ_{ik} 的取值就越大, 而当它位于两个类的边界时它相对这两个类的隶属度函数的取值越接近, 根据这一特点我们把这样的点认为是在两个类的交集内. 于是定义合并算法如下:

(1) 设定阈值 ε_1, ε_2 (2) 遍历划分矩阵中的每一列, 找出其中最大的和其次大的 μ_{ak}, μ_{bk} ($1 \leqslant a, b \leqslant c$). (3) 如果 $|\mu_{ak} - \mu_{bk}| < \varepsilon_1$ 则 $x_k \in X_a \cap X_b$. (4) 计算 f_K, 合并所有 $f_K > \varepsilon_2$ 的簇.

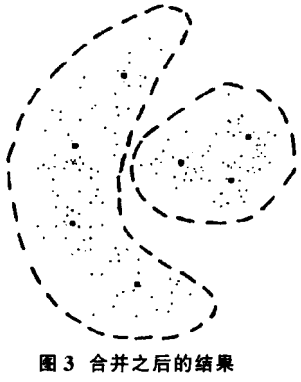

图 3 合并之后的结果

在此可以看到, 由于利用了前面计算中得到的划分矩阵, 合并簇可以在遍历划分矩阵的过程中进行, 加快了算法的执行. 可以看到遍历划分矩阵的时间复杂度为 $O(cn)$, 和直接使用基于距离的合并算法相比可以减少很多计算时间. 用该方法对图 1的结果聚类, 得到的结果如图 3所示.

3 结论

FCM 算法是一种模糊聚类算法, 相对于传统的硬划分算法更能反映现实世界. 但是传统的 FCM 只能够对球形 (类球形)结构分布的数据进行聚类, 而不能够对不规则形状的簇进行聚类. 将基于层次的和基于划分的算法进行混合的聚类算法, 则既可保持 FCM 算法的优点, 又可适用于呈不规则分布的数据, 该方法需要考虑如何选择代表点进行过划分和将过划分的结果快速合并. 本文在这两个问题上进行了探讨: 在代表点选择中, 将空间划分为若干个超立方体将求边界点改为求边界区域, 减少了计算量; 在合并算法中引入了 Kosko子集度量, 利用了过划分计算中所得到的划分矩阵进行合并操作, 从而加快了合并操作的执行速度.

(下转第 76页)

搅拌站中的开发、应用,上述技术难题已被克服.

3 结束语

基于上述控制模型和算法的连续强制式水泥混凝土搅拌站,由于成功地利用了计算机技术、现代控制技术及机电一体化设计技术,与传统的间隙式水泥混凝土搅拌站相比,具有成本低、产量高、节能、环保、维护方便等特点,市场前景十分广泛.产品现已成功进入国际市场.

[参考文献](References)

[1] 王树清, 赵鹏程. 集散型计算机控制系统(DCS)[M]. 杭州: 浙江大学出版社, 1994.
WANG Shuqing ZHAO Pengcheng Distributed Computer Controll Systems[M]. Hangzhou: Zhejiang University Press, 1994. (in Chinese)
[2] GRAHAM C GOOGWIN. Control Systems Design[M]. Beijing Tsinghua University Press, 2002. (in Chinese)
[3] KARL J ASTROM. Computer Controlled Systems[M]. Englewood Cliffs, New Jersey: PrenticeHall, Inc, 1984.
[4] 何克忠, 李伟. 计算机控制系统[M]. 北京: 清华大学出版社, 1998
HE Kezhong LI Wei Computer Controlled Systems[M]. Beijing Tsinghua University Press, 1998 (in Chinese)
[5] 韩曾晋. 自适应控制[M]. 北京: 清华大学出版社, 1995.
HAN Zengjin. Adaptive Control[M]. Beijing Tsinghua University Press, 1995. (in Chinese)

[责任编辑: 刘 健]

(上接第 60页)

[参考文献](References)

[1] JIAWEI HAN, MICHELINE KAMBER. Data Mining Concept and Techniques[M]. San Fransisco: Morgan Kaufmann Publishers, Inc, 2001: 223-239.
[2] 高新波. 模糊聚类分析及其应用[M]. 西安: 西安电子科技大学出版社, 2004: 92-97.
GAO Xinbo. Fuzzy Cluster Analysis and its Application[M]. Xi'an: Xidian University Publisher, 2004: 92-97. (in Chinese)
[3] ESTE J, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[J]. Proc KDD, 1996: 226-231.
[4] PABITRA MITRA C A, MURTHY, SANKAR K PAL. Density-based multiscale data condensation[J]. IEEE Trans PAMI, 2002, 6(24): 734-747.
[5] CHAUDHURI D, CHAUDHURI B B. A novel multiseed nonhierarchical data clustering technique[J]. IEEE Trans SMC, 1997, 10(27): 871-877.

[责任编辑: 刘 健]