

REVIEWS

THE REVIEWS

REVIEWS

REVIEWS



REVIEWS

C215-53
546

北京大学院士文库

石青云文集

石青云 著

北京大学出版社
北京

内 容 提 要

本书由四部分组成。第一部分论述树分类器的设计方法及其在癌细胞自动识别方面的应用。第二部分论述如何用高维属性文法实现统计与句法模式识别的有效结合,及其在图像分析、英文字符识别方面的应用。第三部分是图像数据库的理论、方法与实用系统研制,作为示例着重介绍了作者主持研究的地理图像数据库系统和指纹自动识别系统。第四部分是图像压缩与计算机视觉研究,主要讨论了计算机视觉的代数几何方法和基于小波分析的图像压缩与机器视觉方法。

全书贯穿了数学与视觉信息处理和模式识别相互交叉与促进的特点,可供相关领域的读者参考。

图书在版编目(CIP)数据

石青云文集/石青云著. —北京:北京大学出版社,2001.6

(北京大学院士文库)

ISBN 7-301-04109-8

I. 石… II. 石… III. -文集 IV. TP·0446

书 名:石青云文集

著作责任者:石青云

责任编辑:沈承凤

标准书号:ISBN 7-301-04109-8/TP·0446

出版者:北京大学出版社

地 址:北京市海淀区中关村北京大学校内 100871

电 话:出版部 62752015 发行部 62559712 编辑部 62752032

排 版 者:北京因温特有限公司激光照排中心

印 刷 者:北京大学印刷厂

发 行 者:北京大学出版社

经 销 者:新华书店

787×1092 16开本 16.75印张 282千字

2001年6月第1版 2001年6月第1次印刷

定 价:39.00元

第一部分

树分类器与癌细胞识别应用

树分类器与特征选择*

树分类器的主要特点是每一输入图像必须经过多级判决,才能最后判定其所属类别.这种多级判决过程的形象表示是一棵树,图 1 就是一个例子.它表示一个五类问题的树分类器,其中 n_1 是树根, n_2 — n_6 是中间结点, n_7 — n_{13} 是树叶.通常又称 n_1 — n_6 为非终止结点,而 n_7 — n_{13} 则称为终止结点.从图上可看出,每个非终止结点下面有两个分支,这种情况称为二叉树.

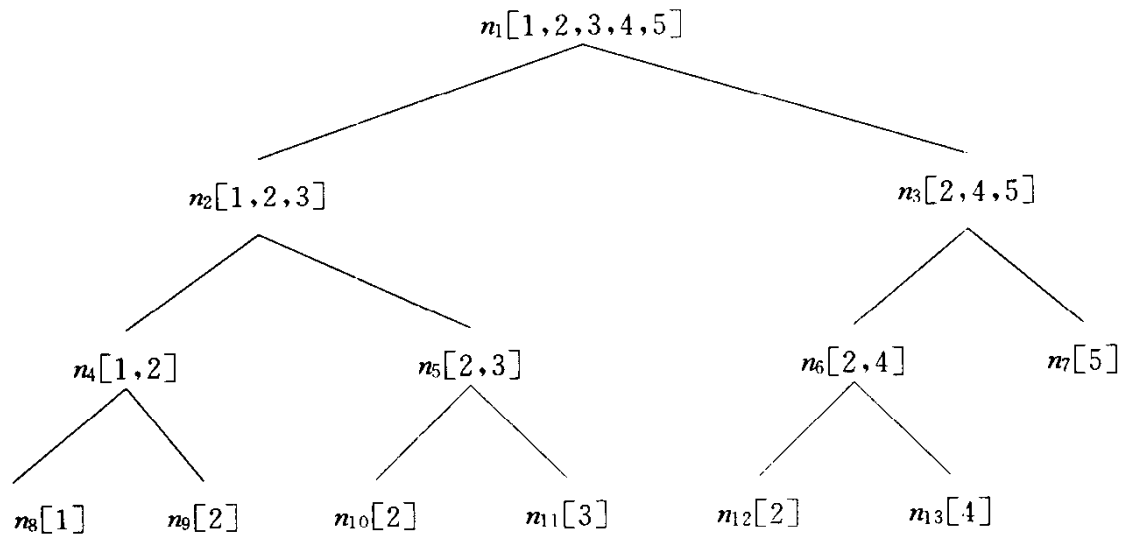


图 1

在树分类器中的每个非终止结点,都有一个与它相联系的判决规则.而在每个终止结点,则有一个与它相对应的类别.一般说来,几个终止结点都对应同一类别的情况是存在的,如图 1 中的 n_9 , n_{10} 和 n_{12} .

当输入一个图像样本时,它首先被送到树根.然后,经过一条路径所对应的一系列判决,最后到达树叶,其所属类别的判定就完成了.这当中经历的判决次数,正好是该路径的长度.例如,设一个图像样本经路径 $(n_1, n_2)(n_2, n_5)(n_5, n_{10})$ 到达终止结点 n_{10} (见图 1),则所经历的判决次数为

* 本文摘自于《图像识别导论》,234—240 页,上海:上海科技出版社,1983.

3,即与非终止结点 n_1, n_2 和 n_3 相联系的判决.一旦它到达终止结点 n_{10} ,就被判定为属于第二类.

为便于了解树分类器的优点,先看一个简单的例子.设两类图像样本在特征平面上均匀地分布于图 2 所示的区域.那么,要设计一个分类器使得经过一次判决就能准确地把两类样本分开,是比较困难的.但若采用图 3 所示的树分类器,则每次只需应用一个特征作线性判决,就可以达到理想的分类结果.

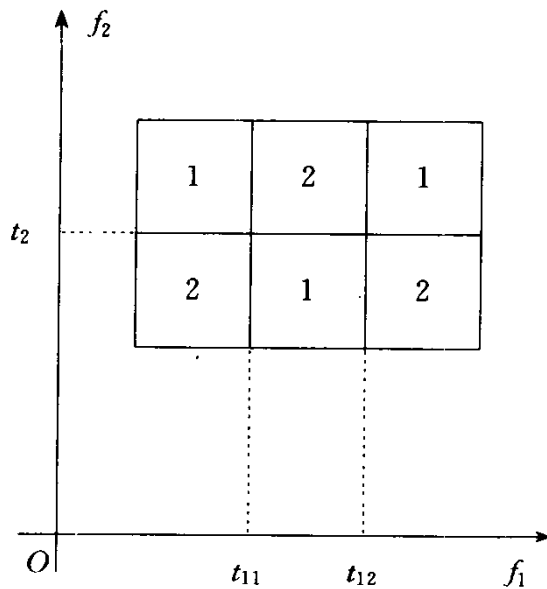


图 2

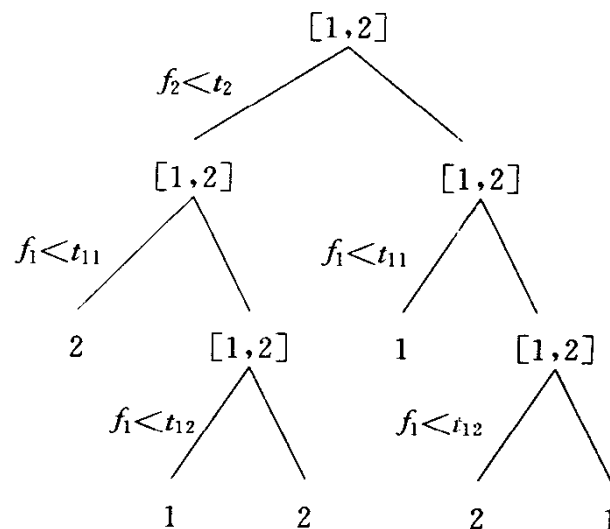


图 3

在处理多类、多特征的问题时,树分类器的优点尤其明显.概括起来,有如下方面:

(1) 有一些特征对区分某些类非常有效,但却往往不被一次判决的分类器所选中,因为这些特征对区分其它的类可能没有用处.然而,对于树分类器,这种特征却能发挥作用.这是由于树分类器中每个非终止结点所对应的判决,都选用那些最有利于划分两个子树的特征,以此来达到提高整个树的正确分类率这一最终目的.

(2) 在处理多类问题时,设计一次判决的分类器常遇到所谓“维数问题”.就是说,若希望多用特征来提高正确分类率,就必须相应地增加训练样本的数目,否则,会适得其反.但对于树分类器,这个问题就不那么突出了.因为树分类器中的每次判决,都只选用少量特征.而不同特征又可在

不同的判决中发挥作用。

(3) 由于树分类器的每次判决相对简单, 虽然判决的次数增多了, 但判定一个样本所属类别的总计算量不一定增加。

现在考虑树分类器的设计问题. 一般说来, 需要完成三方面的任务:

① 确定树的结构; ② 对每个非终止结点, 选择一个有效的特征子集; ③ 确定与每个非终止结点相应的判决规则. 我们只讨论二叉树的情况。

1. 确定树的结构

衡量树分类器结构好坏的标准, 首先是它的正确分类率; 其次, 也要考虑由树根到达每个终止结点所需的平均判决次数. 但是, 整体结构的优化问题非常复杂, 其原因在于: 即使图像样本的类别不太多, 可供选择的结构数量也非常可观. 因此, 目前设计多采取逐点优化方法, 现简述如下:

设 n_0 是一个非终止结点, 到达此结点的训练样本来自 p 类, 我们令 $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_p]^T$ 为分组规则, 其中每个 α_i 取 1 或者 -1. 其次再设正值函数 $Q = Q(\alpha) = Q(\alpha_1, \alpha_2, \dots, \alpha_p)$ 为衡量分组好坏的准则 (Q 的选择依设计者而定, 此处不详述). 那么, 在 n_0 处的结构优化问题, 也就是最合理地划分两个子树的问题, 可以归结为: 选择 α 使得

$$Q(\alpha) = Q(\alpha_1, \alpha_2, \dots, \alpha_p) = \min. \quad (1)$$

有时, 还附加约束条件:

$$\left| \sum_{j=1}^p \alpha_j \right| = \min. \quad (2)$$

这个条件的作用是使所设计的树分类器更加平衡, 以便降低由树根到达每个终止结点所需的平均判决次数. 为此, 我们只须在分量含有 $\left[\frac{p}{2}\right]$ 个 1 和 $p - \left[\frac{p}{2}\right]$ 个 -1 的那些 α 中选择一个, 记为 α^* , 使得 (1) 式成立. 下面的算法实现这一目标.

输入: 准则函数 $Q(\alpha)$;

输出: $\alpha^* = [\alpha_1^*, \dots, \alpha_p^*]^T$ 使得 (1) 和 (2) 式成立.

方法:

(1) 令 $T = Q(1, \dots, 1, -1, \dots, -1)$, 其中 1 共有 $\left[\frac{p}{2}\right]$ 个; -1 共有

$p - \left\lceil \frac{p}{2} \right\rceil$ 个.

(2) 令 $\alpha_j = 1, j = 1, 2, \dots, p$.

(3) 令 $I = 1$ 和 $J = 1$.

(4) 令 $\alpha_j = \alpha_j - 2$.

(5) 若 $I < p - \left\lceil \frac{p}{2} \right\rceil$, 则令 $K(I) = J$, 以及 $I = I + 1, J = J + 1$ 并转到第 4 步.

(6) 若 $Q(\alpha_1, \dots, \alpha_p) < T_h$, 则令 $T_h = Q(\alpha_1, \dots, \alpha_p)$, 以及 $\alpha_j^* = \alpha_j (j = 1, 2, \dots, p)$.

(7) 令 $\alpha_j = \alpha_j + 2$, 然后 $J = J + 1$.

(8) 若 $J \leq I + \left\lceil \frac{p}{2} \right\rceil$, 则转到第 4 步.

(9) 若 $I > 1$, 则令 $I = I - 1$ 以及 $J = K(I)$ 并转到第 7 步.

(10) 输出 $\alpha^* = [\alpha_1^*, \dots, \alpha_p^*]^T$, 停机.

在上述算法完成以后, α^* 就是满足(1)和(2)式的分组规则. 于是, 凡 $\alpha_j^* = 1$ 所对应的那几类样本, 都应该把它们划分到左边的子树, 其余的样本则希望划分在右边的子树. 下一步任务是: 选择有效的特征子集和确定相应的判决规则, 来尽可能准确地实现这一划分.

2. 特征子集的选择

在非终止结点 n_0 处的分组原则确定之后, 要实现分组只需设计一个两类问题的分类器. 前面已进行过充分讨论的有关技术, 如特征选择和判决规则(或边界)的确定, 现在都可以借鉴. 但是, 正如我们曾经提到的, 树分类器中的每次判决, 一般都只选用一个元素不多的特征子集. 因此, 需要再补充一些选择有效的特征子集的方法.

设 $X = \{f_1, f_2, \dots, f_N\}$ 是 N 个元素的特征集合(不妨设它就是原来的特性集合). 如果有一正值函数 $Q = Q(x)$ 在 X 的任一子集 x 上都有定义, 并且 Q 的值可以作为衡量该子集好坏的准则, 那么, 就可以着手进行特征选择了.(用什么样的准则函数, 一般依设计者而定, 此处不详述. 有兴趣的读者, 可参考有关文献.)

为了从 X 中挑出 r 个元素构成较好的特征子集, 最简单的办法是:

设 $x_k = \{f_k\} (k=1, 2, \dots, N)$, 若

$$Q(x_{k_1}) \leq Q(x_{k_2}) \leq \dots \leq Q(x_{k_r}) \leq \dots \leq Q(x_{k_N}),$$

则选取特征子集 $X_* = \{f_{k_1}, f_{k_2}, \dots, f_{k_r}\}$.

上述方法的明显缺点是: 没有考虑特征之间可能存在相关性. 而在相关性的影响下, 一般地说, 用两个最好的特征组合起来, 不一定是两个特征的组合中的最佳者. 克服这个缺点的途径之一, 是采用序贯选入与序贯剔除相结合的搜索法, 下面就是这种方法的一个例子.

(1) 取 $l > k > 0$, 使 r 为 $l-k$ 的整数倍, 并令 X_0 为空集.

(2) 若 $f_i \in X - X_0$ 使得 $Q(X_0 \cup \{f_i\}) = \min_{f_j \in X - X_0} Q(X_0 \cup \{f_j\})$, 则令 $X_0 = X_0 \cup \{f_i\}$.

(3) 重复第 2 步 l 次.

(4) 若 $f_i \in X_0$ 使得 $Q(X_0 - \{f_i\}) = \min_{f_j \in X_0} Q(X_0 - \{f_j\})$, 则令 $X_0 = X_0 - \{f_i\}$.

(5) 重复第 4 步 k 次.

(6) 若 X_0 含有 r 个元素, 则输出 X_0 并停机, 否则, 转到第 2 步.

这种方法的计算量也比较省, 不过, 只能选出次优特征子集. 如果希望从 X 中选出含有 r 个元素的特征子集使准则函数取极小值, 就需要检验 $\binom{N}{r}$ 个可能的特征组合. 在 N 比较大时, 一般都是计算量极大的. 但是, 若准则函数具有某种单调性, 则可应用“分支估界”搜索法. 它既能显著地节省计算量, 又能选出上述意义的最优特征子集. 下面就来介绍这个方法.

首先, 假设准则函数具有下列单调性: 若特征子集 $X_1 \subseteq X_2 (\subseteq X)$, 则

$$Q(X_1) \geq Q(X_2). \quad (3)$$

其次, 考虑需要检验的特征子集的组合方式. 为便于概括一般情况, 先看一个简单的例子: 设有整数集合 $\{1, 2, 3, 4, 5\}$, 现在要从其中任取两个元素构成子集, 那么共有 $\binom{5}{2} = 10$ 种组合方式. 而且, 这些子集恰好是图 4 中那棵树的所有终止结点标记.

现在反过来考虑 $X = \{f_1, f_2, \dots, f_N\}$, 我们按下列方式依次构造特征子集:

$$\begin{aligned}
 \text{一级: } X_{j_1} &= X - \{f_{j_1}\} & (j_1 = 1, 2, \dots, r+1); \\
 \text{二级: } X_{j_1 j_2} &= X_{j_1} - \{f_{j_2}\} & (j_2 = j_1 + 1, \dots, r+2); \\
 & \dots\dots\dots \\
 \text{N-r 级: } X_{j_1 j_2 \dots j_{N-r}} &= X_{j_1 \dots j_{N-r-1}} - \{f_{j_{N-r}}\} & (j_{N-r} = j_{N-r-1} + 1, \dots, N).
 \end{aligned}$$

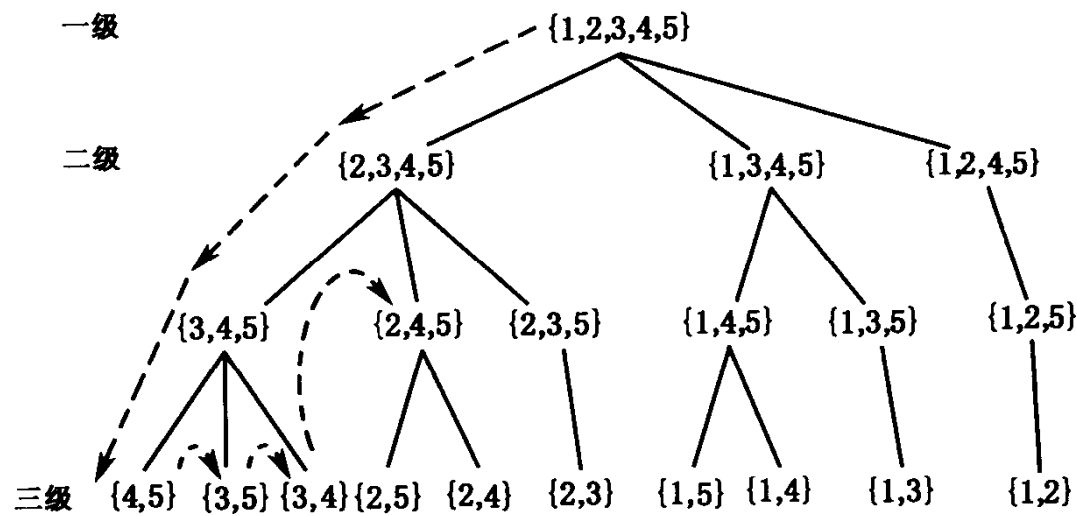


图 4

那么容易看出：若以 X 为树根标记，并依次以上述子集为各级结点标记，就可构成像图 4 中那样的树。而且，所有 $N-r$ 级结点标记表示的那些子集，都恰好含有 r 个元素，正是我们要检验的那些子集。然而，用“分支估界”搜索法，一般不必对所有这些子集都一一加以检验。这是因为：若取初界 $B=Q(X_*)$ ，其中 $X_* = \{f_1, f_2, \dots, f_r\}$ ，并按图 4 中虚线所示的路径开始搜索，根据准则函数 Q 的单调性（见(3)式），只要某个 k 级结点对应的特征子集 X^k 满足 $Q(X^k) \geq B$ ，那么，以 X^k 为根的子树中所有结点对应的特征子集都应该满足 $Q \geq B$ 。于是可以断言：在这个子树中，没有我们需要搜索的目标，这部分工作就可以省掉了。另一方面，若在搜索过程中发现某个 $N-r$ 级结点所对应的特征子集 $X_{j_1 j_2 \dots j_{N-r}}$ 使得 $Q(X_{j_1 \dots j_{N-r}}) < B$ ，则可以断言 $X_{j_1 \dots j_{N-r}}$ 是到目前为止所发现的最佳子集，于是修改 B 和 X_* ，即令 $X_* = X_{j_1 \dots j_{N-r}}$ 和 $B=Q(X_*)$ ，然后再继续搜索。下面的算法是上述思想的具体化。

输入：特征集合 $X = \{f_1, f_2, \dots, f_N\}$ 及准则函数 Q ；

输出：含有 r 个元素的特征子集 X_* 使 $Q(X_*) = \min$ 。

方法：

- (1) 令 $X_* = \{f_1, f_2, \dots, f_r\}$ 和 $B = Q(X_*)$.
- (2) 令 $X_c = X, k=1$ 和 $j=1$.
- (3) 令 $X_c = X_c - \{f_j\}$.
- (4) 若 $Q(X_c) \geq B$, 则转到第 7 步.
- (5) 令 $J(k) = j$.
- (6) 若 $k < N - r$, 则令 $k = k + 1, j = j + 1$ 并转到第 3 步. 否则, 令 $B = Q(X_c)$ 和 $X_* = X_c$.
- (7) 令 $X_c = X_c \cup \{f_j\}$, 然后 $j = j + 1$.
- (8) 若 $j \leq k + r$, 则转到第 3 步.
- (9) 若 $k > 1$, 则令 $k = k - 1$ 以及 $j = J(k)$ 并转到第 7 步.
- (10) 输出 X_* , 停机.

“分支估界”搜索法还有其它具体形式. 特别是结合具体的准则函数, 可以安排得更有利于节省计算量. 有兴趣的读者, 可参考有关文献.

在特征选择之后, 如何确定判决规则, 是前几章已经讨论过的问题, 这里就不再重复了. 但要指出, 必须在每个非终止结点都完成上述三项任务, 才能实现整个树分类器的设计.

最后, 哪些结点可以当作终止结点呢? 这个问题必须根据误分类的容限来决定.

A Method for the Design of Binary Tree Classifiers*

Abstract: A method based on multivariate stepwise regression is proposed for the design of binary tree classifiers. Experimental results of cell classification are given to illustrate the efficiency of the proposed method.

Keywords: Pattern classification, Tree classifier, Feature selection, Branch and bound search, Discriminant function, Multivariate stepwise regression

I. INTRODUCTION

Tree classifiers have been shown by many authors⁽¹⁻⁷⁾ to be more efficient than single stage classifiers for pattern classification. However, the existing approaches to the design of binary tree classifier often require a large amount of computing time and storage. In this paper, a method based on the multivariate stepwise regression is suggested. For the proposed criterion, this method can be used to optimize the structure of a binary tree classifier and to select an optimal or suboptimal feature subset at each nonterminal node. For each nonterminal node, a linear discriminant function is produced at the same time that the feature selection is completed. The computation time and memory space required appear to be quite reasonable.

An experimental comparison between the proposed method and the method proposed by Lin and Fu for cell classification is given to illus-

* Published in PATTERN RECOGNITION, Vol. 16, No. 6, pp. 593-603, 1983.

trate the effectiveness of this method.

II. REVIEW OF MULTIVARIATE STEPWISE REGRESSION

Detailed results of multivariate stepwise regression can be found in Albert^[8]. A brief review is given in the following.

1. Notations

The (i, i) transform of a matrix $A = [a_{ij}]_{m \times m}$ is defined as $A_* = [a_{kl}^*]_{m \times m}$ where

$$a_{kl}^* = \begin{cases} a_{ii}^{-1} & \text{if } k = i, l = i, \\ a_{ii}^{-1} a_{il} & \text{if } k = i, l \neq i, \\ -a_{ki} a_{ii}^{-1} & \text{if } k \neq i, l = i, \\ a_{kl} - a_{ki} a_{ii}^{-1} a_{il} & \text{if } k \neq i, l \neq i. \end{cases} \quad (2.1)$$

Let

$$X = [x_1 x_2 \cdots x_m] = [x_{ij}]_{n \times m} \text{ and } Y = [y_{ij}]_{n \times p}$$

be the observed data. If we introduce the following notations:

$$\bar{x} = \frac{1}{n} X'U, \quad \bar{y} = \frac{1}{n} Y'U,$$

$$L_{xx} = X'X - n\bar{x}\bar{x}', \quad L_{yy} = Y'Y - n\bar{y}\bar{y}'$$

and

$$L_{xy} = X'Y - n\bar{x}\bar{y}' = L_{yx}',$$

then the regression relation between Y and X can be expressed as

$$Y = [UX] \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} + \epsilon, \quad (2.2)$$

where $U = [1 \ 1 \ \cdots \ 1]'$ is an $n \times 1$ vector, $\hat{\beta}_0 = \bar{y}' - \bar{x}'\hat{\beta}$ and $\hat{\beta} = L_{xx}^{-1}L_{xy}$ are the regression coefficients' matrices and $Q = L_{yy} - L_{yx}L_{xx}^{-1}L_{xy}$ is the residual.

2. Multivariate stepwise regression

Suppose that $X = [x_{ij}]_{n \times m}$, $Y = [y_{ij}]_{n \times p}$ and the regression relation be-

tween $[x_1 x_2 \cdots x_{k-1}]$ and Y has been established as

$$Y = [U x_1 x_2 \cdots x_{k-1}] \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} + \epsilon. \quad (2.3)$$

It is desired to insert a new variable x_k in the right-hand side of (2.3). In this case, two questions need to be answered: ① how will the regression coefficients' matrix and residual change; ② how can we evaluate the new variable?

According to the properties of (i, i) Transform, if the "original matrix L " is equal to

$$\begin{bmatrix} L_{xx} & L_{xk} & \cdots & L_{xy} \\ L_{kx} & L_{kk} & \cdots & L_{ky} \\ \cdots & \cdots & \cdots & \cdots \\ L_{yx} & L_{yk} & \cdots & L_{yy} \end{bmatrix},$$

then applying (i, i) Transform $k-1$ times (corresponding to $x_1, x_2, \cdots, x_{k-1}$) will result in the "current matrix L "

$$\begin{bmatrix} L_{xx}^{-1} & L_{xx}^{-1}L_{xk} & \cdots & L_{xx}^{-1}L_{xy} \\ -L_{kx}L_{xx}^{-1} & L_{kk} - L_{kx}L_{xx}^{-1}L_{xk} & \cdots & L_{ky} - L_{kx}L_{xx}^{-1}L_{xy} \\ \cdots & \cdots & \cdots & \cdots \\ -L_{yx}L_{xx}^{-1} & L_{yk} - L_{yx}L_{xx}^{-1}L_{xk} & \cdots & L_{yy} - L_{yx}L_{xx}^{-1}L_{xy} \end{bmatrix}. \quad (2.4)$$

Again by applying (k, k) Transform once (corresponding to x_k), we obtain the "next matrix L "

$$\begin{bmatrix} \cdots & \cdots & \cdots & L_{xx}^{-1}L_{xy} - L_{xx}^{-1}L_{xk}v^{-1}(L_{ky} - L_{kx}L_{xx}^{-1}L_{xy}) \\ \cdots & v^{-1} & \cdots & v^{-1}(L_{ky} - L_{kx}L_{xx}^{-1}L_{xy}) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & -(L_{yk} - L_{yx}L_{xx}^{-1}L_{xk})v^{-1} & \cdots & (L_{yy} - L_{yx}L_{xx}^{-1}L_{xy}) - (L_{yk} - L_{yx}L_{xx}^{-1}L_{xk})v^{-1}(L_{ky} - L_{kx}L_{xx}^{-1}L_{xy}) \end{bmatrix} \quad (2.5)$$

where $v = L_{kk} - L_{kx}L_{xx}^{-1}L_{xk}$ is the (k, k) element of the "current matrix L ". If we denote the (i, j) element of the "current matrix L " by l_{ij} , then the new residual will be

$$Q_* = L_{yy} - L_{yk}l_{kk}^{-1}L_{ky}. \quad (2.6)$$

For the evaluation of a new variable, the following measurement

may be used:

$$T = l_{kk}^{-1} L_{kyc} L_{ykc}. \quad (2.7)$$

The new variable x_k with the maximum value of T is considered as the best candidate for insertion.

Now consider another question. If it is desired to delete a variable from the right-hand side of a regression relation, what should be done? Without loss of generality, suppose that the regression relation between $[x_1, x_2, \dots, x_k]$ and Y has been established, and the variable x_k will be deleted. In this case, we only need to consider (2.5) as the "current matrix L ", whereas (2.4) is the "next matrix L ", so

$$Q_* = Q - L_{ykc} l_{kk}^{-1} L_{kyc} \quad (2.8)$$

and the corresponding measurement is

$$T = - l_{kk}^{-1} L_{kyc} L_{ykc}. \quad (2.9)$$

Then the variable x_k with the minimum value of T is considered as the first candidate for deletion.

III. DESIGN OF BINARY TREE CLASSIFIER

There are three tasks to be completed for the design of a binary tree classifier.

- (1) Determine the structure of the binary tree classifier.
- (2) Select a feature subset at each nonterminal node.
- (3) Choose a decision rule used at each nonterminal node.

In order to explain the proposed method, consider the root of the binary tree as an example. At the root node all n samples are from p classes. The number of samples, which belong to the j th class, is equal to n_j ,

$\sum_{j=1}^p n_j = n$. If the feature set preselected consists of m features, let

$$X = [x_1, x_2, \dots, x_m]_{n \times m}$$

and Y be a $n \times p$ diagonal matrix with diagonal elements $u_j, j = 1, \dots, p$, where u_j is the n_j -dimensional unit column-vector. The column vector x_i

consists of the values of the i th feature. Then form the matrix

$$\begin{bmatrix} L_{xx} & L_{xy} \\ L_{yx} & L_{yy} \end{bmatrix} \quad (3.1)$$

as defined in Section II. When this is completed, perform the three tasks as follows.

1. Determining the structure of the binary tree classifier

Set $\alpha = [\alpha_1 \alpha_2 \cdots \alpha_p]'$, where $\alpha_j = 1$ or -1 , and $y = Y\alpha$. It is easy to see that

$$L_{xy} = X' \left(I - \frac{1}{n} J \right) y = X' \left(I - \frac{1}{n} J \right) Y \alpha = L_{xy} \alpha \quad (3.2)$$

where $J = UU'$.

Similarly,

$$L_{yx} = y' \left(I - \frac{1}{n} J \right) X = \alpha' Y' \left(I - \frac{1}{n} J \right) X = \alpha' L_{yx} \quad (3.3)$$

and

$$l_{yy} = y' \left(I - \frac{1}{n} J \right) y = \alpha' Y' \left(I - \frac{1}{n} J \right) Y \alpha = \alpha' L_{yy} \alpha. \quad (3.4)$$

Therefore, selecting the i th feature will result in the residual

$$Q = l_{yy} - L_{yi} l_{ii}^{-1} L_{iy} = l_{yy} - l_{ii}^{-1} \alpha' L_{yi} L_{iy} \alpha,$$

where l_{ii} denotes the (i, i) element of L_{xx} , L_{iy} denotes the i th row of L_{xy} , and L_{yi} denotes the i th column of L_{yx} . Obviously, $L_{yx} = L_{xy}'$, thus

$$T(i, \alpha) \triangleq l_{ii}^{-1} \alpha' L_{yi} L_{iy} \alpha = l_{ii}^{-1} (\alpha' L_{yi})^2.$$

Now let

$$T(\alpha) = \max_i T(i, \alpha).$$

It is desired to choose $\alpha = [\alpha_1 \alpha_2 \cdots \alpha_p]'$ such that

$$\left. \begin{aligned} T(\alpha) = \max \\ \left| \sum_{j=1}^p \alpha_j \right| = \min. \end{aligned} \right\} \quad (3.5)$$

The second condition in (3.5) will make the designed binary tree classi-