一种基于子空间聚类的图像分层索引方法

许宏丽 须 德 林恩爱
(计算机与信息技术学院,北京交通大学,北京 100044)

摘 要 随着多媒体技术的发展,许多领域产生大量的高维数据集。为了有效地检索这些高维数据,高维索引成为人们研究的热点。聚类树是一种有效地支持高维数据检索的索引结构。提出了一种基于子空间聚类的聚类树结构,该索引结构基于一种改进的 CLIQUE 聚类算法,利用小波变换的多尺度特性对图像特征分布曲线进行不同尺度的小波变换,去除一些小的分类和可能的噪声干扰,从而得到不同粒度下的层次聚类。在层次聚类的基础上,建立起分层索引结构。由于改进的聚类算法使用爬山法确定子空间聚类,因而有效地避免了用户参数的定义。实验结果证明,该方法在不需要用户设定聚类参数下能够进行有效聚类,在不同尺度下构建的聚类结构能够有效地组织图像关系,大大提高图像的检索效率。

关键词 基于内容图像检索 高维数据索引 子空间聚类 聚类树
中图法分类号:TP301.6 **文献标识码**:A **文章编号**: 1006-8961(2009)01-0142-06

An Approach of Hierarchical Image Index Based on Subspace Cluster

XU Hong-li, XU De, LIN En-ai
(*School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044)

Abstract Nowadays large volumes of data with high dimensionality are being generated in many fields. Many approaches have been proposed to index high-dimensional datasets for efficient querying. ClusterTree is a new indexing approach representing clusters generated by any existing clustering approach. Lots of clustering algorithms have been developed, and in most of them some parameters should be determined manually. The authors propose a new subspace-cluster indexing algorithm, which based on the improved CLIQUE and avoids bias on any parameters caused by user. Using multi-resolution property of wavelet transforms to reprocess the distribution curve of samples, the proposed approach can cluster at different resolution and remain the relation between these clusters to construct hierarchical index. The results of the experiment confirm that the subspace-cluster algorithm is very applicable and efficient, and show that this hierarchical indexing structure does well in the content-based image retrieval.

Keywords content-based image retrieval, high-dimensional data index, subspace cluster, cluster tree

1 引 言

随着大容量存储器以及数字信息采集设备的普及和 internet 技术的蓬勃发展,多媒体等高维数据源大量涌现。为了有效地索引高维数据集,研究人员提出许多方法。较流行的类树索引方法包括:基于数据划分和基于空间划分的方法。基于数据划分的索引结构有 R-tree, R*-tree, SS-tree, SS +-tree, SR-tree,这种索引结构的节点使用最小矩形(R-tree, R*-tree)或超球(SS-tree, SS +-tree)包络,其缺点之一是不同节点包络的重叠。空间划分法递归划分数据空间成为不重叠的子空间,其代表结构有 K-D-B 树和金字塔树,而这种方法是基于数据均匀分布,同时金字塔树不能保证一个节点的数据总是近邻的。在这些方法中,聚类树[1]提出了使用聚类构建有

基金项目:国家自然科学基金项目(60602030)
收稿日期:2007-03-01;**改回日期**:2007-06-12

第一作者简介:许宏丽(1963~),女。副教授。北京交通大学计算机与信息技术学院计算机应用技术专业博士研究生。主要研究领域为数据库、图像处理、基于语义的视觉信息检索。E-mail:hlxu@bjtu.edu.cn

效的索引结构的方法。聚类是一种发现数据集分布模式的分析技术。聚类方法根据数据间的相似度测量将数据分组,使得每一组的数据点相似度大于不同组的数据点。每一组就是一个聚类,一个聚类树就是基于聚类信息从粗到细地组织数据,并构成数据集的分层聚类描述。本文在研究高维聚类方法的基础上提出一种基于小波变换的多尺度子空间聚类算法。该算法有效地支持聚类树的生成,构建分层索引结构,提高高维数据检索效率。

2　基于小波变换的 CLIMB 算法

2.1　CLIMB 聚类算法

目前在文献中存在大量的聚类算法,如基于划分的方法、层次的方法(BIRCH[2],CURE[3])、基于密度的方法(DBSCAN[4],DENCLUE[5])、基于网格的方法(STING[6],WaveCluster[7])等。CLIQUE[8]子空间聚类算法是综合了基于密度和基于网格的聚类方法,对于大型数据库中的高维数据聚类很有效。CLIQUE 算法要求用户输入数据聚类空间等间隔距离 ξ 和密度阈值 τ 两个参数。这些参数与样本数据紧密相关,用户一般难以确定。

CLIMB(clustering algorithm based on subspace)是基于 CLIQUE 的一种实用且高效的聚类算法[9]。CLIMB 聚类算法的基本思想是,对于每一个高维向量样本,分别在每一维上进行投影,并且将每一维等分成一定大小的间隔,统计每一个间隔内样本个数得到样本分布曲线。分布曲线每两个相邻的波谷形成一个“山峰”,每个“山峰”对应一个聚类簇,寻找“山峰”的过程使用爬山法。在第 j 维上进行聚类时,假设在第 $j-1$ 维上聚成了 k 类,这样在第 j 维上,对这 k 类的每一类都进行寻找山峰的过程,并将该类样本聚成子类,最终形成对样本的自顶向下、非对称的分类树。如果在第 j 维上进行聚类时,某个已有的类的样本分布曲线只有一个山峰,这意味着只有一个类簇,这种情况下,则在第 j 维上不做任何处理。

CLIMB 算法不需要用户事先指定参数,很好地解决了“用于决定输入参数的领域知识最小化”的问题。与 CLIQUE 算法相比,CLIMB 算法有以下特点:(1)CLIMB 算法无需事先指定参数值,解决了人为指定参数的困难。(2)CLIQUE、DENCLUE 等算法对所有的类别都使用一个相同的密度阈值,没有对不同的密度的聚类区别对待,而 CLIMB 算法则不存在这一问题。(3)CLIMB 算法采用自顶向下的方式,先将整个样本集作为一个类,在每个坐标维上,只通过相邻单元之间的比较来寻找对每个已有类的一个划分,最终生成对输入样本的非对称层次聚类。(4)CLIQUE 算法需要有判断两个聚类是否应该合并的运算,而 CLIMB 算法在识别分类的同时,已经找到了聚类之间的划分,因而 CLIQUE 算法的复杂度大于 CLIMB 算法的复杂度。

2.2　基于小波变换的 CLIMB 聚类算法

在 CLIMB 聚类算法中,为了消除样本分布曲线的噪声干扰问题,使用了插值方法对样本分布曲线进行光滑处理。改进的 CLIMB 算法利用小波多尺度分析特性对样本分布曲线进行处理,去掉了不同程度的细节信息。变换尺度越大,去掉的高频成分越多,得到的样本分类越粗糙。通过选择不同的小波变换尺度即可得到不同粗糙度的类簇,以形成类簇之间的层次关系,从而构造不同尺度下聚类的层次关系形成分层索引结构。

用 $\boldsymbol{S}^d$ 来表示 d 维空间。用 $A_d,\cdots,A_2,A_1$ 来表示 $\boldsymbol{S}^d$ 空间的 d 个维。将 d 维特征空间中的每一维划分为 n 个间隔,令 $O_i=\{f_{i1},f_{i2},\cdots,f_{id}\}$ 为对象 $\boldsymbol{O}_i$ 在 $\boldsymbol{S}^d$ 空间中的特征矢量,$\boldsymbol{D}_j=(v_{j1},v_{j2},\cdots,v_{jd})$ 表示第 j 维的划分,其中,$v_{ji}(1\leqslant i\leqslant n)$ 表示第 j 维的第 i 个间隔,$g(v_{ji})$ 是第 j 维的第 i 个间隔中的对象个数,第 j 维的分布曲线可表示为 $y_j=F(g(v_{ji}))$。则基于小波的 CLIMB 算法 Wavelet-CLIMB 如下:

输入:样本数据 $\boldsymbol{O}_i=\{f_{i1},f_{i2},\cdots,f_{id}\}$

输出:子空间聚类 $\boldsymbol{C}=\{C_1,C_2,\cdots,C_m\}$

(1)计算各维数据密度分布曲线 $g(v_{ji})$

(2)对分布曲线进行小波变换;

(3)对第 j 维($1\leqslant j\leqslant d$),比较 $g(v_{ji})$ 和 $g(v_{ji+1})$。

如果 $g(v_{ji})<g(v_{ji+1})$,则处于爬山阶段;

如果 $g(v_{ji})>g(v_{ji+1})$,则处于下山阶段;

其中,$1\leqslant i\leqslant n$。

(4)如果 $g(v_{ji})>g(v_{ji-1}),g(v_{ji})>g(v_{ji+1})$,记录这个波峰为 i;

如果 $g(v_{ji})<g(v_{ji-1}),g(v_{ji})<g(v_{ji+1})$,记录这个波谷为 l;

(5)对应两个波谷间的对象为原空间的一个聚类;

(6)当 $j \leqslant d$ 时,$j = j + 1$,返回步骤 1;

(7)当 $j > d$ 时,结束。

2.3 坐标离散间隔

计算密度分布曲线需要首先确定坐标离散间隔,然后统计每个间隔之内的样本数,从而得到样本分布曲线。坐标离散间隔的划分将决定算法的性能,间隔的大小应该根据数据空间样本的个数而定。如果样本数很少而间隔很大,将丢失许多聚类。同样,如果样本数很多而间隔很小,将降低算法运算速度。

MAFIA[10]提出一种自适应间隔划分方法。该方法基于数据分布划分,获得的包络比固定尺寸的划分获得更精确的聚类边界。MAFIA 方法采用默认值设定小窗口尺度,同时要定义合并阈值。在 MAFIA[10]的实验中,窗口尺寸为 5,合并阈值为 20%。在我们的实验中,分别选取窗口大小为 5 和 10,合并阈值分别为 10% 和 20%。实验结果表明,当窗口大小为 10,合并阈值为 10% 时,结果最好。

在文献[9]中提出一种数据空间划分方法,算法假设单个间隔内属于同类的平均样本数为 k,共有 N 个待聚类的样本,那么随着 N 的增大,为了能够有效地区分不同类的样本,确保聚类过程收敛,k 必须满足以下 3 式条件的非降序正整数序列:

$$\lim_{N\to\infty} k = \infty \tag{1}$$

$$\lim_{N\to\infty} N^{-1}k = 0 \tag{2}$$

$$\lim_{N\to\infty} (\log N)^{-1}k = \infty \tag{3}$$

函数关系 $k \propto \sqrt{N}$ 是满足上述条件的一个特解,所以可取 $k = \sqrt{N}$,从而得到等分间隔数 $\xi = N/k = \sqrt{N}$。

文献[9]对这种取等分间隔的方法进行了数学上的理论证明,并没有给出实验证明。本文通过实验验证这种等分间隔方法的可行性。从式(1)~式(3)可以得出推论,等分间隔数的取值必须满足一个条件,那就是在同分布下,样本发生变化时,样本分布曲线应该基本保持不变,也就是聚类簇的个数要基本保持不变。因此,只要改变样本的个数,观察不同的样本个数与所得到的聚类个数的变化关系就能看出这种等分间隔的取法是否合理。基于这种指导思想,选取了具有近似分布的 1 000 幅、2 000 幅、3 000 幅、4 000 幅、5 000 幅图片的 RGB 累积直方图作为样本数据进行算法验证。由于验证这种结论是否成立只需要 1 维的数据就可以,为减少计算过程的累积误差,只选取样本的第 1 维特征作进行聚类验,结果如表 1 所示。

表 1 离散间隔取 $\sqrt{N}$ 时,不同样本数下的聚类数

Tab.1 When $\xi = \sqrt{N}$, the number of clustering data in different size of the dataset

样本数	1 000	2 000	3 000	4 000	5 000
聚类簇数	9	9	10	11	11

从表 1 可以看出,当样本数由 1 000 个翻倍增加至 2 000 个后,聚类个数并没有发生变化,而样本数增加至 5 000 个后,聚类数也只有略微的变化,实验的结果符合理论推想。因此,等分间隔数取 $\sqrt{N}$,也就是样本数的开平方是合理的。

3 基于子空间聚类的分层索引算法

3.1 子空间聚类树 SSClusterTree

SSClusterTree 是对数据集的分层聚类表示,它有两种节点:内部节点和叶节点。

内部节点的定义为

$$Node\_id, CL_{id}, CH_{id}, CC_{id}, \gamma, (CP_1, CP_2, \cdots, CP_\gamma)$$

其中,$Node\_id$ 为节点标识,γ 为子节点数。一个子节点表示一个聚类的子类。CP_i 为指向子节点的指针,CL_{id} 和 CH_{id} 为节点 $Node\_id$ 上、下边界坐标。CC_{id} 是类中心,同父子节点按照 CC_{id} 排序。

叶节点的定义为

$$Leaf: Leaf\_id, \kappa, (ADR_1, ADR_2, \cdots, ADR_\kappa)$$

其中,κ 为叶节点中含有的数据个数。ADR_i 是这个数据点的地址。

利用不同尺度下分布曲线的小波变换之间形成的层次关系,构造至上而下的层次聚类索引结构 SSClusterTree。

输入:数据集 $\{O_i, 1 \leqslant i \leqslant N\}$,$\boldsymbol{O}_i = (f_{i1}, f_{i2}, \cdots, f_{id})$

输出:SSClusterTree

(1)初始化样本,从样本集中抽取特征属性集 $\boldsymbol{O}_i = (f_{i1}, f_{i2}, \cdots, f_{id})$;

(2)计算坐标间隔划分常数 ξ;

(3)对特征属性集中的每个属性 i,计算数据分布 $g(v_{ij})$;

(4)对 $g(v_{ij})$ 进行小波变换,记录变换尺度 λ;

(5)计算每一尺度下的聚类

使用 Wavelet-CLIMB 计算子空间聚类,

记录聚类子空间的坐标,并计算类中心;

(6)重复步骤 5 直到尺度 $\lambda = 0$;

(7)输出 SSClusterTree。

3.2　基于 SSClusterTree 索引结构的实现

为了实现分层索引,需要记录下每张图片在每一次小波变换后进行聚类后的聚类号。为了能实现层与层之间的父子联系,需要记录下每次小波变换后每一类簇所对应的高层次小波变换的聚类号,也就是父节点的编号。同时,为了判断待检索的图片属于哪个类范围,还需要记录下每一个聚类的图片的下坐标和上坐标以及这个类的类中心。在本实验中采用数据库表来存储索引结构,表 2 为保存样例图片的特征和子空间聚类编号,表 3 为 SSClusterTree 算法构建的索引结构。

表 2　样例图片的特征和子空间聚类编号表 images 结构

Tab. 2　The file structure about the features and clusters of sample pictures

	图片文件名	RGB 累积直方图	0 次小波变换的类编号	1 次小波变换的类编号	……	*n* 次小波变换的类编号
字段名	*Image Name*	*RgbCumHist*	*Level0Id*	*Level1Id*	……	*Level3Id*
数据类型	文本	OLE 对象	数字	数字	……	数字

表 3　SSClusterTree 算法树节点的结构 ClusterRange

Tab. 3　The data structure of SSClusterTree

	聚类编号	聚类层次编号	父聚类编号	类的下坐标	类的上坐标	类中心
字段名	*ClusterId*	*LevelId*	*ParentId*	*MinOfCluster*	*MaxOfCluster*	*CenterOfCluser*
数据类型	数字	数字	数字	OLE 对象	OLE 对象	OLE 对象

images 表存储的是元数据,每张图片对应其中的一条记录,每条记录包括图片名称、特征以及零次、1 次、2 次、3 次小波变换聚类操作后的每张图片的聚类层次编号(*Level0Id*、*Level1Id*、*Level2Id* 和 *Level3Id*)。ClusterRange 表记录了每个子聚类的属性,包括类编号、类层次(也就是小波变换层次)、类的下坐标、类的上坐标以及类中心等字段。而 *ParentId* 字段(也就是父类的编号)可以从 images 表中取得。类的下坐标、类的上坐标和类中心作为检索的判断条件。

3.3　图像检索过程

建立索引结构后,检索操作只在 ClusterRange 这个表中进行。检索过程如下:

(1)读入待检索的图片,提取其图像特征,放入一个数组 ***m_pFeaQuery*** 中。

(2)取最高聚类层次数 $LevelId = \max(LevelId)$

(3)在 ClusterRange 表中,如果 *m_pFeaQuery* 介于 *MinOfCluster* 和 *MaxOfCluster* 之间,表明待检索图片在这个范围内,返回该聚类号;否则,比较下一个同父子节点。

(4)执行 $LevelId - 1$,如果 $LevelId - 1 \leqslant 0$,返回步骤 3;否则,执行 5。

(5)返回表 images 表中对应聚类号的所有图片。

4　实验结果与性能分析

实验数据来自 Corel Photo Gallery 图像库,图 1 为部分图片样例。系统开发平台是 Windows XP、VC + +6.0 和 MS Access 2003。硬件条件是 AMD Duron(tm) 995MHz 的 CPU,512M 的内存。

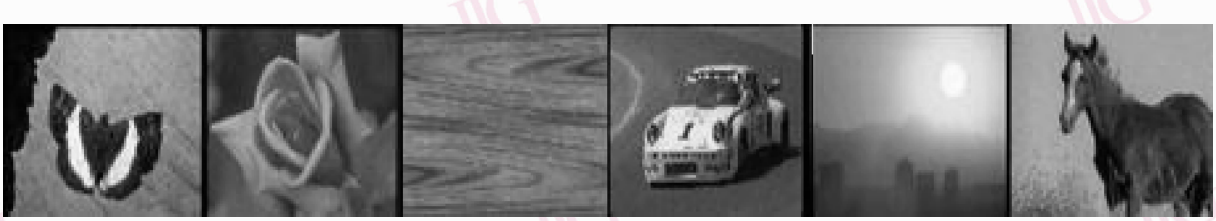

图 1　实验图像库样例图片

Fig. 1　The samples in the experiment image database

4.1 不同尺度下的 Wavelet-CLIMB 聚类

首先,选择花朵、蝴蝶和纹理类图片 1 006 幅,使用 Wavelet-CLIMB 进行聚类,表 4 列出在不同尺度下的聚类结果。其中 RGB 为 RGB 颜色直方图特征,Coarse wavelet 为图像小波近似特征,Co-occurrence 为图像共生矩阵。

表 4 不同特征下基于 Haar 小波的 Wavelet-CLIMB 聚类数

Tab. 4 Clustering image data at different features using Wavelet-CLIMB based on Haar wavelet

图像特征	Haar 小波变换尺度			
	0	1	2	3
RGB	75	7	5	3
Coarse wavelet	9	6	4	1
Co-occurrence	12	9	6	2

其次,分别选择花朵、蝴蝶和纹理类图片各 100 幅,图片特征选择 RGB 颜色累积直方图形成 192 维的空间向量。执行基于 Wavelet-CLIMB 的分层聚类,图 2 列出不同尺度下子聚类的关系。

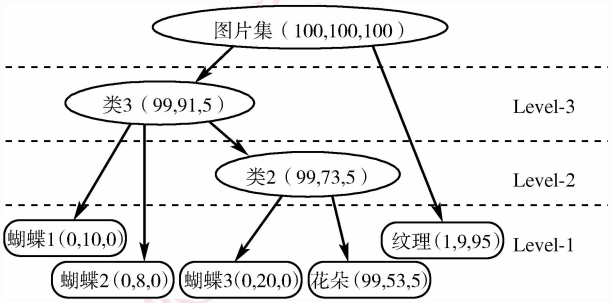

图 2 Wavelet-CLIMB 分层聚类算法。样例图像集的顺序为花朵,蝴蝶,纹理

Fig. 2 An example of the Wavelet-CLIMB. The order of sample images is flower, butterfly, and texture

最后,在不同大小的数据集上执行 Wavelet-CLIMB 算法,表 5 列出了不同次数小波变换后的聚类个数实验结果证明了基于子空间分层聚类算法的有效性。

表 5 不同大小数据集下基于 Haar 小波的 Wavelet-CLIMB 聚类数

Tab. 5 The number of clustering for different size dataset

数据集图像个数	0 次	1 次	2 次	3 次
1 000	78	5	3	1
2 000	82	5	3	1
3 000	86	4	3	1
4 000	128	7	2	1
5 000	151	4	2	1

4.2 算法复杂度比较

CLIQUE 聚类算法是综合了密度和网格聚类算法的典型代表,Wavelet-CLIMB 算法是 CLIQUE 的改进算法,本文把这两种算法都应用在前述的图像数据库中,对图像进行聚类操作。图 3 是两种算法性能比较的实验结果。

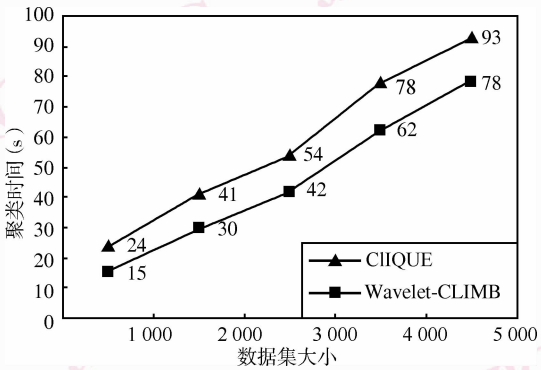

图 3 Wavelet-CLIMB 与 CLIQUE 聚类时间比较

Fig. 3 The Comparison of the Wavelet-CLIMB with CLIQUE

从图 3 可以看出,随着样本图片个数的增加,Wavelet-CLIMB 算法的聚类时间几乎呈线性变化,这跟它的算法复杂度 $O(md)$ (m 是输入的样本数,d 是样本空间的维数)相一致。CLIQUE 算法在聚类的过程中需要增加根据阈值判断是否合并相邻间隔的运算,增加了算法的额外时间和空间开销,虽然其算法复杂度为 $O(C^d + md)$ (C 为常数),与样本个数 m 也呈线性增长,但是多了 C^d 部分,因此,CLIQUE 算法的聚类时间明显大于 Wavelet-CLIMB 算法的聚类时间。当维数增加时,Wavelet-CLIMB 算法的聚类时间仍将呈线性增长,而 CLIQUE 算法的聚类时间将随 C^d 呈指数增长,在这种情况下,Wavelet-CLIMB 将显示出它的优势。由此可见,Wavelet-CLIMB 算法比 CLIQUE 算法具有更好的扩展性能,而且效率更高。

4.3 检索效率的比较

SSClusterTree 层次索引算法与 R-Tree 算法以及顺序检索算法的检索性能实验比较如图 4 所示。

从图 4 中可以看出,层次索引结构算法在检索效率上要远远优于顺序检索,比 R-Tree 算法也有一定的提高。顺序检索算法由于要与图片库中的每张图片都进行比较,因此检索的时间与图片的个数呈线性的递增关系。而基于聚类的层次索引结构在检索前已经对图片进行了前期的聚类预处理,形成索

引结构,并且预先存入到了数据库系统中。检索图片时,只需要在索引结构表中进行,而这个表是按聚类为单位来进行存储的,对于 5 000 幅的图片来说只有 150 多条的记录,而且这 150 多条记录也不需要顺序检索,是按层次来进行检索的,平均检索次数约为 30 次左右,大大缩小了时间复杂度,这是层次索引结构算法在检索效率上要优于顺序检索的根本原因。

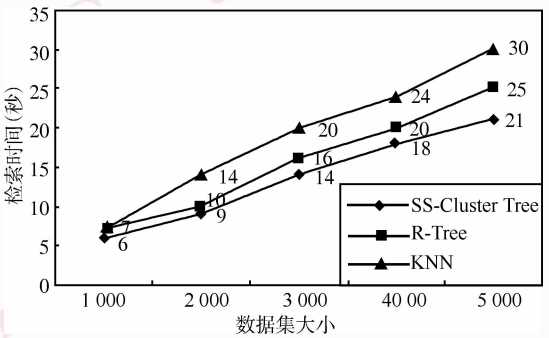

图 4　SSClusterTree 与 R-Tree 和顺序检索算法的耗时比较

Fig. 4　The Comparison of the SSClusterTree retrieval time with R-Tree and KNN

从查全率来说,顺序检索能保证 100% 的查全率,而基于聚类的分层索引结构由于在聚类操作时,会有个别相关图片落入别的类之中,因此会导致某些边缘图片不能被检索到,影响了查全率。但是,这并不会影响到整体的检索效果。综合考虑到检索效率,基于聚类的分层索引结构有着很强的优势,特别是在海量图像数据库的检索方面,分层索引结构所体现出来的效率更是远远优于其他算法。

5　结　论

本文提出了一种基于子空间聚类树的索引方法。此算法利用小波变换对样本分布曲线进行平滑处理,保证处理噪声数据的能力;数据分布曲线的使用使得算法具有对于输入样本的顺序不敏感的性质;由于 CLIMB 算法不需要人为指定聚类参数,使得决定输入参数的领域知识最小化。图像数据的聚类实验结果表明,随着输入样本的增多,算法聚类效果更好。使用该方法对图像数据库进行预处理,并建立索引,将大大提高检索的效率。

下一步的工作将对算法的时间复杂度做进一步的研究,同时研究当图像数据库插入或删除图像数据时,聚类结果将如何变化等问题,从而进一步完善聚类算法;深入研究相应的索引结构特性,实现图像数据库基于内容的索引结构的建立与应用。

参考文献(References)

1　Yu Dantong, Zhang Aidong. ClusterTree: integration of cluster representation and nearest-neighbor search for large data sets with high dimensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, **15**(5):1316-1337.

2　Zhang Tian, Raghu Ramakrishnan, Miron Livny. BIRCH: An efficient data clustering method for very large databases[A]. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data[C], Montreal, Canada, 1996: 103-114.

3　Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for large databases[A]. In: Proceedings of the ACM SIGMOD International Conference on Management of Data[C], Seattle, USA, 1998: 73-84.

4　Ester M, Kriegel H, Sander J, *et al*. A density-based algorithm for discovering clusters in large spatial databases with noise[A]. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining 1996 [C], Portland, USA, 1996: 226-231.

5　Hinneburg A, Keim D. An efficient approach to clustering in large multimedia databases with noise[A]. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining [C], New York, USA, 1998: 58-65.

6　Wang W, Yang J, Muntz R R. STING: A statistical information grid approach to spatial data mining[A]. In: Proceedings of the 23rd International Conference on Very Large Data Bases[C], Athens, Greece, 1997: 186-195.

7　Sheikholeslami G, Chatterjee S, Zhang A D. WaveCluster: A multi-resolution clustering approach for very large spatial databases[A]. In: Proceedings of the 24th International Conference on Very Large Data Bases[C], New York, USA, 1998: 428-439.

8　Rakesh A, Johanners G, Dimitrios G, Prabhakar R. Automatic subspace clustering of high dimensional data for data mining applications[A]. In: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data[C], Minneapolis, USA, 1994:94-105.

9　Wang Jian-hui, Shen Zhan, Hu Yun-fa. An Applicable and Efficient Clustering Algorithm[J]. Journal of Software, 2004, **15**(5):697-705. [王建会,申展,胡运发. 一种实用高效的聚类算法[J]. 软件学报. 2004, **15**(5): 697-705.]

10　Goil S, Nagesh H, Choudhary A. MAFIA: Efficient and Scalable Subspace Clustering Clustering for Very Large Data Sets[R]. TR-9906-010, Illinois, USA: Northwestern University, 1999.