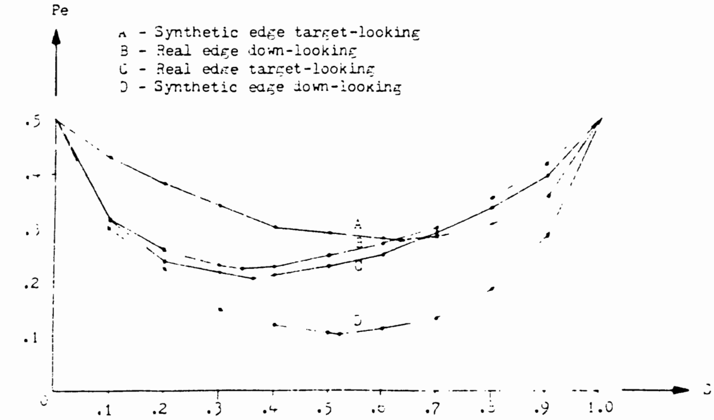

Fig. 8. The variations of Pe as a function of S for the real and synthetic edge images of down-looking and target-looking views of a scene.

views. This phenomenon is also observed in the results of the edge correlations in Figs. 7 and 8.

The comparisons of the various probabilities of error show that (Bayes) Pe $<$ (CH) Pe $<$ (BH) Pe $\ll$ (F) Pe, as was expected. Fisher's criteria produces the highest probability of error, and Bayes the smallest, with Chernoff and Bhattacharyya in between. There are considerable differences in the complexities of the implementation of these measures. Fisher's criteria is easier to use, but may occasionally lead to ambiguous and unreliable decisions. The difference in the measurement values for different images can be attributed to the different ways of obtaining these pictures.

VII. Summary and Conclusion

In this study four separability measures—Bayes probability of error, Chernoff distance, Bhattacharyya distance, and Fisher's distance—were used for two general Gaussian densities with nonequal means and variances. Then an attempt was made to use these separability measures in selecting a reference image from the set of real and synthetic images of down-looking and target-looking views of a particular scene. The experimental results show that

1) the theoretical sequence of bounds holds and demonstrates the difference between the linear and nonlinear decision procedures,

2) for real images, the target-looking view performs better than the down-looking view for both area and edge correlations,

3) for synthetic images, the down-looking view performs better than the target-looking view for both area and edge correlations,

4) the variations in synthetic images, between the target-looking and down-looking views, is larger than in the real images for both area and edge correlations.

References

[1] F. Sadjadi and E. Hall, "Image registration using material information," in *Proc. SPIE 23rd Int. Symp.*, San Diego, CA, 1979.
[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.
[3] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis on the sum of observations," *Ann. Math. Statist.*, vol. 23, pp. 493–507, 1962.
[4] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, no. 3, pp. 99–110, 1943.
[5] F. Sadjadi and E. Hall, "Segmentation using material information," *Opt. Eng.*, May 1980.
[6] K. Fukunaga and R. D. Short, "The feature extraction problem for statistical pattern recognition," Purdue Univ., West Lafayette, IN, TR-EE-78-48, Dec. 1978.

Hierarchical Classifier Design Using Mutual Information

I. K. SETHI AND G. P. R. SARVARAYUDU

Abstract—A nonparametric algorithm is presented for the hierarchical partitioning of the feature space. The algorithm is based on the concept of average mutual information, and is suitable for multifeature multicategory pattern recognition problems. The algorithm generates an efficient partitioning tree for specified probability of error by maximizing the amount of average mutual information gain at each partitioning step. A confidence bound expression is presented for the resulting classifier. Three examples, including one of handprinted numeral recognition, are presented to demonstrate the effectiveness of the algorithm.

Index Terms—Beta functions, decision trees, handprinted numeral recognition, hierarchical partitioning, mutual information, nonparametric methods, Walsh series.

I. Introduction

The problem of classifier design can be considered as one of partitioning the feature space into a number of disjoint regions. The classification then is nothing but the determination of the region to which an unknown sample belongs. In many pattern recognition problems, the pattern classes are multimodal in nature, and it becomes difficult to use conventional partitioning procedures such as the Bayesian rule for classifier design. In such cases, the nonparametric partitioning of the feature space is usually preferred. The rationale for such hierarchical partitioning has been well summarized by Kanal [1], and optimization approaches to the design of hierarchical classifiers appear in Kulkarni and Kamal [2].

One of the simplest nonparametric methods of partitioning the feature space is to use a set of hyperplanes parallel to feature axes. However, the main difficulty in such a partitioning is the determination of the number of hyperplanes and their locations. Henrichon and Fu [3] have suggested an empirical method for finding such partitioning which can be implemented as a layered structure of threshold devices. Recently, the use of the Kolmogorov–Smirnov test has been made to partition the feature space which gives rise to a binary decision tree structure for the classifier [4], [5]. The main drawback of this approach is that it requires as many decision trees to be developed as there are pattern classes. In another recent work, Dattatreya and Sarma [6] suggest the use of the first- and second-order statistics of the labeled samples to discretize the feature space. A dynamic programming algorithm is then used to obtain the minimum cost decision tree for classification.

In this note, we present a simple method of hierarchical partitioning of the feature space using hyperplanes parallel to feature axes. The method is based on the concept of mutual information, and is applicable to multifeature multiclass pattern recognition problems. Besides presenting two illustrative examples, an example of application of the proposed method to the classifier design for numeral recognition is discussed to show the effectiveness of the hierarchical partitioning. A measure of confidence on the error performance of the classifier is also discussed.

Manuscript received May 1, 1981; revised January 20, 1982.

I. K. Sethi is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721 302, India.

G. P. R. Sarvarayudu is with the Jawaharlal Nehru Technological University, India, on leave at the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721 302, India.

0162-8828/82/0700-0441$00.75 © 1982 IEEE

II. Information Measure of Partitioning

Consider a two-category classification problem involving only one measurement x. Let $x = t$ define the partitioning of the one-dimensional feature space. If we view the measurement x taking on values greater or less than t as two outcomes x_1 and x_2 of an event X, then the amount of average mutual information obtained about the pattern classes from the observation of event X can be written as

$$I(C; X) = \sum_{i=1}^{2} \sum_{J=1}^{2} p(c_i, x_j) \log_2 [p(c_i/x_j)/p(c_i)] \tag{1}$$

where C represents the set of pattern classes c_1 and c_2 having *a priori* probabilities $p(c_1)$ and $p(c_2)$. $p(c_i, x_j)$ is the joint probability of occurrence of c_i and x_j and $p(c_i/x_j)$ is the probability that the observation comes from class c_i given the outcome x_j of event X. Clearly, for better recognition, the choice of threshold t should be such that we get maximum information about the classes c_1 and c_2 from the event X. This means that the value which maximizes (1) should be selected over all possible values of t. If both classes have unimodal distribution, it can easily be seen that this choice of threshold t turns out to be the same as that provided by the Bayesian rule for the zero-one loss function.

Suppose we set up a decision process to recognize the pattern classes c_1 and c_2 by observing the event X, i.e., whether the measurement x is greater than or less than the threshold t. Let P_e be the probability of error allowed in recognition; then the following inequality determines the limit on the equivocation $H(C/X)$ of C with respect to X [7]:

$$H(C/X) \leqslant H(P_e) + P_e \log_2 (m - 1) \tag{2}$$

where $H(P_e)$ is the error entropy and m is the number of pattern classes. Since the average mutual information can also be written as

$$I(C; X) = H(C) - H(C/X), \tag{3}$$

we get the following inequality from (2) and (3):

$$I(C; X) \geqslant H(C) - H(P_e) - P_e \log_2 (m - 1). \tag{4}$$

Using the equality sign in the above, we obtain the smallest amount of average mutual information needed between C and X for a given probability of error. Denoting this by $I_{\min}$ and expanding for $H(C)$ and $H(P_e)$, we get

$$\begin{aligned} I_{\min} = &- \sum_{J=1}^{m} p(c_j) \log_2 p(c_j) \\ &+ P_e \log_2 P_e + (1 - P_e) \log_2 (1 - P_e) \\ &- P_e \log_2 (m - 1). \end{aligned} \tag{5}$$

This equation thus relates the probability of error and the corresponding minimum value of average mutual information required for a recognition process.

Let us now consider a recognition problem involving m pattern classes with n features. The type of partitioning which results using hyperplanes parallel to feature axes can be conveniently represented in the form of a binary tree. For example, Fig. 1 shows a partitioned two-dimensional feature space and its tree representation. The average mutual information about pattern classes from such hierarchical partitioning can be obtained in terms of the mutual information available at the nonterminal nodes of the partitioning tree, since with each such node, an event is associated. Let l_k be an internal node of the partitioning tree T; then the average mutual information available at the node l_k can be written as

$$\begin{aligned} I_k(C_k, X_k) = &\sum_{C_k, X_k} P(c_{ki}, x_{kj}) \\ &\cdot \log_2 [P(c_{ki}/x_{kj})/P(c_{ki})] \end{aligned} \tag{6}$$

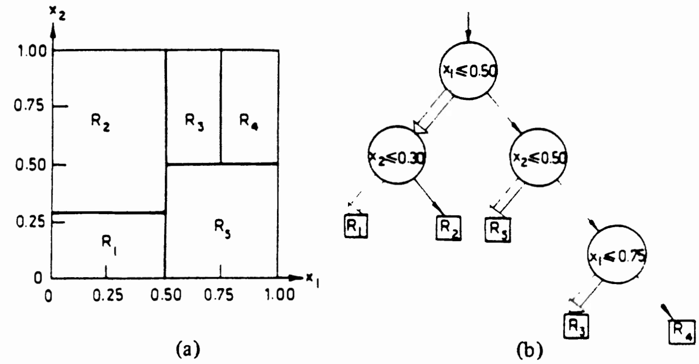

Fig. 1. (a) Two-dimensional partitioned space. (b) Corresponding hierarchical partitioning tree representation.

where C_k is the set of pattern classes and X_k is the event associated with the node l_k. If p_k is the probability of the pattern class set C_k, i.e., the sum of the probabilities of the partitioned regions that can be reached through the node l_k, then the average mutual information between the set C of pattern classes and the partitioning tree T can be expressed as

$$I(C; T) = \sum_{k=1}^{L} p_k I_k(C_k; X_k) \tag{7}$$

where $L \geqslant m - 1$ is the number of internal nodes of the tree T, or in other words, the number of parallel hyperplanes defining the hierarchical partitioning of the feature space. Now if we use the partitioning tree T as a classifier and treat the average mutual information $I(C; T)$ as $I_{\min}$ of (5), we can determine the probability of error in classification by (5). Alternately, if we specify the probability of error, (5) tells us the amount of mutual information to be provided by the partitioning tree T.

Thus, given a hierarchical partitioning tree T, we can evaluate the average mutual information and its classification performance using the above equations. However, the problem we face here is how to obtain an "efficient" hierarchical partitioning given a set of pattern vectors from various classes and specified error rate. By efficient, we mean that the partitioning should be such that it maximizes the average mutual information or it minimizes the average error of recognition for the given size of the tree. This can be done by following the recursive partitioning algorithm of the next section. The main feature of this algorithm is that it maximizes the amount of mutual information gain at each stage of partitioning to keep the size of the partitioning tree small.

III. Recursive Partitioning Algorithm

Formally, the problem we treat here can be stated as follows.

Given a set of n-dimensional design samples $Y_1, Y_2, \cdots, Y_N$ coming from m pattern classes, determine an efficient hierarchical partitioning tree for a specified probability of error P_e. As mentioned above, the proposed algorithm maximizes the amount of gain in average mutual information at each partitioning step. This is done by choosing a threshold t_{kj} for the jth feature at the kth partitioning step such that $p_k I_k(C_k; X_k)$ is maximum over all possible values of j and over all possible threshold values for each j. However, in the absence of the true values of the various probabilities, their estimates based on the available design samples are used to calculate the gain in average mutual information in the following fashion.

Suppose BELOW(i) and ABOVE(i) are, respectively, the number of samples from category c_i falling below and above the threshold value t_{kj} on the jth feature axis at the kth node. The event X_k then has two outcomes X_{k1} and X_{k2} where X_{k1} implies that the sample is falling below the threshold and X_{k2} means that the jth component of the sample vector is greater than the threshold. The various probabilities appearing in (6) can be computed by the following relations:

$$P(c_{ki}) = \text{ABOVE}(i) + \text{BELOW}(i)/\text{SUM},$$

$$P(c_{ki}, x_{k1}) = \text{BELOW}(i)/\text{SUM},$$

$$P(c_{ki}, x_{k2}) = \text{ABOVE}(i)/\text{SUM},$$

$$P(c_{ki}/x_{k1}) = \text{BELOW}(i)/\text{BSUM},$$

and

$$P(c_{ki}/x_{k2}) = \text{ABOVE}(i)/\text{ASUM}$$

where

$$\text{ASUM} = \sum \text{ABOVE}(i),$$

$$\text{BSUM} = \sum \text{BELOW}(i),$$

and

$$\text{SUM} = \text{ASUM} + \text{BSUM}.$$

The steps of the recursive partitioning algorithm are as follows.

1) Initialization:

a) Using the specified value of P_e, calculate $I_{\min}$ using (5). Initialize ITREE, i.e., the average mutual information provided by the hierarchical partitioning tree, equal to zero.

b) Initialize NODESET with 1. At any instant, the NODESET contains the list of nodes which are possible candidates for further partitioning.

c) Assign all the available design samples to LIST (1). In general, LIST (NODE) consists of samples associated with node NODE of the partitioning tree where NODE is a member of NODESET.

2) Looping: For each NODE of NODESET, do the following.

a) Order the samples from LIST (NODE) on all the feature axes in ascending fashion. For each such ordering, prepare an array of class labels such that s_{ij} is the ith-order sample on the jth feature axis. Fig. 2, for example, shows the formation of such a label array for ordered samples from three categories.

b) Examine the label arrays for possible threshold points, i.e., locations of the parallel hyperplanes. A threshold point is said to exist between $Y_{(ij)}$ and $Y_{(i+1j)}$ on the jth feature axis if $s_{ij} \neq s_{i+1j}$, i.e., if two consecutive entries in the label array are different, a threshold point exists between the corresponding ordered samples. For each possible threshold point, determine the threshold value as $t_j = [x_{(ij)} + x_{(i+1j)}]/2$ where x_{ij} is the jth feature of ith-order sample Y_{ij} and calculate the amount of gain in average mutual information possible by counting the different labels on both sides of the threshold point and making use of (6). If no possible threshold point is detected, then the node under consideration is termed as the terminal node with proper label and is deleted from NODESET.

c) Select the threshold point giving the maximum amount of gain in average mutual information over all possible threshold locations along different feature axes. In case of a tie between many threshold locations, select the one for which the quantity $[x_{(i+1j)} - x_{(ij)}]/[x_{(i+1j)} + x_{(ij)}]$ is the largest. Let $t_{j\max}$(NODE) be the threshold value of the selected point which occurs on the feature axis $j$max (NODE). Let AMIG(NODE) be the corresponding average mutual information gain.

3) Decision:

a) Determine the node NMAX from NODESET such that p_{NMAX}AMIG(NMAX) $\geqslant p_{\text{NODE}}$AMIG(NODE) for all nodes of NODESET. p_{NODE} for any node is calculated as the ratio of the number of samples in LIST (NODE) to the total number of design samples. Node NMAX then defines the current partitioning hyperplane which is located at $t_{j\max}$(NMAX) along the feature axis $j$max (NMAX).

b) Increment ITREE by p_{NMAX}AMIG(NMAX) and check whether it equals or exceeds $I_{\min}$. If the answer is no, go to the next step; otherwise, go to the step Termination of the algorithm.

c) Replace NMAX in NODESET by its left and right descendent nodes NMAXL and NMAXR. Determine the LIST(NMAXL) and LIST(NMAXR) by classifying the samples of LIST(NMAX) with the threshold chosen above and go back to step Looping.

4) Termination: Denote all the nodes of NODESET as terminal nodes of the partitioning tree. Label each terminal node with the class label having the majority count in the list of design samples associated with that terminal node.

In the next section, we present three examples of the hierarchical partitioning trees obtained using this algorithm.

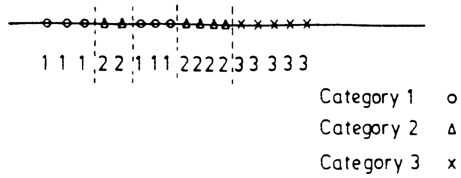

Fig. 2. Ordered samples from three categories and the corresponding label array. Dotted vertical lines show the possible threshold points.

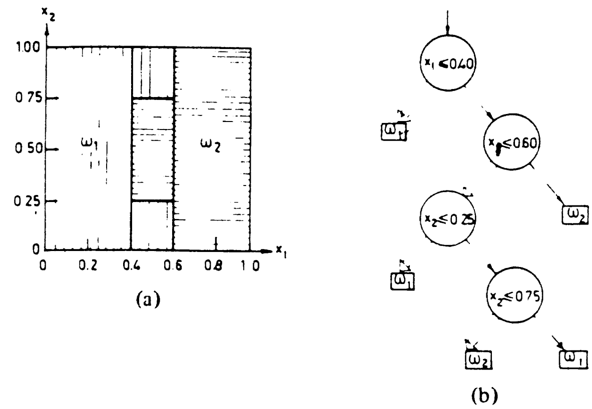

Fig. 3. (a) Two-dimensional feature space along with parallel hyperplanes for Example 1. (b) The partitioning tree generated by the recursive partitioning algorithm.

IV. Tree Design Example

Example 1: This example involves two perfectly separable pattern classes from a two-dimensional feature space. A design set consists of 200 independent bivariate observations of pattern class w_1 from the vertically hatched region of Fig. 3(a) having uniform distribution and 200 bivariate observations of pattern class w_2 drawn similarly from the horizontally hatched region of Fig. 3(a). Specifying zero error, the recursive partitioning algorithm generates the hierarchical tree of Fig. 3(b). $I(C; T)$ for this tree is 1 bit, which is the same as $I_{\min}$ given by (5) with P_e equal to zero.

Example 2: This example utilizes the Iris data [8] of three categories in the four-dimensional space. Using the first 25 samples from each category, the above algorithm was applied to obtain a classifier with $I_{\min}$ equal to 1.28 bits, which corresponds to P_e of 5 percent. The resulting hierarchical partitioning tree is shown in Fig. 4. From the 75 samples of the design set, only one sample belonging to category Iris Versicolor was found to lie in the wrong region when the algorithm terminated. Using the remaining Iris data as a test set, the classification performance of the tree of Fig. 4 was found to be 97.33 percent. It is also seen from Fig. 4 that the first two features of the Iris data, i.e., sepal length and sepal width, do not appear at all in the partitioning tree. Thus, the algorithm has inherent feature selection capability. This fact is further illustrated by the next example.

Example 3: This example demonstrates the capability of the recursive partitioning algorithm in a real application involving handprinted numeral recognition. The above algorithm was applied to design a hierarchical classifier which is expected to be a part of the low-cost handprinted numeral recognition system currently under development. The input to this classifier

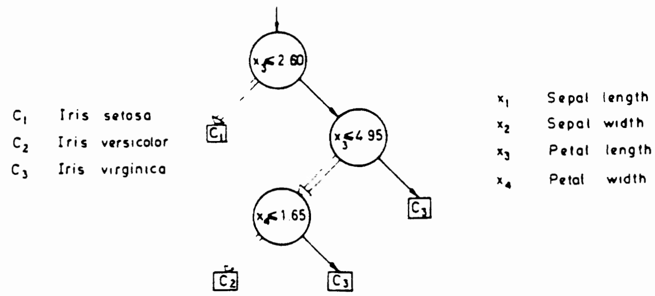

Fig. 4. Partitioning tree for the Iris data of Example 2.

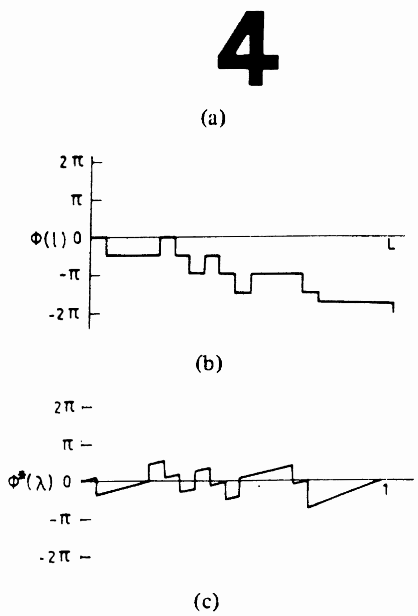

Fig. 5. (a) Numeral four. (b) Cumulative angular bend function $\phi(l)$ for numeral four. (c) Normalized cumulative angular bend function $\phi^*(\lambda)$. $\phi^*(\lambda) = \phi(l) + 2\pi\lambda$ where $\lambda = l/L$.

is a set of features which is obtained from the coefficients of the Walsh series expansion approximating the boundary of the unknown pattern [9]. Letting $\phi^*(\lambda)$ represent the boundary information in the form of a normalized cumulative angular bend function as the pattern boundary is traced, we use the following finite Walsh series expansion to approximate it:

$$\phi^*(\lambda) = a_0 \text{ Wal } (0, \lambda) + \sum_{i=1}^{N/2} [a_c(i) \text{ Cal } (i, \lambda) + a_s(i) \text{ Sal } (i, \lambda)] + a_s(N/2) \text{ Sal } (N/2, \lambda)$$

where

$$a_0 = \int_0^1 \phi^*(\lambda) \text{ Wal } (0, \lambda)\, d\lambda,$$

$$a_c(i) = \int_0^1 \phi^*(\lambda) \text{ Cal } (i, \lambda)\, d\lambda,$$

and

$$a_s(i) = \int_0^1 \phi^*(\lambda) \text{ Sal } (i, \lambda)\, d\lambda.$$

Fig. 5 shows an example of $\phi^*(\lambda)$ for the numeral four along with the unnormalized cumulative angular bend function. The seven features used by the classifier are obtained as

$$A_i = [a_c(i)^2 + a_s(i)^2]^{1/2}, \qquad i = 1, 7.$$

Each of these features thus corresponds to one of the first seven amplitudes in the sequency spectra of $\phi^*(\lambda)$. As the Walsh sequency spectra is sensitive to the choice of the starting point for the boundary trace, a normalization step in the computation of A_i's is required. The details of this are given in [9]. Taking 160 digitized samples of handprinted numerals, 16 each from categories 0–9, the above seven features for each pattern were extracted, and the resulting set of 160 seven-dimensional pattern vectors was passed on to the recursive partitioning algorithm. Because of the size of the design set and earlier results of the classification on these data [9], P_e was specified to be 20 percent. This corresponds to $I_{\min}$ of 1.96 bits assuming equal *a priori* probability for all pattern classes. The hierarchical partitioning tree generated by the above algorithm is shown in Fig. 6 where Roman numerals at each interval node indicate the sequence in which these nodes were generated. Also indicated are the number of samples and their respective labels present in the LIST (NODE) for all terminal nodes at the instant of termination of the algorithm. This indicates an average correct classification rate of 82.5 percent on the design set. The independent test set of 40 samples yielded a correct classification of 70 percent. This compares favorably with the classification performance of the nearest neighbor classifier on the same data. In addition, the following important observations can be made from the tree of Fig. 6.

1) Numeral categories 2, 5 and 6, 9 show a great deal of resemblence in the normalized Walsh sequency spectra domain. This is an expected result because of the nature of the sequency computation normalization step [9].

2) Numerals 0 and 8 also show a lot of similarity. However, this similarity is due to the size of the quantization grid, and is again expected.

3) Features A_5 and A_6 do not appear at all in the partitioning tree, while feature A_7 appears only once. Thus, it can be safely said that the first four features are more powerful, and possibly one can work with them only. This indicates the usefulness of the algorithm as far as feature selection and ordering are concerned.

Because of the small size of the design set, although not much confidence can be placed in the tree of Fig. 6, it is clearly evident that the proposed algorithm has a good capability of producing an effective partitioning and bringing out the similarity between patterns.

V. Confidence Bounds on Error Probability

Since the structure of the hierarchical classifier generated by the algorithm of Section III is similar to the empirical classifier of Henrichon and Fu [3], the bounds obtained by them are directly applicable here. Thus, the probability of error of the hierarchical classifier can also be expressed by the following inequality in addition to (5), i.e.,

$$P_e < \sum_{J=1}^{m} P(c_j)\, u_j$$

where u_j is the probability that a pattern comes from category c_j. u_j is a random variable having beta distribution as

$$u_j \sim B(q_j, n_j - q_j + 1)$$

where n_j is the total number of samples of category c_j in the design set, and q_j is the sum of misrecognized design samples from category c_j and the smallest number of terminal nodes of the hierarchical tree when its terminal nodes are relabeled as of category c_j or not c_j and merged to yield minimum nodes in the partitioning tree. Assuming feature independence, the upper bound on the probability of error can be determined to any degree of confidence by the following relationships:

$$P_r(P_e \leqslant \sum_{J=1}^{m} P(c_j)\, u_{j\,\text{crit}}) > \beta^m$$

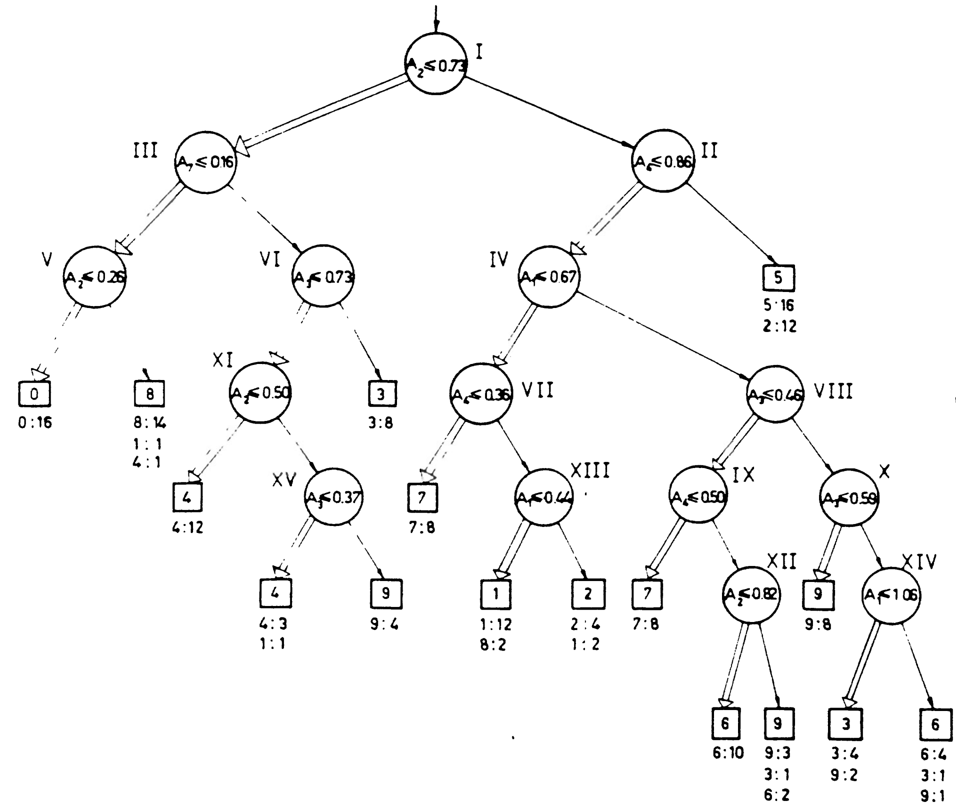

Fig. 6. Partitioning tree for handprinted numeral recognition. Numbers in the square boxes represent the terminal labeling. Numbers below the square boxes represent the label and number of design samples falling at that terminal node.

where $u_{j\,\mathrm{crit}}$ is chosen to yield

$$P_r(u_j \leqslant u_{j\,\mathrm{crit}}) \geqslant \beta.$$

VI. Conclusions

An algorithm has been proposed to partition the feature space for multifeature multicategory problems. The algorithm maximizes the mutual information gain at each paritioning step in the local sense, and therefore gives rise to a locally optimum decision tree. In case the globally optimum tree is desired, one can make use of available algorithms [10] which produce an optimal tree once the partitioning is specified. The resulting partitioning tree produced by the proposed algorithm can be used for classification as well as for data interpretation and pattern similarity analysis. Moreover, the classifier can also be implemented easily in hardware using threshold circuits.

References

[1] L. N. Kanal, "Problem-solving models and search strategies for pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 194–201, Apr. 1979.

[2] A. V. Kulkarni and L. N. Kanal, "An optimization approach to hierarchical classifier design," in *Proc. 3rd Int. Joint Conf. Pattern Recognition*, Nov. 1976, pp. 459–466.

[3] E. G. Henrichon and K. S. Fu, "A nonparametric partitioning procedure for pattern classification," *IEEE Trans. Comput.*, vol. C-18, pp. 614–624, July 1969.

[4] J. H. Friedman, "A recursive partitioning decision rule for nonparametric classification," *IEEE Trans. Comput.*, vol. C-26, pp. 404–408, Apr. 1977.

[5] E. M. Rounds, "A combined nonparametric approach to feature selection and binary decision tree design," *Pattern Recognition*, vol. 12, pp. 313–317, 1980.

[6] G. R. Dattatreya and V. V. S. Sarma, "Design tree design for pattern recognition including feature measurement cost," in *Proc. 5th Int. Conf. Pattern Recognition*, vol. II, Dec. 1980, pp. 1212–1214.

[7] R. M. Fano, *Transmission of Information.* Cambridge, MA: M.I.T. Press and New York: Wiley, 1963.

[8] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, part II, pp. 179–188, 1936; also in *Machine Recognition of Patterns*, A. K. Agrawala, Ed. New York: IEEE Press, 1977.

[9] I. K. Sethi and G. P. R. Saravarayudu, "Boundary approximation using Walsh series expansion for numeral recognition," in *Proc. 5th Int. Conf. Pattern Recognition*, vol. II, Dec. 1980, pp. 879–881.

[10] H. J. Payne and W. S. Meisel, "An algorithm for constructing optimal binary decision trees," *IEEE Trans. Comput.*, vol. C-25, pp. 905–916, Sept. 1977.

Fuzzy Tree Automata and Syntactic Pattern Recognition

EDWARD T. LEE

***Abstract*—An approach of representing patterns by trees and processing these trees by fuzzy tree automata is described. Fuzzy tree automata are defined and investigated. The results include that the class of fuzzy root-to-frontier recognizable Σ-trees is closed under intersection, union, and complementation. Thus, the class of fuzzy root-to-frontier recognizable Σ-trees forms a Boolean algebra. Fuzzy tree automata are applied to processing fuzzy tree representation of patterns based on syntactic pattern recognition. The grade of acceptance is defined and investigated.**

Manuscript received August 20, 1981; revised January 25, 1982.
The author is with the Department of Mathematical Sciences, Memphis State University, Memphis, TN 38152.

0162-8828/82/0700-0445$00.75 © 1982 IEEE