文章编号:1672-3961(2010)05-0137-04

基于 BIRCH 的木材缺陷识别

吴东洋,业宁
(南京林业大学信息技术学院, 江苏 南京 210037)

摘要:提出了一种新的基于分层的平衡迭代归约及聚类(balance iterative reducing and clustering using hierarchies, BIRCH)算法的木材缺陷识别方法,讨论了关于分支因子(B,L)、阈值 T 的选取及非缺陷类判别问题。该方法通过在一定阈值内构建 CF 树,产生初始聚类,对初始聚类进行非缺陷类判别,自动识别木材缺陷类及位置并标记。实验结果表明,该算法能有效地进行木材缺陷识别,平均识别查准率约为 86.3%,平均识别查全率约为 90.1%。

关键词:分层的平衡迭代归约及聚类;聚类分析;木材缺陷
中图分类号:TP391 **文献标志码**:A

Wood defect recognition based on the BIRCH algorithm

WU Dong-yang, YE Ning
(School of Information Technology, Nanjing Forestry University, Nanjing 210037, China)

Abstract: A new method for wood defect recognition based on the BIRCH algorithm was proposed. The problems about branch factor (B, L), the selection of threshold T and the discrimination of non-defect class are discussed. To produce the initial clustering, distinguish non-defect class for the initial clustering, and automatically identify the location of the wood′s defects and mark it, a CF-tree within a certain threshold was built. The experimental results showed that this algorithm could efficiently identify the wood′s efficiently defects, the average defect precision ratio was about 86.3%, and the average defect recall ratio was about 90.1%.

Key words: balance iterative reducing and clustering using hierarchies; clustering method; wood defect

0 引言

在木板材加工生产中,木材缺陷的影响会降低木材强度,影响加工质量或外观。传统的木板材缺陷识别方法存在实验条件苛刻、设备成本高等问题。采用机器自动对木材缺陷检测与定位,可以降低人为识别过程中主观因素影响,提高识别效率和板材的利用率。提出了一种基于无监督聚类方法的木材缺陷识别,采用 BIRCH 算法,只需一次扫描数据集就可产生较高的聚类质量,讨论了分支因子 B 和 L、阈值 T 的选取及非缺陷类判别问题。实验取得了较好的识别效果,有效地提高了识别准确率。

1 背景知识

1.1 聚类概念和聚类过程

迄今为止,聚类还没有一个学术界公认的定义。这里给出 Everitt[1] 在 1974 年关于聚类的定义:一个类簇内的实体是相似的,不同类簇的实体是不相似的;一个类簇是测试空间中点的汇聚,同一类簇的任意两个点间的距离小于不同类簇的任意两个点间的距离;类簇可以描述为一个包含密度相对较高的点集的多维空间中的连通区域,它们借助包含密度相对较低的点集的区域与其他区域(类簇)相分离。

传统的聚类分析方法可以分为以下几类[2]:基

收稿日期:2010-04-02
基金项目:国家自然科学基金资助项目(60573024);江苏省自然科学基金资助项目(BK2009393)
作者简介:吴东洋(1978-),女,辽宁沈阳人,讲师,研究方向为数据挖掘. E-mail:eassun2000@sina.com.cn

于层次化的方法（hierarchical method）、基于划分式方法（partitioning method）、基于密度方法（desity-based method）、基于网格的方法（grid-based method）和基于模型的方法（model-based method）。所有的聚类方法都有各自的特点，目前普遍认为不存在某种方法适合解决所有的聚类问题。

1.2 BIRCH 算法思想

BIRCH[3-4] 算法最早在 1996 年由 Tian Zhang 提出的。该算法是一个较为有效的综合的层次聚类方法，特别适合大数据集，它将数据集首先以一种紧凑的压缩格式存放，直接在压缩集上进行聚类，单遍扫描数据集就可以生成较好的聚类，可选的另一遍或多遍扫描用于进一步改进聚类质量。

首先给出如下定义：

在一个聚类中的 N 个 d 维数据点：$\{\boldsymbol{X}_i\}$，其中 $i=1,2,\cdots,N$。

定义 1 半径 R：

$$R=\left[\frac{\sum_{i=1}^{N}(\boldsymbol{X}_i-\boldsymbol{X}_0)^2}{N}\right]^{\frac{1}{2}}, \tag{1}$$

半径代表了从一个聚类的数据点到质点的平均距离。

BIRCH 算法引入了两个概念：聚类特征和聚类特征树（CF 树）。

定义 2 聚类特征（CF）给定在一个聚类中 N 个 d 维数据点$\{X_1,X_2,\cdots,X_N\}$，聚类特征（CF）定义为一个三元组 CF = (N,LS,SS)，其中 N 是聚类中数据点的数量，LS 是 N 个数据点的线性和，即$\sum_{i=1}^{N}\boldsymbol{X}_i$，SS 是 N 个数据点的平方和，即$\sum_{i=1}^{N}\boldsymbol{X}_i^2$。其中线性和反映了聚类的质心，平方和反映了聚类的直径大小。

定理 CF 可加性定理 假设 $\mathrm{CF}_i=(N_i,\ \mathrm{LS}_i,\ \mathrm{SS}_i)$ 与 $\mathrm{CF}_j=(N_j,\ \mathrm{LS}_j,\ \mathrm{SS}_j)$分别为两个类的聚类特征，合并后新类的聚类特征为 $\mathrm{CF}_i+\mathrm{CF}_j=(N_i+N_j,\mathrm{LS}_i+\mathrm{LS}_j,\mathrm{SS}_i+\mathrm{SS}_j)$。即聚类特征具有可加性。

聚类特征树（CF 树）是一个有两个参数的高度平衡的树：分支因子（每个非叶节点最多含有 B 个分支条目，而每个叶节点最多含有 L 个条目）及阈值 T，它用来存储聚类特征。每一个含有最多 B 个非叶节点的形式为：$\{\mathrm{CF}_i,\mathrm{child}_i\}$，其中 $i=1,2,\cdots,B$，child_i 是一个指向它的第 i 个子节点的指针，而 CF_i 是该子节点所代表的所有子聚类组成的聚类。每个叶节点最多有 L 个条目，而每一个条目的形式为 $\{\mathrm{CF}_j\}$，其中，$j=1,2,\cdots,L$ 及 CF_j 是它的第 j 个子聚类的 CF。一个叶节点代表一个由它的各条目代表的子聚类组成的聚类。叶节点中的所有条目必须满足阈值 T 的要求，T 越大，树越小。CF 树结构见图 1。

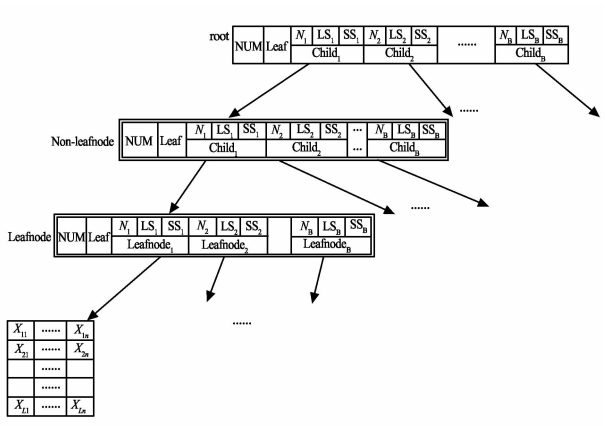


图 1 CF 树结构

Fig. 1 Structure of CF-tree

2 木材缺陷识别算法描述

2.1 CF 树的建立

BIRCH 算法的核心就是建立一颗 CF 树，下面给出将一个数据条目 K_i 插入到 CF 树的算法。

CF 树建立算法 b_tree_insert()：

首先指定该树的叶子节点阈值 T、分支因子 B 及 L。假定要插入数据条目 K_i。

（1）从根开始，依据欧式距离 D_0 沿着 CF 树向下递归寻找最接近的节点。

（2）判断途经节点：

（a）如果数据条目 K_i 到达节点的分支条目的数目 $N_t!=B$，则继续向下寻找；

（b）否则，调用 b_tree_spilt_child()算法分裂该节点。

（3）到达要插入的叶子节点 L_j，插入数据条目 K_i，判断该叶子节点中数据条目数 N_j 及半径 R_j。

（a）如果 $N_j<L$，同时该叶子节点半径 $R_j<T$，则数据条目 K_i 被该叶子节点吸收；

（b）否则，删除 K_i，调用 b_tree_spilt_leaf()算法分裂该叶子节点；

（c）依据欧式距离 D_0，重新寻找插入位置 m，将数据条目 K_i 插入。

i 如果 $R_m<T$，则转到（4）；

ii 如果 $R_m>T$，则提升 T，CF 树重建。

（4）读取新的数据条目 K_{i+1}，转到（1）。

分裂非叶节点算法 b_tree_spilt_child()，

开辟两个新的节点 t_1、t_2：

（1）计算当前节点 t 中距离最大的两点 t_a、t_b 分别作为新开辟节点 t_1、t_2 的中心点，根据欧式距离，将剩余节点分配到两个节点 t_1、t_2 中；

（2）将新节点 t_1、t_2 插入，删除原节点 t。

分裂叶节点算法 b_tree_spilt_leaf( ):

(1) 开辟两个新的叶子节点 l_1、l_2;

(2) 计算当前叶节点 l 中距离最大的两点 l_a、l_b 分别作为新开辟节点 l_1、l_2 的中心点,根据欧式距离,将剩余节点分配到两个节点 l_1、l_2 中;

(3) 将新节点 l_1、l_2 插入,删除原节点 l。

2.2 参数估计

(1) 分支因子

试验中,对 141 幅木材图像进行统计,发现同一块板材图像中,没有缺陷的图像占 4.96%,包含 1 类缺陷的占 52.48%,包含 2 类缺陷的占 39%,大于等于 3 类缺陷的占 3.55%。所以考虑对 B 的取值为大于 5,尽量减小由于节点数目导致的初始聚类分裂,可取 $B=20$。为了避免因为 L 的影响导致初始聚类数目增加,取 L = 样本数 + 10,即尽量使属于同一类的样本插入到相同的叶子节点当中。

(2) 阈值

采用 VTT Building Technology 提供的 WOOD IMAGE DATABASE,使用颜色矩特征提取。通过大量实验,对在不同阈值下建树所呈现出的叶子节点初始聚类数目的不同所需的阈值进行统计。图 2 反应了其中 40 组样本,产生不同聚类数目所需的阈值。

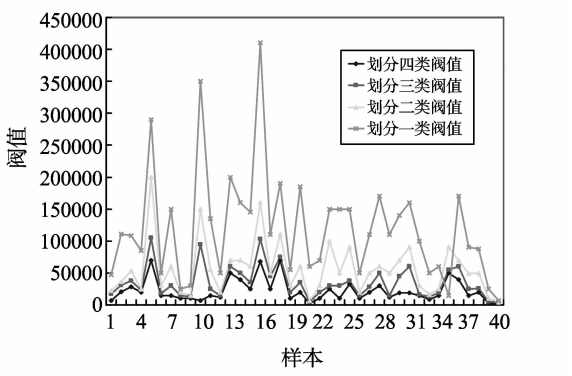


图 2　阈值统计

Fig. 2　Statistical threshold

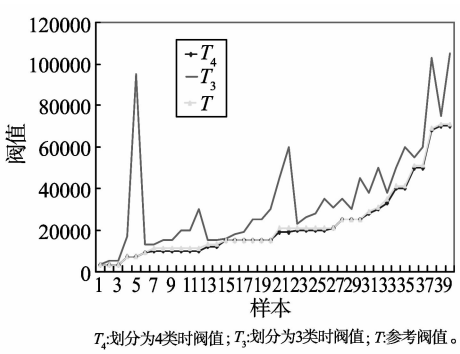
40 组样本中,发现取阈值为 25 000 时,25% 样本由于阈值选取太小须重建 CF 树,当阈值提高到 50 000时,7.5% 样本须重建 CF 树;当阈值选取为 25 000时,97.5% 样本被划分成至少 2 类,当阈值提高到 50 000 时,77.5% 样本被有效划分,见表 1。

表 1　40 组样本统计

Table 1　Statistical sample of 40 groups

阈值 T	需重建 CF 树的样本数	划分为 4 类的样本数	划分为 3 类的样本数	划分为 2 类的样本数	划分为 1 类的样本数
25 000	10	12	7	8	1
30 000	8	8	8	12	4
40 000	5	7	12	12	4
50 000	3	5	10	13	9

对于阈值选取太小,存在提升阈值的情况,对阈值的增量,即步长,作了简单分析。为了清楚起见,首先将样本按照划分为 4 类所需最低阈值进行升序排序,如图 3 所示。其阈值最小值为 3 000,最大值为 70 000。发现,首先取阈值为最小值 3 000,如果需要通过提升阈值重构 CF 树,则每次阈值提升 2 000,即:$T_{i+1}=T_i+2\,000$,可以让所有阈值落入划分为 3 类和 4 类之间,即得到样本的 4 类划分。



T_4:划分为4类时阈值; T_3:划分为3类时阈值; T:参考阈值。

图 3　阈值比较

Fig. 3　Comparison of threshold

2.3 非缺陷类判别

根据 CF 树中叶子节点记录的信息,可知每一子类中样本个数,以及每一子类的中心。可以通过两种方法判断出所有子类中,哪类对应的是非缺陷类。

(1) 通过叶子节点中记录的该子类中样本个数来判定。每一幅木材图像中,通常缺陷区域面积小于非缺陷区域,所以可以判断,叶子节点中包含样本数目最多的那一类就是非缺陷类。

(2) 根据相似性判定。随机统计 63 个非缺陷木材样本的颜色矩,求出它们的均值 $\bar{X}$。取出 CF 树中所有叶子节点的中心,通过欧式距离,计算 CF 树中所有叶子节点的中心与 $\bar{X}$ 的距离 D,将距离 D 最小的节点对应的样本集标记为非缺陷类,其余样本集分别用不同颜色矩形框标记。

3 仿真实验及测试

3.1 木材缺陷识别算法流程

根据聚类算法,设计了本次木材缺陷识别算法过程,如图 4 所示。

3.2 实验

在实验中,从 VTT Building Technology 提供的 WOOD IMAGE DATABASE 中选取了 81 个具代表性的样本作为聚类样本。通过颜色特征[5]提取来进行木材缺陷聚类,检测结束后,绘制切割示意图,并分析实验结果,判别 BIRCH 聚类算法对木材缺陷的识别能力。

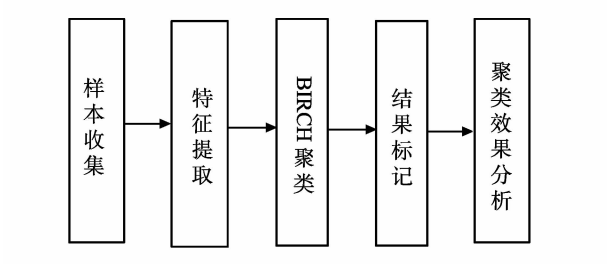


图4 木材缺陷识别过程
Fig. 4 Procedure of wood defects detection

通过与 *k*-Means 算法进行对比试验,82 组结果表明,BIRCH 算法具有较好的识别效果。给出其中 3 次实验的结果对比图,如图 5 所示。

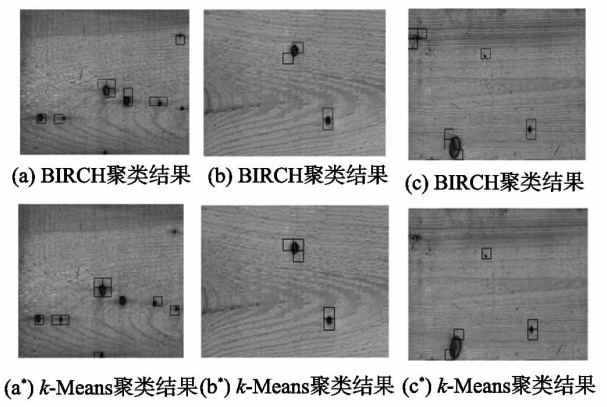


图5 木材缺陷识别结果对比
Fig. 5 Contrast of wood defect detection result

图 5(a)中存在小的虫眼,*k*-Means 算法未标记出来,而 BIRCH 算法正确标记。图 5(b)中缺陷类型为节子,*k*-Means 算法通过 3 个矩形框未将节子节点区域标记完全,BIRCH 算法则全都标记出来。图 5(c)中,左上角有小块腐朽缺陷,*k*-Means 算法未能识别,而 BIRCH 算法正确标记,并且下方节子节点区域比较大,*k*-Means 算法通过两个矩形框分别标记,而 BIRCH 算法几乎完全标记。

由表 2 可知,BIRCH 算法的平均识别查准率约为 86.3%,平均识别查全率约为 90.1%。与 *k*-Means 算法对比,查准率相差不大,但是查全率有较明显提高。其中 BIRCH 算法对于节子、虫眼的识别效果均明显优于 *k*-Means 算法,对于其它缺陷,如裂纹、变色等缺陷,*k*-Means 和 BIRCH 算法识别效果均不佳。

表2 木材缺陷识别结果比较
Table 2 Comparison of wood defect detection result

算法	查全率/%				平均查准率/%
	节子	虫眼	其它	平均	
k-Means	88.3	76.6	54	81.1	86.5
BIRCH	92.1	96.3	48.7	90.1	86.3

4 结论

介绍了一种新的基于 BIRCH 算法的木材缺陷识别方法。实验数据来源于 VTT Building Technology 提供的 WOOD IMAGE DATABASE。首先提取木材图像的颜色矩特征,再分别用 BIRCH 算法与 *k*-Means 算法进行缺陷类识别。对比结果表明,BIRCH 算法可以有效实现木材缺陷识别,具有被应用到实际生产的价值。

参考文献:

[1] AK Jain, RC Dubes. Algorithms for clustering data[M]. London: Prentice-Hall Advanced Reference Series, 1988: 1-334.

[2] SUN Jigui, LIU Jie, ZHAO Lianyu. Clustering algorithm studying[J]. Learned Journal of Software, 2008, 19(1): 48-61.

[3] ZHANG Tian, RAMAKRISHNAN Raghu, LINVY Miron. BIRCH: an efficient data clustering method for very large databases[C]//Proc ACM SIGMOD Int Conf on Management of Data. [S. l.]: ACM Press, 1996: 103-114.

[4] SHAO Fengjing, ZHANG Bin. Multi-threshold BIRCH clustering algorithm and it's applications[J]. Computer Engineering and Applications, 2004, 40 (12): 174-176.

[5] THEODORIDIS Sergios, KOUTROUMBAS Konstantinos. Pattern recognition[M]. LI Jingjiao, WANG Aixia, ZHANG Guangyuan, et al tran. Beijing: Electronic Industry Press, 2006.

(编辑:陈燕)