

数据挖掘中聚类方法比较研究

王鑫¹, 王洪国², 王珣², 王金枝³

(1. 山东师范大学 管理学院, 山东 济南 250014;

2. 山东师范大学 信息管理学院, 山东 济南 250014;

3. 烟台大学 海洋学院, 山东 烟台 264005)

摘要:数据挖掘是近年来信息产业界非常热门的研究方向, 聚类分析是数据挖掘中的核心技术。聚类算法已被广泛深入地研究, 其间产生了许多不同的适用于数据挖掘的聚类算法, 但这些算法仅适用于特定的问题及用户。为了更好地使用这些算法, 文中对数据挖掘领域的聚类分析方法及代表算法进行了分析, 提出了数据挖掘对聚类的典型要求, 并基于这些要求对数据挖掘中常用的聚类算法作了比较, 以便于人们更容易、更快速地选择一种适用于具体问题的聚类算法。

关键词:数据挖掘; 聚类; 聚类算法

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2006)10-0020-03

Comparison of Clustering Methods in Data Mining

WANG Xin¹, WANG Hong-guo², WANG Jun², WANG Jin-zhi³

(1. Management School of Shandong Normal University, Jinan 250014, China;

2. Information and Management School of Shandong Normal University, Jinan 250014, China;

3. Ocean School of Yantai University, Yantai 264005, China)

Abstract: Data mining is one of pop research in information industry last few years. Clustering analysis is the core technique of data mining. Clustering method has been studied very deeply. During the time occurred many different clustering methods that suit data mining, but these methods are only suited special problems and users. In order to use these methods better, analysis the clustering methods and representative clustering algorithm, put forward the typical requests of clustering and compared the common clustering algorithm, so that people can easily find a clustering method that suit a special problem.

Key words: data mining; clustering; cluster algorithm

0 引言

随着数据挖掘研究领域技术的发展, 作为数据挖掘主要方法之一的聚类算法, 也越来越受到人们的关注。聚类分析是数据挖掘研究和应用中一个重要的部分。简单地讲, 聚类分析就是将数据对象分组成多个类或簇(cluster), 在同一个簇中的对象之间具有较高的相似度, 而不同簇中的对象差别较大^[1]。

1 数据挖掘对聚类的典型要求

数据挖掘的聚类一般是针对大数据集而言的, 因此在数据挖掘中聚类方法的比较应该满足以下要求:

1) 可伸缩性。算法在满足小数据集的同时能否满足大数据集、高复杂性、高增量的要求。

2) 处理不同类型属性的能力。算法在处理数值类型数据的同时能否处理其他的数据类型, 如二元类型、分类/标称型、序数型及混合数据类型。

3) 发现任意形状的类。许多基于距离的算法只能发现具有相近尺度的球状簇, 而对于一个簇可能是任意形状的, 算法能否发现任意形状的簇很重要, 如螺旋型。

4) 用于决定输入参数的领域知识最小化。许多算法要求用户输入一定的参数(如希望产生的簇数)。聚类结果对输入的参数十分敏感, 通常参数较难确定, 尤其对于含高维参数的数据集更是如此。要求人工输入参数加重了用户的负担, 而且也使聚类质量难以控制。

5) 处理噪声数据的能力。实际数据集都包含孤立点、空缺、未知数据或错误等。算法能否降低这些噪声数据的影响。

6) 对输入数据顺序的敏感性。算法能否与输入顺序无关。

7) 处理高维数据的能力。算法在应付低维数据的同

收稿日期: 2005-12-14

基金项目: 山东省自然科学基金重大项目(Z2004G02); 山东省优秀中青年科学家奖励基金项目(03BS003); 山东省科委资助项目(012090101)

作者简介: 王鑫(1979-), 女, 山东烟台人, 硕士研究生, 研究方向为数据挖掘、知识发现; 王洪国, 博士后, 教授, 硕士生导师, 主要研究方向为数据挖掘。

时能否处理高维空间的非常稀疏、高度偏斜的数据。

2 数据挖掘领域中聚类算法分类

聚类算法大体上可以分为以下几种: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。

2.1 划分方法

给定一个有 N 个元组或者记录的数据集,构造 K 个分组,每一个分组代表一个聚类($K < N$)。对于给定的 K ,可以给出一个初始的分组方法,以后通过反复迭代来改变分组,使得每一次改进之后的分组方案都较前一次好。这些聚类方法在对中小规模的数据库中发现球状簇很适用。为了对大规模的数据集进行聚类,基于划分的方法需要进一步的扩展。

2.2 层次方法

对给定的数据集进行层次似的分解,直到某种条件满足为止。层次聚类方法可分为自下而上和自上而下两种基本方法。自下而上方法是以数据对象作为原子类,然后将这些原子类进行聚合。逐步聚合成越来越大的类,直到满足终止条件。自上而下方法是首先将所有数据对象作为一类,然后逐步分解成越来越小的类,直到满足终止条件。层次聚类方法的缺陷在于,一旦一个步骤完成,它就不能被撤销,因此就不能更正错误的决定。改进层次方法聚类质量的一个有希望的方向是将层次聚类和其他聚类技术进行集成,形成多阶段聚类。

2.3 基于密度的方法

其主要思想是:只要邻近区域的密度超过某个阈值,就继续聚类,数据稀疏区域中的数据点认为是噪音数据。这样的方法可以用来过滤“噪音”孤立点数据,发现任意形状的聚类。

2.4 基于网格的方法

首先将数据空间划分成为有限个单元(cell)的网格结构,所有的处理都是以单个的单元为对象的。这种方法的主要优点是处理速度快,其处理时间独立于数据对象的数目,只与量化空间中每一维的单元数目有关。

2.5 基于模型的方法

给每一个聚类假定一个模型,然后去寻找能够很好地满足这个模型的数据集。这个模型可能是数据点在空间中的密度分布函数,它由一系列的概率分布决定。

3 数据挖掘领域中常用的聚类算法

为了找到一个效率高且通用性强的聚类算法,人们从不同角度提出了数十种聚类算法。在数据挖掘中,常用到的有:BIRCH 算法^[2],DBSCAN 算法^[3],CURE 算法^[4], K -prototypes 方法,CLARANS 算法,CLIQUE 算法,Wave Cluster 算法等^[5~7]。

3.1 BIRCH 算法

即平衡迭代消减算法,是一种综合优化的多阶段聚类

技术,该算法的聚类特征树是一个具有两个参数分枝因子 B 和类直径 T 的高度平衡树。它的核心是采用了一个三元组的聚类特征树(CF 树)汇总了一个簇的有关信息,从而使一个簇的表示可以用对应的聚类特征,而不必用具体的一组点表示。分枝因子规定了树的每个节点子女的最多个数,而类直径体现了对一类点的直径大小的限制,即这些点在多大范围内可以聚为一类,非叶子结点为它的子女中的最大关键字,可以根据这些关键字进行插入索引,它总结了其子女的信息。

聚类特征树可以动态构造,因此不要求所有数据读入内存,而可在外存上逐个读入数据项。新的数据项总是插入到树中与该数据距离最近的叶子中。如果插入后使得该叶子的直径大于类直径 T ,则把该叶子节点分裂。其它叶子结点也需要检查是否超过分枝因子来判断其分裂与否,直至该数据插入到叶子中,并且满足不超过类直径,而每个非叶子节点的子女个数不大于分枝因子。可以通过改变类直径大小,修改特征树大小来控制其所占内存容量。

BIRCH 算法通过一次扫描就可以进行较好的聚类,由此可见,该算法适合于大型数据库。对于给定的 M 兆内存空间,其空间复杂度为 $O(M)$,时间复杂度为 $O[dNB \ln B(M/P)]$,其中 d 为维数, N 为节点数, P 为内存页的大小, B 为由 P 决定的分枝因子, I/O 花费与数据库成线性关系。但 BIRCH 算法只适用于类的分布呈凸形及球形等情况,并且由于 BIRCH 算法需提供正确的聚类个数和簇直径 T 的限制,对不可视的高维数据是不行的。

3.2 DBSCAN 算法

DBSCAN 算法即基于密度的聚类算法,该算法利用类的高密度连通性可以快速发现任意形状的类。其基本思想是:对于一个类中的每个对象,在其给定半径的邻域中包含的对象不能少于某一给定的最小数目。在 DBSCAN 中,发现一类的过程是基于这样的事实:一个类能够被其中的任意一个核心对象所确定。为了发现一个类,DBSCAN 先从对象集 D 中找到任意一对象 P ,并查找 D 中关于 R 和 P_{min} 的从 P 密度可达的所有对象(其中 R 为半径, P_{min} 为最小对象数)。如果 P 是核心对象,也就是说,半径为 R 的 P 的领域中包含的对象不少于 P_{min} ,则根据算法,可以找到一个关于参数 R 和 P_{min} 的类,如果 P 是一个边界点,则半径为 R 的 P 领域包含的对象数小于 P_{min} ,则没有对象从 P 密度可达, P 被暂时标注为噪声点,然后,DBSCAN 处理数据库 D 中的下一个对象。

为了有效地执行区域查询,DBSCAN 算法使用了空间查询中 R^+ -树结构。在进行聚类前,必须建立针对所有数据的 R^+ -树。另外,DBSCAN 要求用户指定一个全局参数 R (为了减少计算,预先确定参数 P_{min}),为了确定 R 值,DBSCAN 计算任意对象与它的第 k 个最临近的对象之间的距离。然后,根据求得距离由小到大进行排序,并给出排序后的图,称做 k -dist 图。 k -dist 图中的横坐标表

示数据对象与它的第 k 个最近的对象间的距离;纵坐标则为对应于某一 $k - \text{dist}$ 距离值的数据对象的个数。 R^+ 树的建立、 $k - \text{dist}$ 图的绘制是非常消耗时间的过程。此外,为了得到较好的聚类结果,用户必须根据 $k - \text{dist}$ 图,通过试探选定一个比较合适的 $k - \text{dist}$ 值,即 R 值。再就是,DBSCAN 不进行任何的预处理而直接对整个库进行聚类操作,当数据库非常大时,就必须有大内存量支持,I/O 消耗也非常大。其时间复杂度为 $O(N \log N)$ (N 为数据库中对象数目),聚类过程的大部分时间用在区域查询操作上。

3.3 CURE 算法

CURE 算法即使用代表点的聚类算法,该算法首先把每个数据点看成一类,然后再合并距离最近的类直至聚类个数为所要求的个数为止。它对传统聚类方法中类的表示方法进行了改进,回避了用所有点或简单地用中心和半径这样单一条件来表示一个类,而是从每个类中抽取固定数量、分布较好的点作为描述此类的代表点,并将这些点乘以一个适当的收缩因子,使它们更靠近类的中心点。

将一个类用多个代表点来表示,就使得聚类的外延可以向非球形的形状扩展,从而可调整聚类的形状以表达那些非球形的类。另外,收缩因子的使用减小了噪音对聚类的影响。同时,CURE 算法采用随机抽样与分割相结合的办法来提高算法的空间和时间效率,并且在算法中用了堆和 K_d 树结构来提高算法效率,对大型数据库有良好的伸缩性。

3.4 K-prototypes 算法

该算法结合 $K - \text{means}$ 方法和根据 $K - \text{means}$ 方法改进的能够处理符号属性的 $K - \text{modes}$ 方法。同 $K - \text{means}$ 方法相比,该方法能够处理符号属性。数值类型用欧式距离测量相似性,符号类型用海明距离测量相似性,将两部分综合起来构造评价函数,是一种有约束优化的聚类方法。

3.5 CLARANS 算法

CLARANS 算法即随机搜索聚类算法,是一种分割而非分层的聚类方法。该算法将采样技术和 PAM 相结合,首先随机选择一个点作为当前点,然后随机检查它周围不超过参数 Maxneighbor 个的一些邻接点,假如找到一个比它更好的邻接点,则把它移入邻接点,否则把该点作为局部最小量。然后,再随机选择一个点来寻找另一个局部最小量,直到所找到的局部最小量数目达到用户要求为止。

该算法要求聚类的对象必须预先都调入内存,并且需扫描多次数据库,这对大型数据库而言,无论从时间复杂度还是空间复杂度而言,都是不太适用的。虽后来通过引入 R^+ 树结构对其性能进行改善,使之能够处理基于磁盘的大型数据库,但 R^+ 树的构造和维护代价太大,计算复杂度为 $O(n^2)$ 。虽然该算法对脏数据和异常数据不敏感,但对数据输入顺序异常敏感,且只能处理凸形或球形边界聚类。

3.6 CLIQUE 算法

CLIQUE 算法即自动子空间聚类算法,该算法是一种基于密度(关系)和网格(变换)的聚类方法,利用自顶向上方法求出各个子空间的聚类单元,主要用于找出高维数据空间中存在的低维聚类;如果一个 k 维单元是密集的,那么它的 $k - 1$ 维空间上的投影也是密集的。中心思想是:给定一个多维数据点的大集合,数据点在数据空间中通常不是均衡分布的。该算法区分空间中稀疏的和“拥挤的”区域(单元),发现数据集合的全局分布模式,如果一个单元中包含的数据点数超过了某个输入参数,则该单元是密集的。在 CLIQUE 算法中,簇定义为相连的密集单元的最大集合。

为了求出 d 维空间聚类,必须组合给出所有 $d - 1$ 维子空间的聚类,导致其算法的空间和时间效率都较低,而且要求用户输入两个参数:数据聚值空间等间隔距离 ξ 和密度阈值 τ 。这些数据与样本数据紧密相关,用户一般难以确定。但此算法对数据的不同顺序不敏感。

3.7 Wave Cluster 算法

该算法是一种基于多分辨率变换的聚类方法,它首先在数据空间上强加一个多维网格结构来汇总数据,然后采用一种小波变换来变换原特征空间,在变换后的空间找到聚类区域。由于小波变换的特性使该算法具有很多优点:计算复杂度为 $O(n)$;发现任意形状的簇;成功处理孤立点;对输入顺序不敏感;领域独立;可以处理多达 20 维的数据。

4 聚类算法的性能比较

基于上述的分析,下面对常用聚类算法的性能从适合的数据类型、发现的聚类形状、对“脏数据”的敏感性、对数据输入顺序的敏感性、可伸缩性、高维性和算法效率 7 个方面进行比较,如表 1 所示。

表 1 聚类算法性能比较

	适合的数据类型	发现的聚类形状	对脏数据的敏感性	对数据输入顺序的敏感性	可伸缩性	高维性	算法效率
BIRCH	数值	凸形或球形	不敏感	不太敏感	较高	较低	高
DESCAN	数值	任意形状	敏感	敏感	一般	一般	一般
CURE	数值	任意形状	不敏感	不太敏感	较高	一般	较高
K-prototypes	数值和符号	凸形或球形	敏感	一般	一般	较低	一般
CLARANS	数值	凸形或球形	不敏感	非常敏感	较低	较低	较低
CLIQUE	数值	凸形或球形	一般	不敏感	高	高	较低
Wave Cluster	数值	任意形状	不敏感	不敏感	高	高	高

由表 1 可得如下结论:

(1)从算法效率、对脏数据或异常数据的敏感性及数据输入顺序的敏感性考虑,BIRCH 和 Wave Cluster 算法较好;

(2)BIRCH, CURE, CLARANS 和 Wave Cluster 对脏数据或异常数据具不敏感性;

(3)只有 K-prototypes 能处理数值和符号的数据;

(4)DBSCAN, CURE 和 Wave Cluster 能发现任意形状

(下转第 25 页)

则 $\Phi(d(t))$ 可表示为:

$$\Phi(d(t)) = \Phi d(t) \quad (14)$$

将式(14)代入式(13),得:

$$\mathbf{A}^T(t+1) = \mathbf{A}^T(t) + \eta(t)(\mathbf{y}(t)\mathbf{b}(t)^T - \mathbf{L}T[\mathbf{y}(t)\mathbf{y}(t)^T]\mathbf{A}^T(t)) \quad (15)$$

由式(14)经过分解可得:

$$a_{ij}(t+1) = \begin{cases} a_{ij}(t) + \eta(t)y_i(t) - \eta(t)y_i(t) \cdot \sum_{k=1}^i a_{kj}(t)y_k(t), & b_j = 1 \\ a_{ij}(t) - \eta(t)y_i(t) \sum_{k=1}^i a_{kj}(t) \cdot y_k(t), & b_j \neq 1 \end{cases} \quad (16)$$

$$\text{其中 } y_i(t) = \sum_{r=1}^l a_{ir}(t)\Phi(d_r)\Phi(d(t)) = \sum_{r=1}^l a_{ir}(t)k(d_r, d(t))$$

3 实验结果及分析

在实验室里,利用了车牌牌照识别系统的前期工作得到大量的像素的字符图像,为了使算法具有较强的泛化性和实用性,图像都是在现场不同环境中获取的。文中只对车牌中的字母进行训练和识别。分别在各类中选取 30 个样本作为训练样本,再分别选取 10 个样本作为测试样本。

实验的目的是为了论证基于 GHA 的 KPCA 特征提取在 SVM 字符识别中的作用,以及此算法与 KPCA 特征提取在性能上的不同。分别采用无特征提取、KPCA 特征提取以及基于 GHA 的 KPCA 特征提取进行实验,获得各自的识别率和训练速度,其中软件的编写使用 VC++ , SVM 程序部分改写了 LIBSVM 工具箱中的函数,表 1 给出了统一提取前 50 个特征的实验结果。

从实验结果可以看出,KHA 虽然消耗更长的时间来进行学习,但它利用自适应迭代的学习方法无需直接计算核矩阵、无需寻求矩阵的特征值和特征向量,需要更少的

存储空间并可在线学习,尤其在使用大样本的应用实例时,更能显示此方法的优势。

表 1 实验数据

	SVM	
	识别率(%)	训练速度(m)
无特征	89.9	2.52
KPCA	94.3	5.31
KHA	93.8	6.35

4 总结

文中提出了一种基于 GHA 的核成分分析的方法,它是利用核函数的方式求取数据空间中的非线性特征,在计算过程中,利用 GHA 求解 KPCA 问题,使 KPCA 特征提取的方式在大样本分类问题中成为可能,实验结果表明了此算法的可行性。但是在实际应用中,还存在着训练速度慢和选取特征数固定的问题,下一步的研究中将着力解决这些问题。

参考文献:

- [1] Scholkopf B, Smola A, Muller K. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation, 1999, 10(5):1299-1319.
- [2] Kim K I, Franz M O, Scholkopf B. Kernel hebbian algorithm for iterative kernel principal component analysis[R]. [s. l.]: Planck Institute for Biological Cybernetics, 2003.
- [3] Oja E. A simplified neuron model as a principal component analyzer[J]. Journal of Mathematical Biology, 1982, 15: 267-273.
- [4] Sanger T D. Optimal unsupervised learning in a single-layer linear feedforward neural network[J]. Neural networks, 1989 (2): 459-473.
- [5] Cao L J, Chua K S, Chong W K, et al. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine[J]. Neurocomputing, 2003, 55: 321-336.

(上接第 22 页)

的类。

由于每个方法都有其优缺点和不同的适用领域,在数据挖掘中,用户应该根据实际需要选择恰当的聚类算法。

参考文献:

- [1] Han Jiawei, Kamber M. Data Mining Concepts and Techniques[M]. 北京:高等教育出版社, 2001. 223-262.
- [2] Zhang T. BIRCH: An efficient data clustering method for very large databases[A]. Proc. of the ACM SIGMOD Int'l Conf on Management of Data[C]. Montreal: ACM press, 1996. 73-84.
- [3] Enter M. A density-based algorithm for discovering clusters in large spatial databases with noise[A]. In Proc of 2nd Int'l Conf on Knowledge Discovering in Databases and Data Mining KDD-96[C]. Portland: AAAI Press, 1996.
- [4] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases[A]. In Proc of the ACM SIGMOD Int'l Conf on Management of Data[C]. Seattle: ACM Press, 1998. 73-84.
- [5] GEHRKE J, AGRAWAL R, GUNOPUL O. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications[J]. ACM SIGMOD, 1998, 72(2): 94-105.
- [6] BETUR V, DASARAEH Y. Data Mining and knowledge Discovery: Theory Tool, and Technology II[A]. Orlando, florida 2000 SPIE-The International Society for Optical Engineering [C][s. l.]: [s. n.], 2000. 259-264.
- [7] 王实, 高文. 数据挖掘中的聚类方法[J]. 计算机科学, 2000(4): 42-45.