

文章编号:1001-9081(2009)03-0830-03

基于网格与分形维数的聚类算法

梁敏君^{1,2},倪志伟^{1,2},倪丽萍^{1,2},杨葛钟啸^{1,2}

(1. 合肥工业大学 管理学院,合肥 230009; 2. 合肥工业大学 过程优化与智能决策教育部重点实验室,合肥 230009)
(minjunliang@sina.com)

摘要:提出了一种基于网格和分形维数的聚类算法,它结合了网格聚类和分形聚类的优点,克服了传统网格聚类算法聚类质量降低的缺点,改进了分形聚类耗时较大的问题。此算法首先根据网格密度得到初始类别,再利用分形的思想,将未被划分的网格依次归类。实验结果证明,它能够发现任意形状且距离非邻近的聚类,且适用于海量、高维数据。

关键词:聚类;分形维数;网格

中图分类号: TP311.13 **文献标志码:** A

Clustering algorithm based on grid and fractal dimension

LIANG Min-jun^{1,2}, NI Zhi-wei^{1,2}, NI Li-ping^{1,2}, YANGGE Zhong-xiao^{1,2}

(1. School of Management, Hefei University of Technology, Hefei Anhui 230009, China;
2. Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education,
Hefei University of Technology, Hefei Anhui 230009, China)

Abstract: Combining the approaches based on grid and fractal, a new kind of clustering algorithm called grid and fractal dimension based clustering algorithm (GFDC) was presented. It overcame the shortcoming of low clustering quality in traditional grid-based clustering method and solved the time-consuming problem of fractal clustering method. In the initial stage of the new algorithm, some initial clusters were got according to the grid density. Then in the expansion stage, the unclassified grids were categorized using the idea of fractal. Experimental results confirm that GFDC is able to capture arbitrary shapes and non-neighboring clustering, and can be applied to the massive and high-dimension dataset.

Key words: clustering; fractal dimension; grid

0 引言

聚类就是将数据对象分成类的过程,使得同类对象间的相似度尽可能高,而不同类中的对象高度相异^[1]。国内外学者针对不同的应用,不同的聚类目的或不同的数据类型,提出了很多的聚类方法。然而,基于划分的聚类方法^[2-3]是基于欧氏距离度量机制,虽然在低维数据分析中获得了优良的性能和令人满意的聚类结果,但无法发现非球形聚类或者大小差别很大的聚类;基于层次^[4]、密度^[5-6]和网格^[7-8]的聚类技术虽在大数据集、增量及任意形状聚类方面与基于划分的聚类方法相比具有较大的优越性,然而却无法应对聚类内部密度不均匀及聚类非邻接的情况。

分形理论提出了一套定量描述自然界中不规则、复杂现象的强有力工具,它的核心思想是局部和整体之间的自相似性^[9],这个思想已经在图像处理、图像压缩、信号处理等领域得到广泛应用。同时,分形的思想也渐渐渗透到数据挖掘领域,比如被用以解决属性约简^[10]、聚类^[11]等问题。

文献[11]提出的分形聚类算法(Fractal Clustering, FC),虽然在分布相隔较远的实验数据集上取得了较好的效果,但它所使用的树状结构空间代价高、维护难度大、时间复杂度高。针对这一问题,本文结合了基于网格的方法,提出一种基于网格和分形维数的聚类算法(Grid and Fractal Dimension based Clustering, GFDC)。本方法在聚类过程中将网格中的所

有点作为一个整体处理,具有速度快、可扩展性好,能识别不同形状及分布较远的聚类的优点,且能有效地处理高维、海量的数据。

1 分形及分形维数

定义 1 分形。如果一个数据集在所有的观察尺度下都具有自相似性,即一个数据集的部分有着与整体分布相似的结构或属性,称这个数据集是分形。

定义 2 嵌入维数。数据集中的数据点所在的欧式空间的维数称为该数据集的嵌入维数,即一个数据集中属性的个数。比如一条直线若位于二维平面内,其嵌入维为 2;若位于三维空间之中,则嵌入维数为 3。

定义 3 固有维数。一个数据集的固有维数是指一个数据集所表示的空间的实际维度,而不随数据集嵌入维数的不同而改变。它在一定程度上定量地描述了数据集内部结构的复杂性。如果设数据集的嵌入维数为 E ,固有维数为 D ,则有 $D \leq E$ 。

定义 4 分形维数。一个数据集的分形维数体现于数据集的固有特征。它是描述分形集合不规则性和复杂性程度的度量。

在数据挖掘中,分形维数是对数据集分布自由程度的估计,反映了数据在多维空间中的分布特性和对空间的填充能力。对于嵌入维数为 n 的数据集,将它嵌入到每一个单元格边长为 r ($r \in (r_1, r_2)$) 的 n 维格子中, (r_1, r_2) 是数据集具有分形特性的

收稿日期:2008-09-05;修回日期:2008-11-07。

基金项目:国家 863 计划项目(2007AA04Z116);国家自然科学基金重点项目(70631003)。

作者简介:梁敏君(1983-),女,浙江台州人,硕士研究生,主要研究方向:数据挖掘、人工智能;倪志伟(1963-),男,安徽桐城人,教授,博士生导师,主要研究方向:管理信息化、决策科学与技术、软件工程。

边长度量的变化范围。这样,可以计算出落入第 i 个单元格中数据点的数目,记作 C_i^r ,则其分形维数 D_q 定义如下:

$$D_q = \frac{1}{q-1} \frac{\partial \ln \sum_i C_i^r}{\partial \ln r}; r \in (r_1, r_2) \quad (1)$$

其中 q 的不同取值可以用来计算不同的分形维度值,这些分形维度值从不同的角度描述了数据集的特征。当 $q=0$ 时,计算得到的 D_0 是Hausdorff分形维数;当 $q \rightarrow 1$ 取极限时,计算得到的 D_1 是信息维数;当 $q=2$ 时,计算得到的 D_2 是相关维数。 D_1 和 D_2 在数据挖掘中应用比较多。 D_2 表征了随机选取的两个点之间距离小于一个特定值的概率,反映了数据的分布特性。文献[4]证明了 D_2 用于聚类问题时反映数据点密度和维度比较明显,在本文及相关实验中也使用 D_2 作为分形维数的尺度。

式(1)中将 $\ln \sum_i C_i^r$ 对 $\ln r$ 取偏导数,实际上就是求曲线 $(\ln r, \ln \sum_i C_i^r)$ 的斜率,在本文实验中,对不同的 r 值分别计算并保存 $\ln \sum_i C_i^r$ 和 $\ln r$ 的值,然后用最小二乘法得到曲线的斜率,即该数据集的分形维数 D_q 。

2 基于网格与分形维数的聚类算法

2.1 数据结构及概念

定义5 空间 S 。设 $A = \{A_1, A_2, \dots, A_d\}$ 是有界域的集合,那么 $S = A_1 \times A_2 \times \dots \times A_d$ 是一个 d 维数据空间,其中 A_1, A_2, \dots, A_d 表示 S 的 d 个维或 d 个属性域。

定义6 点集 X 。 $X = \{x_1, x_2, \dots, x_n\}$ 表示 S 上的 N 个点的集合,其中 $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ 表示一个数据点, x_i 的第 j 维分量 $x_{ij} \in A_j$ 。

定义7 值域长度 D_j 。设点集 X 第 j 维的所有分量的最大值为 $D_{j\max} \in A_j$,则对 $\forall x_{ij} \in A_j$ 都有 $D_{j\max} \geq x_{ij}$,同理设点集 X 第 j 维的分量的最小值 $D_{j\min} \in A_j$,对 $\forall x_{ij} \in A_j$,都有 $D_{j\min} \leq x_{ij}$,则点集 X 第 j 维的值域长度 $D_j = D_{j\max} - D_{j\min}$ 。

将点集 X 中所有数据点按照其各维坐标值映射到空间 S 中,构建多层网格结构,将数据集空间整体视为第1层网格,则第 j 维坐标上第 k 层网格的边长为 $r_{kj} = D_j/2^{k-1}$,此时空间 S 中共有 $2^{(k-1)d}$ 个网格。

以二维数据空间为例,为描述方便,设 $D_1 = D_2 = 1$,则嵌套的下层网格的边长按照如下方式划分:1, $1/2^1$, \dots , $1/2^{m-1}$,其中 m 是网格结构的层数。图1给出二维空间从底层网格坐标逐层映射为其高层网格坐标的过程,其中 $m=4$ 。顶层网格对应数据空间整体,包含4个第二层网格(边长为 $1/2^{(2-1)}$),可见上层的每一个网格均包含其直接下层的 $4(2^2)$, d 维空间为 2^d 个网格(边长为上层网格边长减半)。

设点集 X 构建的多层网格结构共 m 层,则底层网格共 $2^{(m-1)d}$ 个单元格,该层第 j 维网格的边长为 r_{mj} 。设定数据点 x_i 所在底层网格的坐标编号为 $(g_{m1}g_{m2}\dots g_{md})$,其中 g_{mj} 是点 x_i 第 j 维分量所在的底层网格的坐标编号,则 $g_{mj} = \frac{x_{ij} - D_{j\min}}{r_{mj}}$ 。

本算法只需在创建初始的底层网格结构时,扫描一遍点集 X ,记录每个数据点所属的网格坐标,统计每个网格中包含的数据点数,其他任意一层的网格坐标及其统计信息都可由底层网格映射得到。若在第 k 层的某一网格 $Grid_i$ 的坐标编号为 $(g_{k1}g_{k2}\dots g_{kd})$,其中 g_{kj} 是该网格在第 j 维坐标上的编号,而该网格所属的直接上层即第 $(k-1)$ 层的网格坐标编号为 $(g_{(k-1)1}g_{(k-1)2}\dots g_{(k-1)d})$,其中 $g_{(k-1)j}$ 为其在第 j 维坐标上的编

号,则从第 k 层向第 $(k-1)$ 层的映射按照式(2)进行。

$$g_{(k-1)j} = \text{int}(g_{kj}/2) + (g_{kj} \bmod 2) \quad (2)$$

定义8 网格密度。网格 $Grid_i$ 的密度 $\text{density}(Grid_i) = S_i/N$,其中 S_i 为网格 $Grid_i$ 所包含的数据点数, N 为数据集的数据点数。

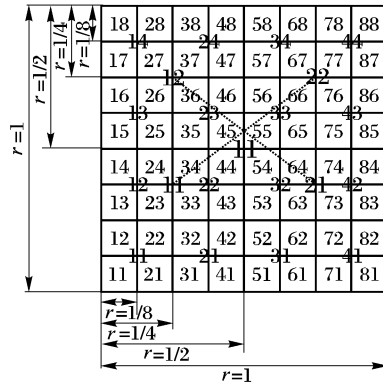


图1 二维嵌套网格结构及网格编号

2.2 GFDC 算法描述

为了能够处理大量数据并且对数据有增量处理能力,本文提出的基于网格和分形维数的聚类算法(GFDC)分为初始聚类阶段和扩展聚类阶段。在初始聚类阶段通过一遍扫描数据集来创建初始的底层网格结构,利用网格和密度的聚类技术在整个数据集中寻找部分点来生成初始类;在扩展阶段,利用初始阶段创建的网格结构,结合分形理论,将其他数据点以网格为单位添加到初始类中去,最终形成在整个数据集上的聚类结果。

2.2.1 初始聚类

相同密度的区域之间具有较强的相似性,但是不同密度区域之间的相似性很弱,数据的分布是不同密度区域混杂的,初始阶段根据网格密度的相似性及连通性在不同密度区域混杂的情况下,快速选出一些网格密度相同的区域作为扩展的基础。

算法1步骤如下:

步骤1 扫描数据集,构建 m 层网格结构。计算点集 X 中的每个点所在的底层网格坐标 $Grid_i = (g_{m1}g_{m2}\dots g_{md})$,并统计每个底层网格所包含的数据点数 S_i 。

步骤2 计算底层网格密度 $\text{density}(Grid_i)$,以网格密度最大的网格 $Grid_h$ 为中心,扫描其邻近的网格 $Grid_l$,若 $|\text{density}(Grid_h) - \text{density}(Grid_l)| < \tau$,则聚为一类,其中 τ 为密度差别阈值。

步骤3 上一步聚得的类记为 C_i ,在剩余网格中选取单元网格密度最大的单元 $Grid_h$,若 $|\text{density}(C_i) - \text{density}(Grid_h)| < \tau$,则将 $Grid_h$ 并入类 C_i ,然后重新搜寻剩余网格中密度最大者,否则将其与邻近网格的密度相比较,若差值小于密度差别阈值 τ ,则合并为一新类。

步骤4 重复步骤3,直至初始类数达到用户设定的初始聚类数 k 。

2.2.2 扩展聚类

扩展阶段处理的思路是:在初始聚类的基础上,对于数据集中没有被划分的网格逐个进行处理,如果它属于某个类,即它所具有的内在结构与整体的结构是相似的,那么将它加入到现有类中不会引起类的分形维数值的剧烈变化;相反,如果它与某个现有类不具有相同的内在结构,那么它的加入会引起这个类的分形维数值的剧烈变化。而且在扩展阶段利用了初始阶段创建的多层网格结构,大大加快了分形维数的计算

效率。

算法 2 步骤如下:

步骤 1 分别计算在初始聚类阶段得到的 k 个初始类的分形维数 $f_i, i = 1, 2, \dots, k$ 。

步骤 2 对于底层网格中没有被划分的网格逐个分别加入到各个初始类中,此时再计算各类的分形维数 $f_i', i = 1, 2, \dots, k$,令 $\Delta f_i = f_i' - f_i, i = 1, 2, \dots, k$ 。

步骤 3 选取出 Δf_i 最小的类,对设定的阈值参数 δ ,若 $\min(\Delta f_i) < \delta, i = 1, 2, \dots, k$,则将该网格中的数据点加入到这个类中;否则,新建一类,把该网格中数据点加入到新类。

步骤 4 重复步骤 2 和步骤 3,直至所有的网格都被归类。

步骤 5 步骤 4 得到聚类数 L 和各类的分形维数 $F_i, i = 1, 2, \dots, L$ 。此时若 $L \geq 5k$,则令 $\Delta = |F_i - F_j|, i, j = 1, 2, \dots, L$,对于给定的阈值参数 ε ,当 $\Delta < \varepsilon$ 时,那么将 F_i 和 F_j 合并为一类,以合并后得到的结果作为最终的聚类结果。

3 实验研究

本文实验的硬件环境是 P4 CPU 2.93 GHz, 512 MB 内存;操作系统为 Windows XP,采用 C#语言在 vs2005.net 环境下实现 GFDC 算法。

实验 1 所用数据集是由 Matlab 软件生成的二维数据集,其数据分布如图 2 所示。现改变数据集的点来测试本文算法的聚类性能,其结果如表 1 所示。GFDC 算法将该数据集聚为两类,分布图中左下角的正方形区域 A 中的数据与右上角的环形区域 B 中的数据被聚为第一类,数据分布密度较小的区域 C 中的数据被聚为第二类,可见本算法可以识别任意形状的簇,且不受空间上限制,能够发现距离非邻接的类。从表 1 中还可看出 GFDC 算法同样适用于大数据集,且具有良好的聚类精度。

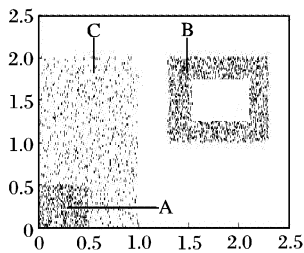


图 2 实验 1 数据分布

表 1 聚类结果

点数	时间 /s	所得 类	点的 分配	点来源		精确度 /%
				Cluster1	Cluster2	
30 000	9	1	20 125	19 814	311	99.07
		2	9 875	186	9 689	96.89
300 000	41	1	200 082	199 004	1 086	99.5
		2	99 918	996	98 914	98.91
3 000 000	312	1	2 002 470	1 998 313	4 157	99.92
		2	997 530	1 687	995 841	99.58

在效率方面,本文算法提出的数据结构采用由小粒度到大粒度的策略来计算分形维数值,并在计算小粒度网格统计信息的同时生成大粒度网格的统计信息,只涉及简单的数值计算,这与 FC 算法所使用的树状结构相比,降低了维护的难度,提高了运算的效率。实验 2 用包含 30 000 条数据的数据集来测试,将 GFDC 算法与 FC 算法^[4]作比较,分析随着维数增加算法所需时间的变化,结果如图 3 所示。通过上述实验

可以看出,本文提出的 GFDC 算法的执行效率较高,且适用于高维的数据。

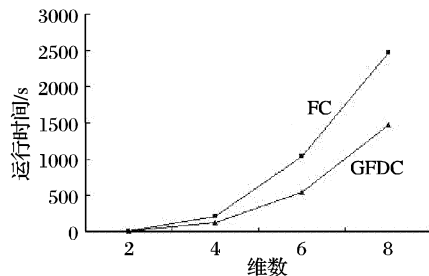


图 3 实验 2 维数与运行时间变化

4 结语

本文提出的基于网格和分形维数的聚类算法只需对数据集进行一次扫描来建立网格结构,并且自始至终都以网格单元为整体进行运算,而这样的多层网格结构大大降低了分形维数计算的空间复杂度和时间复杂度,并且具有良好的可扩展性。实验结果证明了本算法能够准确有效地完成聚类任务,且适用于高维、海量的数据,可应用于空间数据聚类、商业数据聚类等领域。

参考文献:

- [1] HAN J, KAMBER M. 数据挖掘: 概念与技术[M]. 2 版. 北京: 机械工业出版社, 2007.
- [2] XIA SHIXIONG, LI WENCHAO, ZHOU YONG, et al. Improved k-means clustering algorithm[J]. Journal of Southeast University (English), 2007, 23(3): 435-438.
- [3] PELLEGE D, MOORE A W. X-means: Extending K-means with efficient estimation of the number of the clusters[C]// Proceedings of the 17th ICML. San Francisco, CA: Morgan Kaufmann, 2000: 727-734.
- [4] GUHA S, RASTOGI R, SHIM K. CURE: An efficient clustering algorithm for large databases[C]// Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. New York: ACM, 1998: 73-84.
- [5] ESTER M, KRIEGER H-P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proceedings of the 2nd ACM SIGKDD. New York: ACM, 1996: 226-231.
- [6] 周水庚, 周傲英, 曹晶, 等. 一种基于密度的快速聚类算法[J]. 计算机研究与发展, 2000, 37(11): 1287-1292.
- [7] WANG W, YANG J, MUNTZ R. STING: A statistical information grid approach to spatial data mining[C]// Proceedings of the 23rd Conference on VLDB. San Francisco, CA, USA: Morgan Kaufmann, 1997: 186-195.
- [8] 陈宁, 陈安, 周龙骧. 基于密度的增量式网格聚类算法[J]. 软件学报, 2002, 13(1): 1-7.
- [9] 张济忠. 分形[M]. 北京: 清华大学出版社, 2001.
- [10] 鲍玉斌, 王琢, 孙焕良, 等. 一种基于分形维的快速属性选择算法[J]. 东北大学学报: 自然科学版, 2003, 24(6): 527-530.
- [11] BARBARA' D, CHEN PING. Using the fractal dimension to cluster datasets[C]// Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000). New York: ACM Press, 2000: 260-264.
- [12] 闫光辉, 李战怀, 党建武. 基于多重分形的聚类层次优化算法[J]. 软件学报, 2008, 19(6): 1283-1300.
- [13] 姜园, 张朝阳, 仇佩亮, 等. 用于数据挖掘的聚类算法[J]. 电子与信息学报, 2005, 27(4): 655-662.
- [14] 王鑫, 王洪国, 王珏, 等. 数据挖掘中聚类方法比较研究[J]. 计算机技术与发展, 2006, 16(10): 20-22.