Lars Information Note 090174

THE DECISION TREE APPROACH TO CLASSIFICATION

Chialin Wu

David Landgrebe

Philip Swain

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

TR-EE 75-17

May 1975

The work reported in this report was conducted under sponsorship of NASA Grant NGL-15-005-112 and NASA Contract NAS9-14016.

LARS Information Note 090174

NASA CR-
141930

(NASA-CR-141930) THE DECISION TREE APPROACH TO CLASSIFICATION (Purdue Univ.) 198 p HC $7.00 CSCI 05E

N75-29528

Unclas
G3/43 31040

THE DECISION TREE APPROACH TO CLASSIFICATION

C.L. WU
D.A. LANDGREBE
P.H. SWAIN

The Laboratory for Applications of Remote Sensing

Purdue University, West Lafayette, Indiana

1974

THE DECISION TREE APPROACH TO CLASSIFICATION

Chialin Wu
David Landgrebe
Philip Swain

TR-EE 75-17
May 1975

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

The work reported in this report was conducted under sponsorship of NASA Grant NGL-15-005-112 and NASA Contract NAS9-14016.

TABLE OF CONTENTS

TABLE OF CONTENTS, cont.

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES, cont.

LIST OF FIGURES, cont.

LIST OF FIGURES, cont.

ABSTRACT

A class of multistage decision tree classifiers is proposed and studied relative to the classification of multispectral remotely sensed data. The decision tree classifiers will be shown to have the potential for improving both the classification accuracy and the computation efficiency. To explain these advantages, the problem of dimensionality in pattern recognition is discussed in some detail; two theorems on the lower bound of logic computation for multiclass classification are also derived. After introducing the method of uniquely specifying the decision tree structure, several approaches to the design of decision tree classifiers are discussed. Both interactive and automatic approaches are included. Emphasis of the discussion is placed on the automatic approach, i.e. the optimization approach. In this approach, two design strategies will be introduced: one focuses on designing classifiers with higher accuracy, the other on designing classifiers with optimal "overall

performance". Finally, experimental results on real data are reported, which clearly demonstrate the usefulness of decision tree classifiers.

CHAPTER 1

INTRODUCTION

1.1 The Decision Tree Classifier

The objective of this study is to develop a class of decision tree classifiers for multivariate and multiclass classification. The practical application of the proposed classifier is also investigated for pattern recognition problems encountered in multispectral remote sensing [1,2], where the data is gathered in digitized form in several spectral bands over a particular area of the earth under observation; the purpose of classification is to obtain information about the types of ground coverage in that area.

The conventional approach to multivariate and multiclass classification would be to perform tests on the unknown pattern* against all classes using a particular feature subset and then assign the unknown to one of these classes. The decision tree [3] approach classifies the unknown through a hierarchical decision procedure. That is, if after a decision is made, the outcome is not a terminal one, another decision will be made until a terminal decision is reached. This terminal decision determines to which class the unknown sample being tested belongs.

*In this work, the terms pattern, datum, and sample are used interchangeably.

In classifying multispectral remotely sensed data, a typical example of the decision tree is shown in Fig. 1.1, where an unknown datum (a ground resolution cell) is classified into the class water or bare soil through only one stage of decision (i.e. these two classes would be terminal decisions), however for the unknown to be classified into other vegetation classes it takes several stages of decision. In feature space, the idea of the multistage decision tree approach is to partition the feature space step by step, as shown in Figure 1.2. Here the circled numbers indicate the order of the decision boundaries to partition the feature space. These two figures are two simple examples to illustrate the functioning of the decision tree classifiers. More complex and realistic decision trees will be constructed in later chapters.

The reason to pursue this investigation of the decision tree approach is based on the advantages this approach may have. Three major advantages have been found, namely, the higher accuracy, higher efficiency and more meaningful interpretation of the classification scheme.

The obstacle to implementing the decision tree classifier is mainly the difficulty in designing the classifier structure. To find solutions to the design problem and to test their usefulness thus become the major work in developing the decision tree classifiers.

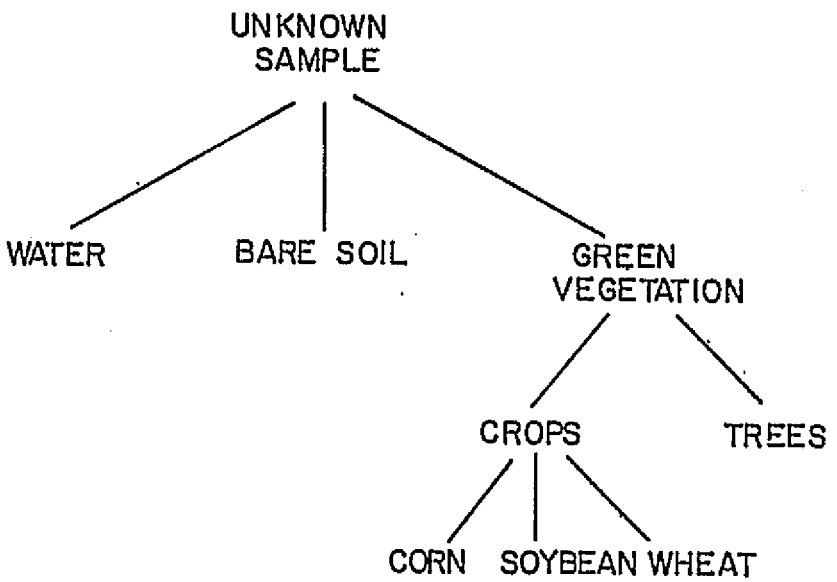


Figure 1.1 An Example of Decision Tree in Classifying Agricultural Data.

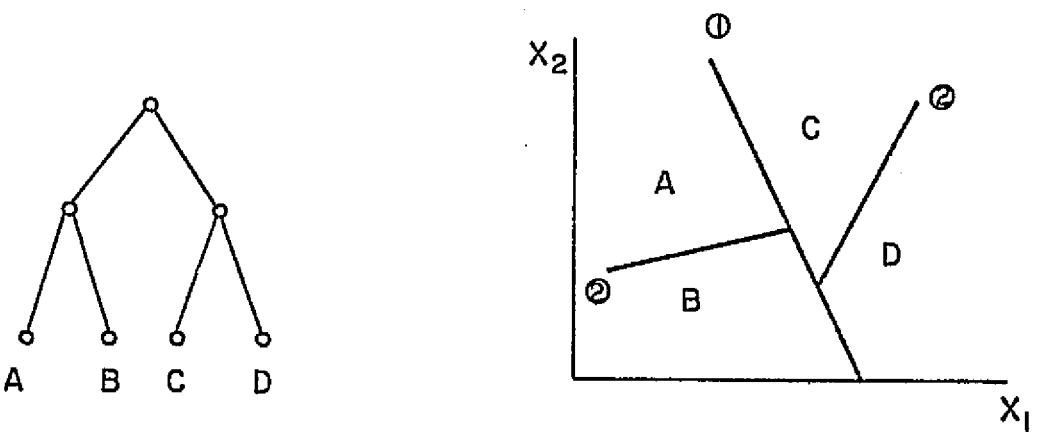


Figure 1.2 Feature Space Partitioning by Multistage Decision Tree.

1.2 A Review of Related Work

The decision tree classifier is just one type of multistage classifiers. A multistage (multileveled or layered) classifier can be defined as a classifier which may use more than one decision function in a sequential manner to classify an unknown sample into a class. The decision function (as will be used in later discussions) is defined as the mathematical formulation of a decision rule for simple or multi-hypothesis test. Classifiers which have only one decision function, such as the maximum likelihood classifier, are called single-stage classifiers.

Most of the literature of pattern recognition deals with single-stage classifiers and different types of discriminant functions. For a broad understanding of various pattern recognition techniques, the reader may refer to the books by Duda and Hart [4], by Fununaga [5] and by Meisel [6], also to the survey papers by Fu and Swain [7], by Ho a. d Agrawala [8], by Kanal [9], and by Nagy [10]. For multispectral pattern recognition problems, a very complete survey has been reported by Nagy [2].

For the particular case of multistage classifiers, the research work reported can be summarized into three categories. They are the sequential probability ratio test, the decision tree method and the perceptron method. Some important features of these methods will be briefly

introduced in the following paragraphs.

The application and generalization of Wald's sequential probability ratio test (SPRT) [11] for pattern recognition are described in the book by Fu [12]. In this method, observations are taken in a sequential manner. After taking each observation, a decision is made; and this decision determines whether the unknown sample is classified or another observation is necessary for classification. This sequential method is very useful for many practical problems where the observations are sequential in nature, and the cost of taking measurements is considered important.

A brief introduction to the decision tree method has been given in the beginning of this chapter. Decision tree classifiers so far reported in the literature are of the binary tree type [13,14], i.e. at each stage of decision there are only two possible outcomes.

Perceptron theory results from the study of neuro-dynamics. The engineering application of perceptron theory can be found in the books by Minsky and Papert [15], and by Nilsson [16]. A perceptron is a multiple-input threshold logic unit. A layered perceptron machine (as discussed in the book by Minsky and Papert) then consists of several level of perceptrons.

These three methods so far discussed are three important families of multilevel classifiers. Other proposals [17,18,19] can generally be fitted into, or considered as

a generalized form, of one of these three methods. A class of multistage decision logic worth mentioning is the decoding trees, e.g. Ref. [20,21]. These are in the form of binary trees, and are being studied extensively in the area of digital communication and information theory. Since the nature of this class is different from those classifiers where the received signals are physical observations of unknown samples instead of predesigned codes, the application context is somewhat different.

The sequential method and the decision tree method have the similarity that different feature sets can be used in later stages of decision in order to reach a final decision. The third method above is very distinct in this aspect, because new features are formed by a manipulation (linear combination with threshold) of the old features. The distinction between the sequential and decision tree methods is also clear. Considering the generalized sequential method (GSPRT [12]), the features are used in a sequential manner, and the number of possible decisions (which correspond to the classes retained for further consideration) for each stage can be varied according to different samples. For the decision tree method, the sets of features used along a decision path can be different from those of another path, and the number of possible decisions at each particular stage in a decision tree is fixed.

As far as the design procedure is concerned, for the perceptron method, the values of the coefficients (of linear combination) are usually obtained by learning, as proposed by Nilsson [16]. However, analytical procedures such as linear programming and extrema seeking can also be found in literature [22,23]. For the sequential method, the mathematical programming approach [24] is popular. Slager and Lee [25] proposed the game tree search approach to order features in implementing the sequential method.

For the decision tree method, early work by Mattson and Damman [13] laid the basic background for designing the tree structure. Meisel and Michalopoulos [14] suggested a two step approach to solve the design problem: the first step involved decision boundaries of a single variable to be found by a nonparametric method, while at the second step, dynamic programming was used to arrange these decision boundaries (or functions) into a binary tree decision-making structure. Both approaches have the drawback that the types of tree structures and discriminant functions are highly restricted (they must be binary tree structure with linear discriminant functions). Thus for the purpose of efficiently designing a good decision tree which is general enough to handle multivariate and multiclass data (for which nonlinear discriminant functions are usually involved in classification), several approaches to the design will be proposed in this report.

1.3 Summary of Contents and Contributions

In Chapter 2, the advantages of the decision tree classifier are discussed. Three major advantages are included; they are to improve the classification accuracy, to improve the computational efficiency and to provide convenience in applications.

In Chapter 3, the structure of the decision tree classifier and a method of its representation are specified. Notations adopted from graph theory are introduced for clearer explanation.

In Chapter 4, several approaches to design decision tree classifiers are proposed. Briefly, they are: the histogram approach, the sequential clustering approach and the optimization approach.

In Chapter 5, experimental results on real and simulated data are demonstrated. Finally, Chapter 6 concludes the whole study. Some analytical and experimental details are placed in Appendices, for the purpose of reducing disgression.

Since the application of the decision tree classifier to multispectral remote sensing data is emphasized, the assumption of multivariate normal data distributions which is often a reasonable assumption for remote sensing data [1,2] will be constantly used in later derivations involving data distributions.

The major contributions of this study are summarized as follows:

1) The derivation of several theoretical results on computation complexity for optimal classification, both feature and logic complexities considered.

2) The search approach to the design of decision tree classifiers, which includes two procedures for two different goals of decision tree optimization: one being the maximization of accuracy, another the maximization of "overall performance".

3) The development of a nonsupervised clustering procedure which is easy to use and effective in determining the associativity of points in clusters (when completely separable clusters can not be found).

Indeed, using a decision tree approach within the context of the multispectral remote sensing problem is new.

CHAPTER 2

NEED FOR A DECISION TREE CLASSIFIER

Several needs or potential advantages of the class of decision tree classifiers will be discussed in this chapter, through decision theoretical, computational efficiency, and application users' considerations. These needs stimulate the investigation of the decision tree classifier, and are discussed to some detail for the purpose of understanding what can be achieved by a decision tree classification procedure.

2.1 Decision Theoretical Considerations

The first need for the decision tree classifier originates from the dimensionality problem [Ref. 1,26,27; summarized in Ref. 4, pp. 66-73] which can be described as follows: there may be some feature subsets which are more effective than the complete set. In other words, the dimensionality problem implies that the error frequency for multivariate classification may not be a monotonically decreasing function of variable dimensionality. In two class classifications, the problem calls for an effective method for feature selection in which the optimal feature subset can be selected out of the complete feature set. For multiclass (more than

two classes) classification the situation is even more complicated. This is because optimal feature subsets for different subsets of classes may be different. Therefore, a conventional procedure which uses only one feature subset in all tests may not be optimal. The decision tree classifier which has the ability to classify different class subsets by using different feature subsets certainly has the potential to improve the classification accuracy.

The theoretical evidence for the dimensionality problem will be discussed, because of its importance to the selection of optimal dimensionality for classification.

2.1.1 The Dimensionality Problem

The dimensionality problem has been studied by many researchers [27] - [34]. To seek an understanding of this problem is important because the fact contradicts one's initial impression that in estimation, prediction or classification of stochastic systems the higher the observation dimensionality the better would be the results. And a solution to the problem or the need to obtain a reliable method to predict the optimal dimensionality is urgent. For multispectral remote sensing, such a solution will not only provide optimal feature selection for ground data processing but will also help in the selection of channels in designing on board sensor systems.

Generally speaking, the dimensionality problem is attributed to the insufficient number of training samples.

Error involved in density estimation accumulates as feature dimensionality increases, and if the accumulation of error is faster than the increase of separability, the dimensionality problem occurs. Among those reported work, the early work of Hughes [28] and its later developments [29] - [32] can be thought of as an approach to the explanation from a nonparametric point of view. This is similar to the explanation of the relationship between error rate, the size of the training set and the width of Parzen's window function [35] in a nonparametric classification approach. The explanation given by Wacker and Landgrebe [34] is of another nonparametric case, where the Euclidean distance measure is used for discrimination. And assuming a fixed signal-to-noise ratio in each dimension, it has been shown that the ratio of the means of between and within class distances decreases monotonically with dimensionality.

Consider the problem involved in parametric classification schemes. Allais [27] first derived the mean performance of the least square linear classifier. For the class of maximum likelihood classifiers with multivariate normally distributed data, not much work concerning the dimensionality problem has been reported yet. For the purpose of having a closer look, some derivations have been made here, which provide some quantitative explanation to the dimensionality problem in this particular circumstance.

Estimation of probability densities is involved in many practical classification problems. Assuming data of each class are of multivariate normal distribution, the statistical parameters may then be estimated in the following manner:

$$\hat{M} = \frac{1}{n} \sum_{j=1}^{n} X_j \tag{2.1a}$$

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{j=1}^{n} (X_j - \hat{M})(X_j - \hat{M})^T \tag{2.1b}$$

where X_i is a m-dimensional column vector with m the feature dimensionality, and n is the number of training samples. According to these parameters, the estimated conditional probability $\hat{P}(X|\omega_i)$ for a given class ω_i is expressed as:

$$\hat{P}(X|\omega_i) = N(\hat{M}_i, \hat{\Sigma}_i) \tag{2.2}$$

where $N(\cdot,\cdot)$ denotes the multivariate normal density functions and suffix i is added to the quantities in Eq. 2.2 to indicate the class designation of the estimated parameters. With the assumption of zero-one loss function and equal a priori probabilities, based on these estimated density functions, the Bayes decision rule for minimum risk can be written as:

$$\hat{g}_k(x) = \min_{1 \le i \le N} \hat{g}_i(x) \longrightarrow X \in \omega_k \tag{2.3}$$

where $$\hat{g}_i(x) = -\log \hat{P}(X|\omega_i) \tag{2.4}$$

and N is the total number of classes to be classified. Again, a hat is used for the quantity $\hat{g}_i(x)$ to indicate that it is also an estimated quantity. Since the true value of $g_i(x)$ gives the optimal result for classifying unknown samples, any deviation of $\hat{g}_i(x)$ from $g_i(x)$ certainly degrades the result. The total amount of degradation expressed by the increase in error rate in N-class classification is bounded above by the sum of degradations of $\binom{N}{2}$ two-class classifications each being a class pair of the N classes to be classified [33].

Considering the degradation for two-class classification, the variance of the difference of true and estimated likelihood ratios r_{12} and $\hat{r}_{12}$ will be examined first, where the ratios are defined as follows:

$$r_{12} = \log \frac{P(X|\omega_1)}{P(X|\omega_2)} \tag{2.5a}$$

$$\hat{r}_{12} = \log \frac{\hat{P}(X|\omega_1)}{\hat{P}(X|\omega_2)} \tag{2.5b}$$

The mean square error of r_{12} is expressed as

$$V[\Delta r] = \underset{X,\hat{\Omega}}{E}[(r_{12}-\hat{r}_{12})^2] \tag{2.6}$$

where the squared quantity is averaged over the distributions of sample points X and the estimated parameters given in Eq. 2.1. With the assumption given by Eq. 2.7

$$n_1 = n_2 = n \tag{2.7a}$$

$$n > m \tag{2.7b}$$

$$\Sigma_1 \doteq \Sigma_2 \tag{2.7c}$$

where n_i is the number of training samples for class ω_i, and m is the feature dimensionality, an approximation of V[Δr] in Eq. 2.6 is evaluated and is shown in Eq. 2.8 (the detailed derivations are placed in Appendix A)

$$V[\Delta r] \doteq \frac{1}{4n}[2m^2 + 20m + 2mD + 14D + D^2] + 0(\frac{1}{n^2}) \tag{2.8}$$

where n is given by Eq. 2.7a, D is the divergence of two multivariate normal distributions, which is expressed as

$$D = \frac{1}{2}tr[\Sigma_1-\Sigma_2][\Sigma_2^{-1}-\Sigma_1^{-1}] + \frac{1}{2}[M_1-M_2]^T[\Sigma_1^{-1}+\Sigma_2^{-1}][M_1-M_2] \tag{2.9}$$

Eq. 2.8 is an approximate expression. If the variances Σ_i are known, the exact problem-averaged expression for V[Δr] is as follows:

$$V[\Delta r]_{\hat{\Sigma}_i = \Sigma_i} = \frac{3m + 2D}{2n} + \frac{2m - 3m^2}{2n^2} \tag{2.10}$$

From both Eq. 2.8 and 2.10, it is noted that the problem averaged variance of Δr increases with dimensionality m. However, as m increases, with more features the class separability does not decrease. This implies that classification accuracy may be improved as m is increased; nevertheless it is also clear that the dimensionality problem occurs if the first effect overrides the second. An expression for approximating the overall inference of these two effects is given by Eq. 2.11 (which is an exact expression for the case with equal covariances $\Sigma_1 = \Sigma_2$, and is a rough approximation otherwise as explained in Appendix A)

$$\varepsilon = \text{erf}\{-\frac{1}{2}(\frac{1}{D} + \frac{V[\Delta r]}{D^2})^{-1/2}\} \qquad (2.11)$$

where ε is the error rate for two-class classification, D is the Divergence given by Eq. 2.9 and erf(x) is expressed as follows:

$$\text{erf}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{\alpha}{2}} d\alpha \qquad (2.12)$$

Simulated data sets which were generated with $\Sigma_1 = \Sigma_2$ have been used to test the validity of Eq. 2.11, and the results are given in the beginning of Chapter 5. The dimensionality problem in real classification problems will also be shown in that chapter.

2.1.2 Discussion

The existance of the dimensionality problem for pattern recognition with multivariate normal distributions is explained in the previous subsection. The remaining question is how to find optimal feature subsets for different class subsets. Although Eq. 2.9 and 2.11 have shed light on the theoretical prediction of optimal dimensionality, practical difficulties still exist.

Basically there are two difficulties: One is that the divergence value "D" calculated from the estimated parameters by using Eq. 2.9 is not always close to its true value. Although Eq. 2.1a and 2.1b are expressions for unbiased estimators for the mean and covariance matrix, Eq. 2.9 is not an unbiased estimator for D. And the deviations can be large; some experimental results are shown in Chapter 5.

The second difficulty is that in the case with unequal covariances, Eq. 2.11 is not a good approximation of the error probability. It is known from past experience in multispectral pattern recognition, that in classifying a pair of spectral classes based on a limited number of training patterns the effective number of spectral features can be four or less, and this number will be used for maximum feature dimensionality in most of the experiments given in Chapter 5.

2.2 Computation Efficiency Consideration

As often cited, an advantage of the multistage decision procedure (cited in several reports [12,13,14]) is higher computation efficiency. These multistage procedures reduce either the number of measurements or the number of tests necessary to reach a terminal decision. As an example, it has been shown [36] that for a two-class classification the sequential probability ratio test (SPRT) [11,12] with a fixed stopping boundary (specified by a given error rate) is optimal in the sense of minimizing the average number of measurements. It should be mentioned that this does not apply to the generalized sequential method (GSPRT) for multiclass classification. In a decision tree procedure the feature subset used at each stage can be designed according to the class separability at that stage. For different patterns to be classified, the sequences of feature subsets used may not be the same (following different paths in a decision tree). Thus the use of features can be more flexible than in the sequential method, making the decision tree procedure more favorable than the sequential method as far as optimal use of feature complexity is concerned.

Looking at the economic aspect of classification of multispectral data, after they are gathered, cost of computation is the major expense involved. The problem then is reducing this cost without trading off (loosing)

optimal classification results. Since this cannot be achieved simply by reducing the number of features of an optimal classifier, the only alternative is to try to reduce the number of tests.

Two theorems on the lower bound of the number of tests required for optimal classification results have been derived and they will be given later in this section.

Now for a closer look at the definition of the term "test". In multiclass classification, a test is defined as a comparison of the likelihood functions (or discriminant functions) of a pair of classes. According to this definition, in a conventional maximum likelihood procedure for N class classification, the number of tests required to classify a pattern would be N-1, since N-1 comparisons are involved. Actually, with the same amount of classification error the number of necessary tests on the average can be reduced. The lower bound on the number of tests is given by the following two theorems:

Theorem 2.1 Assuming P_i is the probability that a pattern belongs to class ω_i, and that successive patterns are statistically independent, for N-class classification, the expected number of tests E[U] necessary to classify an unknown pattern correctly satisfies:

$$E[U] \geq -\sum_{i=1}^{N} P_i \log_2 P_i \qquad (2.13)$$

Before proving Theorem 2.1 , Lemma 2.1 will be stated first.

Lemma 2.1 If each class designation of a sample can be uniquely specified by m binary bits, then there exists a sequence of m tests to classify a sample into one of those classes.

The proof of this Lemma is as follows: In each test the outcome can be one of the two possibilities, thus the result of a test can be represented by a single binary bit. After a sequence of m properly designed tests performed on a sample of unknown class, the result is a m-bit word of class designation, so the unknown sample is classified.

With the above Lemma, Theorem 2.1 can be proved with relative ease. Notice the right hand side of Eq. 2.13 is the entropy H [Ref. 37, p. 50] of class information, which according to Shannon's theorem on source coding [Ref. 37, p. 54; Ref. 38, p. 43] equals the average number of bits per source letter (with length of sequence approaching infinity) required to specify a sequence of letters efficiently (only one source sequence can be assigned to each code sequence). With Lemma 2.1 we know the effective average number of tests to classify a sample is H, i.e. E[U] = H. Since H is for the most efficient coding, this leads to the fact that E[U] can not

be less than H for correct classification. Thus, E[U] must be greater than or equal to H for correct classification, and this proves Theorem 2.1.

If one is willing to sacrifice accuracy to gain efficiency (by means of reducing the number of tests), for a given error rate, the theoretical limitation on test efficiency is provided by the following theorem:

Theorem 2.2 Assuming P_i is the probability that a sample belongs to class ω_i, and that successive samples are statistically independent, for N-class classification, the expected number of tests necessary to classify a sample of unknown class with expected error rate $\varepsilon (\leq 1 - \underset{1 \leq i \leq N}{\text{Max}} P_i)$ satisfies:

$$E[U] \geq \underset{P(i,j)}{\text{Max}} \left[-H_{\bar{\varepsilon}} - \sum_{i=1}^{N} P_i \log_2 P_i \right] \tag{2.14}$$

subject to the constraint $\bar{\varepsilon} \leq \varepsilon$

with $$H_{\bar{\varepsilon}} = - \sum_{i,j}^{N} P(i,j) \log_2 P(i|j) \tag{2.15}$$

and $$\bar{\varepsilon} = \sum_{\substack{i,j \\ i \neq j}}^{N} P(i,j) \tag{2.16}$$

where $P(i,j)$ is the probability of joint occurrence that sample X belongs to class j but is classified into class i, and $P(i|j)$ is the conditional probability

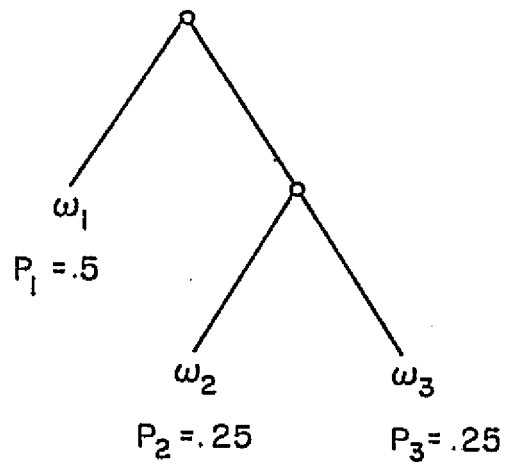
of the joint occurrence stated above.

The proof of this theorem is as follows: Notice the right hand side of Eq. 2.14 is by definition the rate-distortion function [Ref. 37, p. 112; Ref. 38, p. 444] with 0-1 distortion measure. Since this is the minimum rate for source coding with a given distortion measure which in our case corresponds to ε in Eq. 2.16, the number of tests which equals to the code rate according to Lemma 2.1 then can not be less than this minimum rate. Thus Theorem 2.2 is proved.

The theorems stated above are the theoretical limitation of the number of tests for multiclass classification. In practical problems these lower bounds usually can not be attained. However, from these theorems it is clear that the class of decision tree classifiers has the capability of achieving these limits*. An example is shown in Fig. 2.1, where the efficiency of a decision tree procedure is compared with the efficiency of a one stage conventional procedure. As one may observe in this ideal case the lower bound on the number of tests is achieved by the decision tree procedure. For real cases, besides the fact that some classes can be classified by

---

*This statement is true if U^*, the lower bound of E[U], is greater than or equal to one. If U^* is less than one, a type of block classification schemes which classify several samples together will have the possibility of achieving these lower bounds, but this scheme will not be discussed in this report.



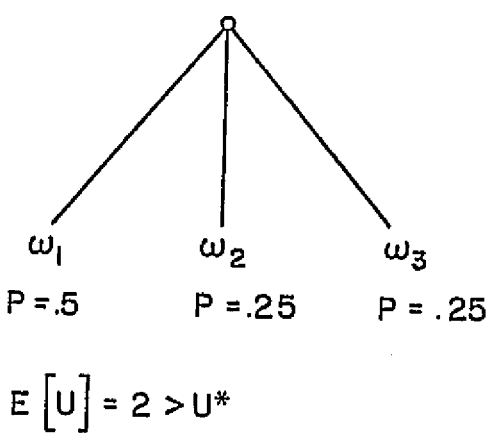


Figure 2.1 A Hypothetical Example Illustrating the Classification Efficiency of the Decision Tree Approach.

using a lower number of features, the reduction of tests is also expected in a decision tree procedure. This shows quite clearly one of the advantages of the decision tree classifier.

2.3 Application Oriented User's Consideration

Using digital computer techniques to analyze remotely sensed data has been referred to as the "numerically-oriented systems" approach [39], which together with the "image oriented systems" approach make up the two major trends in analysing remotely sensed data. Using the image oriented approach, in determining the extent, location and/or condition of the resources, one tends to follow a kind of logical hierarchy. An example is cited from the work by Hoffer [40]; it is shown in Fig. 2.2.

Upon applying this concept to the numerically oriented approach, a multistage classifier such as the decision tree classifier will be more desirable than a one-stage conventional classifier. Not only is a multistage classifier more efficient, but it is also more flexible in adapting the concepts of the image-oriented approach.

Once an objective and nonsupervised design procedure for a multistage classifier is obtained, some feedback from the numerically-oriented approach to the image-oriented approach can be expected. For example information gained in the numerically-oriented approach, such as the separability

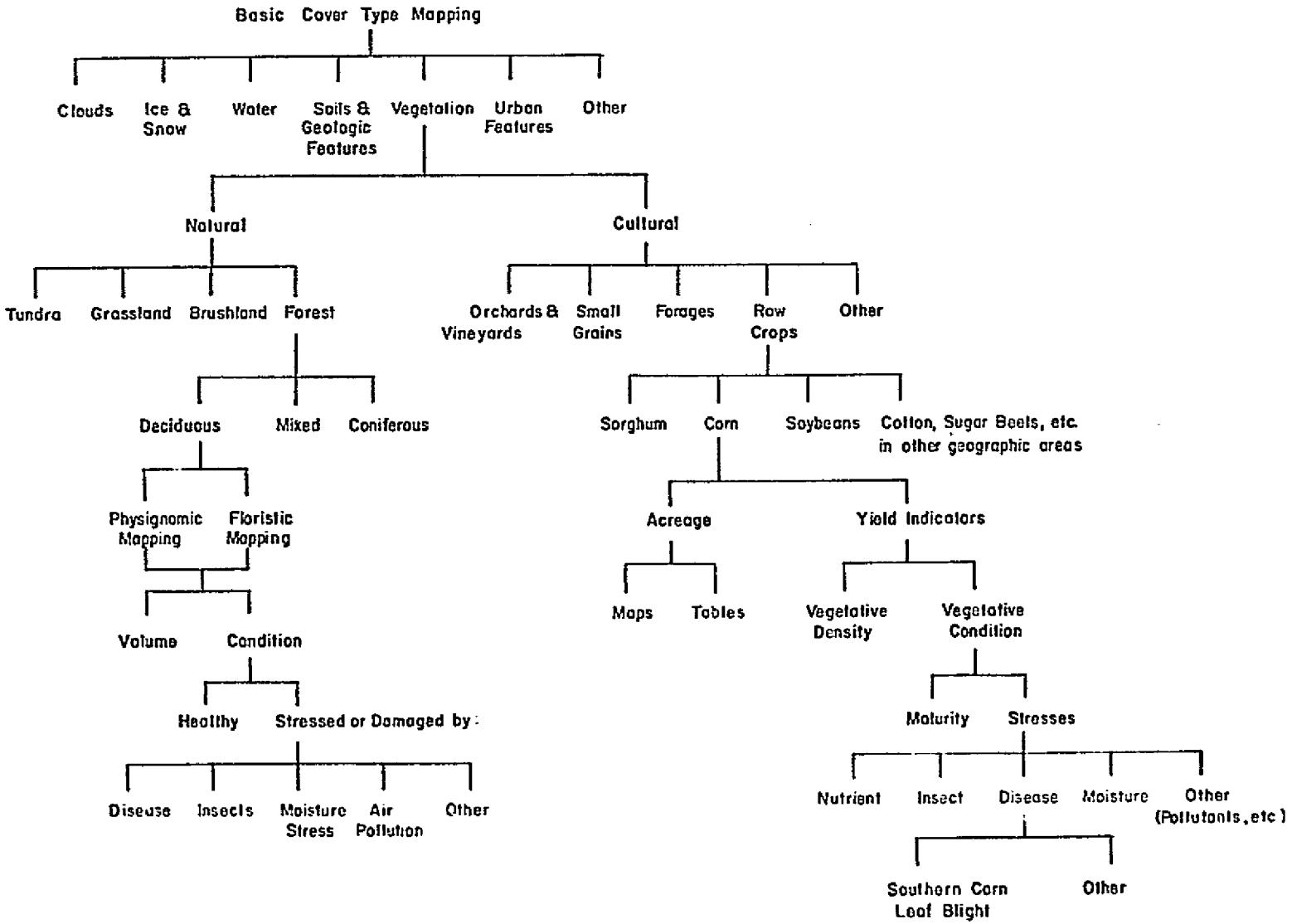


Figure 2.2 An Earth Resources Data Analysis Sequence for Selected Cover Types, Based upon Spectral Characteristics and User Requirements [40].

studies in different spectral and temporal circumstances and the choice of effective decision hierarchy structures, can be helpful to the image oriented users.

Through the above discussions, it is clear that the decision tree classifiers provide the users a better approach to classification than the conventional one stage approach. It is better in the sense that it can be more accurate, and/or more efficient.

CHAPTER 3

THE DECISION TREE CLASSIFIER

Before discussing the details of the decision tree classifier, it is desirable to define the term "tree". Applying the terminologies of graph theory, a simple definition of "tree" is stated as 'a connected graph* without cycles' [41]. Or according to Nilsson [49], a tree is a graph each of whose node has a unique ascendant node, except for the starting node which is called the root node. A tree thus defined has the property that a path from the root node to any given node is unique. In pattern recognition, the decision tree procedure corresponds to the partitioning of the feature space into different regions by a fixed ordering of the decisions. The property of a tree mentioned above is desirable because it implies that the mapping of a decision to its associated region in feature space is unique and the reverse is also true. Other useful terms are "terminal node" and "nonterminal node". A

*Strictly speaking, a "graph" G(N,C) is a set of elements N and a collection C of unordered pairs (a,b) of elements of N. An elements of N may be called a "node" or "vertex" of the graph, while the pair (a,b) is called an "arc" or "edge" of the graph. Other notations like "cycle" and "path" are also defined in Ref. [41].

terminal node is one that has only one ascendant node, while a nonterminal node has both ascendant and descendant nodes. In the decision tree classifier, a terminal node corresponds to a terminal decision i.e. the decision-making procedure terminates and the unknown being classified is assigned to the class of that node. However, a nonterminal node is an intermediate decision; another stage of decision will be made and its immediate descendant nodes represent the possible outcomes of that decision.

Using these concepts, the classification in a decision tree procedure follows a path in the tree, which starts from the root node and ends at a terminal node.

To specify a decision tree uniquely, two sets of information are necessary. One set tells how the nonterminal and terminal nodes are linked while the other specifies the decision functions of all the nonterminal nodes. For a tree with a simple structure, such as a binary tree with univariate linear discriminant functions for nonterminal nodes, a set of n-tuples (which is a combined description of the above two sets of information) can be used to specify it uniquely [14]. For cases where the tree structures are complex, i.e. the number of immediate descendant nodes of a nonterminal node is not fixed, and also where the decision functions are complicated, e.g. they may be multivariate and quadratic, it is desirable to treat these two sets of information separately. The method to

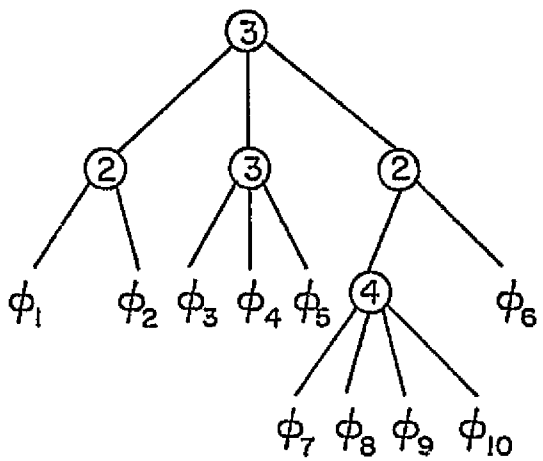
characterize this information will be discussed in the following section.

3.1 Tree Structure Information

By assigning different symbols to nonterminal and terminal nodes, the tree structure can be coded into a string. The rule for encoding is breadth first, from left to right and then top to bottom. The reason for following the rule of breadth first is because in describing each decision function (of each nonterminal node) in a decision tree, it is convenient to pack the statistics parameters (which correspond to probability densities associated with the immediate descendant nodes) together, and the rule of breadth first serves this purpose.

Two sets of symbols will be used for coding to represent the terminal and nonterminal nodes respectively. They are $\{\phi\}$ and $\{N_i\}$. In the first set, there is only the symbols "ϕ"; all terminal nodes are represented by it. In the second set, there are many symbols; each symbol "N_i" is associated with a value i (integer greater than one) being equal to the number of immediate descendant nodes that the nonterminal node has.

A simple example is shown in Fig. 3.1, where the symbol 'ϕ_i' (the subscript is used to indicate its relative position in S) stands for a terminal node, and the numbers stand for the nonterminal nodes. The string S which is the encoding of



S= 3 2 3 2 ϕ_1 $\phi_2\phi_3\phi_4$ ϕ_5 4 ϕ_6 $\phi_7\phi_8\phi_9\phi_{10}$

Figure 3.1 A Tree with Its String.

the tree according to the rule of breadth first is shown at the bottom of Fig. 3.1.

When a tree structure is given, its associated string S can be found; from this string S, an identical tree structure can be reconstructed by a left to right scanning of the string: Each symbol of the string corresponds to a node of the tree. For the first symbol, a root of the tree is formed, and a number i (i = 3 in the example) of descending branches are drawn from the root. Place the next successive i symbols at the ends of those branches. If there are additional nonterminal nodes in the string (represented by nonterminal symbols), corresponding numbers of descending branches will be drawn from them. This step repeats, i.e. place symbols at ends of branches (following the rule of breadth first) and draw branches for nonterminal nodes, until no more symbols are left in the string. For computer processing, after a left to right scanning of the string "S", a set of arrays are generated which tell how the nodes are linked.

It has been found that the set of strings "S" which are codes of tree structures, with the rule for coding described earlier, form a context free language (The definitions of language and grammar can be found in many books dealing with formal languages and automata theory, e.g. Ref. [42]). Two relevant theorems are stated below:

Theorem 3.1 The set of strings which are codes of node structures of trees, and are coded by following the rule described above with symbol set $\{\phi, N_i\}$, forms a context free language L(G). The associated grammar G is given below:

$$G = \{V_N, V_T, P, S\}$$

with $V_N = \{S\}$

$V_T = \{\phi, N_i\}$

P: $S \rightarrow N_i S^i$

$S \rightarrow \phi$

where N_i and i have been introduced earlier, and S^i is a string of i consecutive "S".

Proof: The first production rule implies that when a nonterminal symbol N_i is generated, i other new symbols (represented by S) are also generated and are placed to the right of N_i. This leads to the fact that in the tree reconstruction as described earlier, when a nonterminal node (corresponding to N_i) is constructed, there are i symbols always available as immediate descendant nodes. Since this is true for all nonterminal nodes, and the second production rule does not change length of the string,

every nonterminal node N_j has j immediate descendant nodes (represented by j symbols in S) and every node (except the root) has an ascendant node. Thus each string of L(G) can be used to reconstruct a tree and is equivalent to the tree. That is, the set of language L(G) is identical to the set of codes for tree structures. And according to the production rules, L(G) is context free. This proves the theorem.

Theorem 3.2 The corresponding pushdown automaton M which accepts this set of strings L(G) in Theorem 3.1 is

$$M = [\{q_0\}, \{\phi, N_i\}, \{Z\}, \delta, q_0, Z, \psi]$$

with δ: $\delta(q_0, \phi, Z) = \{(q_0, \varepsilon)\}$

$$\delta(q_0, N_i, Z) = \{(q_0, Z^i)\}$$

where Z^i is i consecutive Z's in the pushdown stack.

Proof: The grammar G_1 of the language $L(G_1)$ accepted by M can be derived [Ref. 42, p. 76] to have the production rules:

$$P_1: \quad S \rightarrow [q_0, Z, q_0]$$

$$[q_0, Z, q_0] \rightarrow \phi$$

$$[q_0, Z, q_0] \rightarrow N_i [q_0, Z, q_0]^i$$

Equating the symbol $[q_0, Z, q_0]$ to S, the above production rule P_1 is identical to the production rule P of grammar G in Theorem 3.1, and also V_N, V_T of G_1 are the same as V_N, V_T of G. This leads to the statement that G_1 is equivalent to G, thus the theorem is proved.

These two theorems also imply the one-one correspondence of a tree structure and a string. Briefly, this is because both the grammar G and the automaton M described above are deterministic.

3.2 Decision Function Information

For classifying remotely sensed data, as mentioned previously in Chapter 1, the maximum likelihood classifier with normal density functions will be used. The classification scheme then is parametric; for each stage the decision function can be uniquely specified by a set of statistical parameters. The parameters represent the density functions of various classes, and they can be estimated from a set of training samples. In a sequence of decision stages the original densities of the classes are always used, in spite of the fact that certain classes have been

partitioned in earlier stages (and more than one terminal node refers to each of those classes). The reason for not updating the original statistics for partitioned data is to maintain the decision boundaries of the conventional one stage maximum likelihood classifier which is considered Bayesian optimal for a zero-one loss function. In case an outcome of a decision corresponds to a collection of classes, the pooled statistics of some of these classes may be used in the parametric decision function of the succeeding stage.

With the decision function for all the nonterminal nodes described along with the string which gives the structure of the nodes, the decision tree procedure is completely and uniquely specified.

CHAPTER 4

APPROACHES TO THE DESIGN OF THE DECISION TREE CLASSIFIER

Several approaches to the design of an effective decision tree classifier will be discussed in the following sections. In the histogram approach and sequential clustering approach, interaction with the analyst is necessary to design a good decision tree structure. The optimization approach is the most sophisticated but the least amount of interaction is needed. For the purpose of maximizing the accuracy (when the dimensionality problem might occur) or the overall performance, two design procedures will be introduced in the section on the optimization approach.

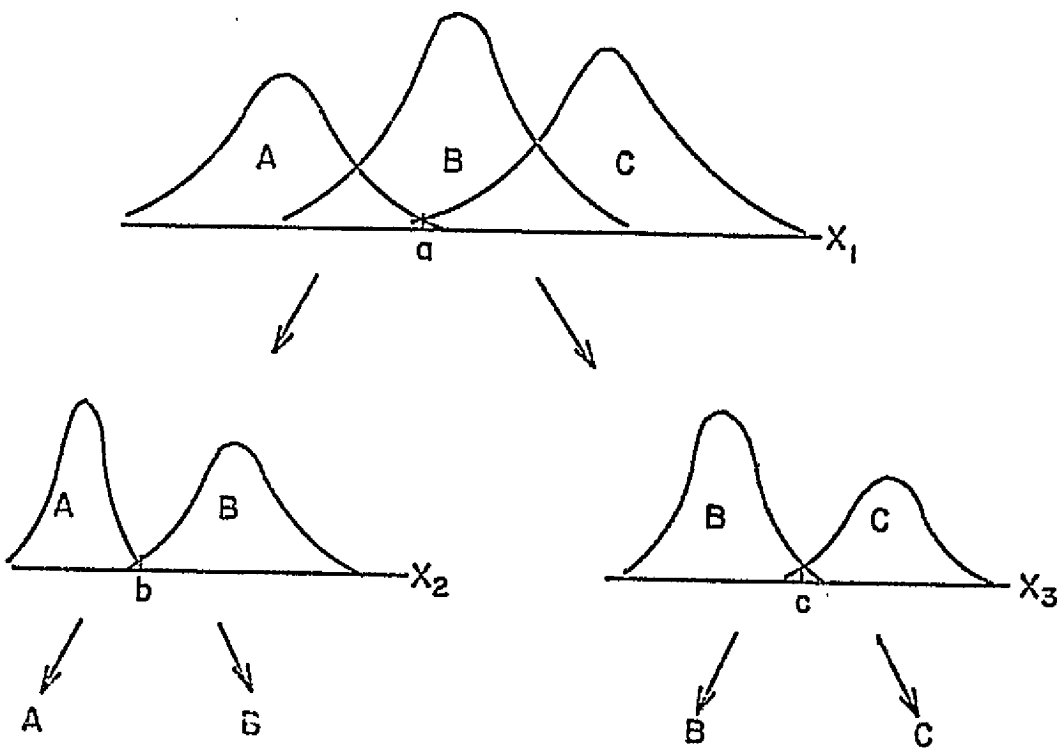
4.1 The Histogram Approach

The strategy of the historgram approach to decision tree design is very basic and is similar to the method in the paper of Mattson and Damman [13]. The approach can be described as follows: The histogram of training data of all classes is plotted on each feature dimension with the same scale. By observing the histograms one can find decision boundaries (or threshold values) to partition those classes into several groups. If a group contains more than

one class, the same procedure is repeated until all classes are uniquely classified. When this state is achieved the design is complete.

A simple example is illustrated in Fig. 4.1, where a decision tree classifier is constructed for three classes with three features. For multispectral data, coincident spectral plots* too can be another source of information from which the decision tree classifier can be designed. In these plots the means and standard deviations of all classes (assuming each class of data is of normal distribution) are plotted with the same scale, so that the decision boundaries (or threshold values) can be observed. An example of the coincident spectral plot is shown in Fig. 4.2 where a character indicates the class and locates the mean of that class with respect to that dimension. For the five classes shown in this figure, a two stage decision tree procedure is designed. A single feature $\{f_4\}$ is used for the first stage; class, {A} and {D} are two representative classes for two groups. In the second stage, since no single feature can separate the classes in two groups satisfactorily, a maximum likelihood classifier with all features will be used for terminal classification. When the maximum likelihood procedure is used at each

*Output of the statistics processor of LARSYS [43], a software system for remote sensing pattern recognition.



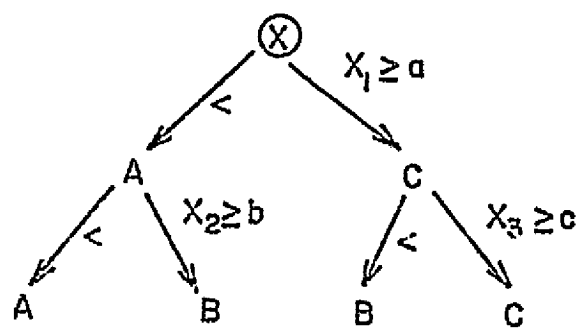


Figure 4.1 A Simple Example of the Histogram Approach to Design a Decision Tree Classifier.

LABORATORY FOR APPLICATIONS OF REMOTE SENSING
PURDUE UNIVERSITY

COINCIDENT SPECTRAL PLOT (MEAN PLUS AND MINUS ONE STD. DEV.)

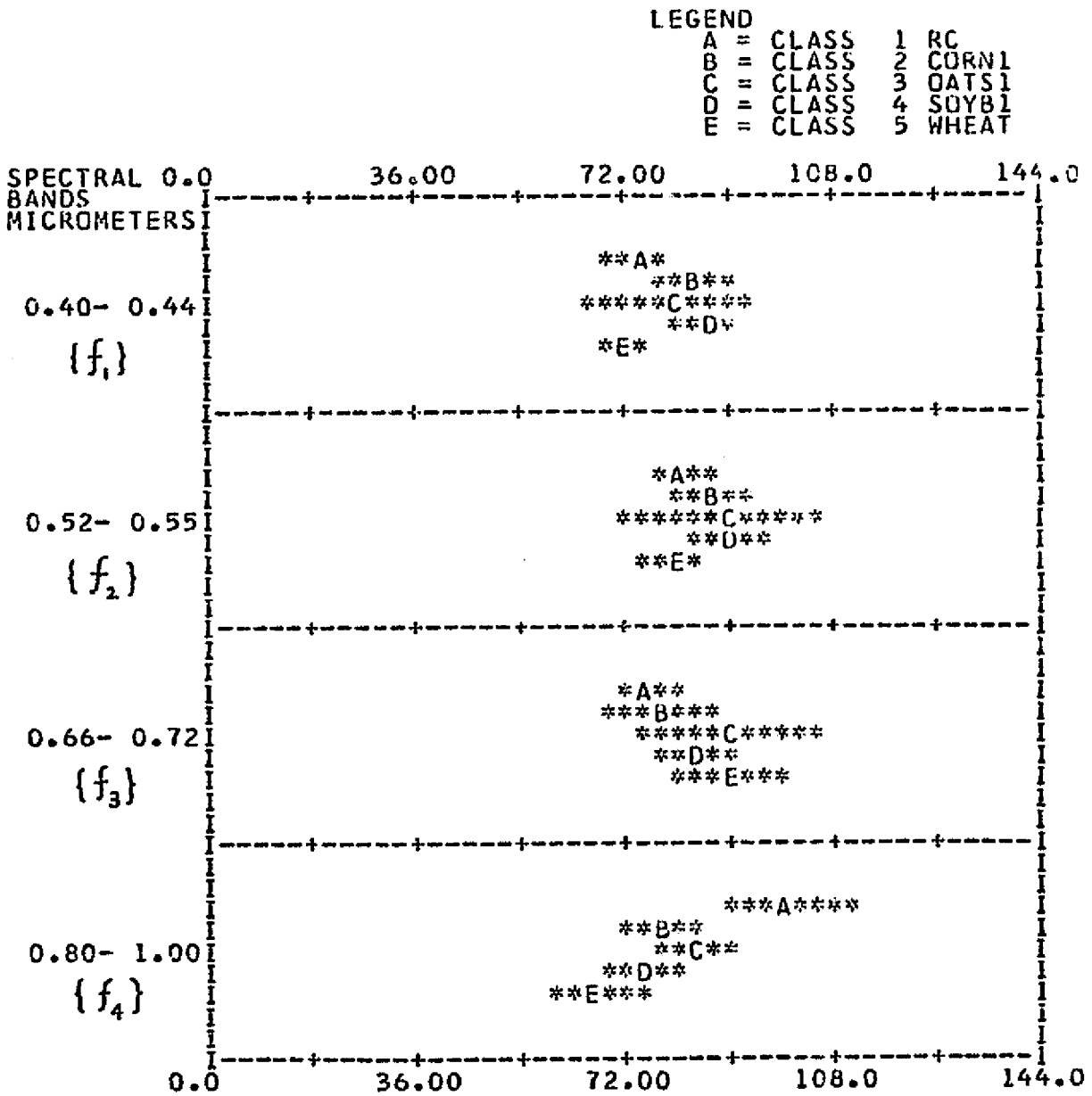


Figure 4.2 A Coincident Spectral Plot of Five Classes.

stage, a decision rule can be specified by a feature subset and a class subset, assuming the statistics for all classes are given. A set up of the classifier designed for digital computer application is shown in Fig. 4.3.

The performance of the classifier designed by this approach is subject to the experience of the designer, yet this approach provides a convenient and basic method for designing a decision tree classifier.

4.2 The Sequential Clustering Approach

In the sequential clustering approach, a decision tree is designed through successive stages of clustering. Actual class information is necessary to determine whether the training samples have been properly clustered into the required information classes. The class information of the multispectral remotely sensed data, usually referred to as "ground truth" is generally represented by two dimensional maps (e.g. USGS Topographic Maps) and aerial photographs. The cluster maps (results) obtained from the computer are compared with the conventional maps of photographs and this is where human interaction is involved.

An example of the procedure for this approach is illustrated in Fig. 4.4. Here a scene is first clustered into three classes A. B and C. After this result (clustering map) is compared with the ground truth map, class A, C are further clustered into three and two subclasses

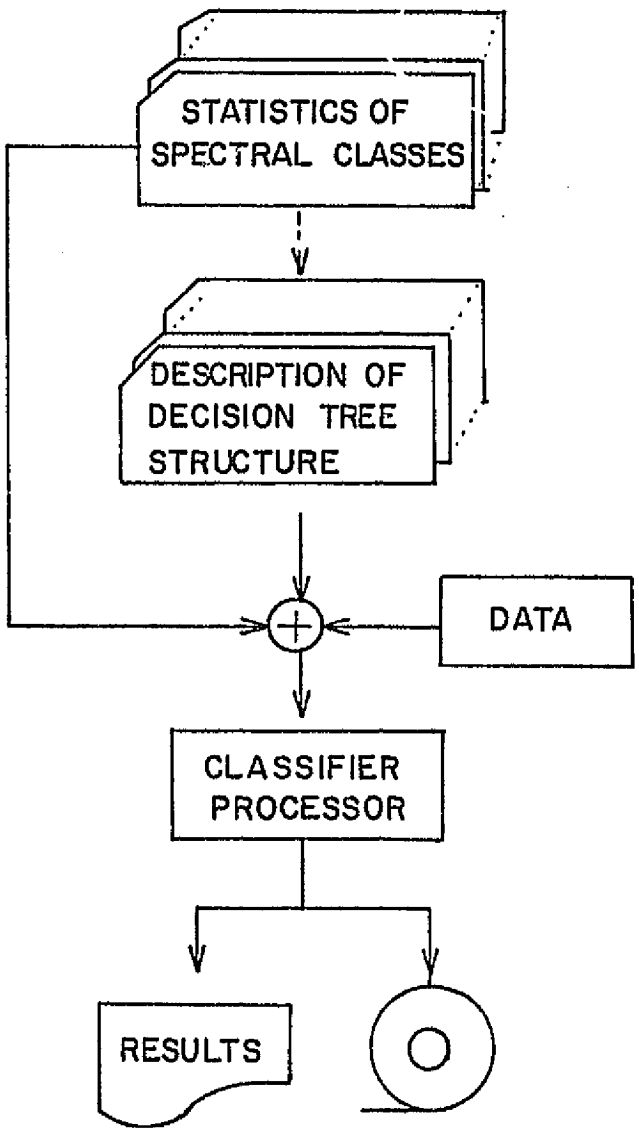


Figure 4.3 Input/Output Set Up of Decision Tree Procedure with Histogram Approach.

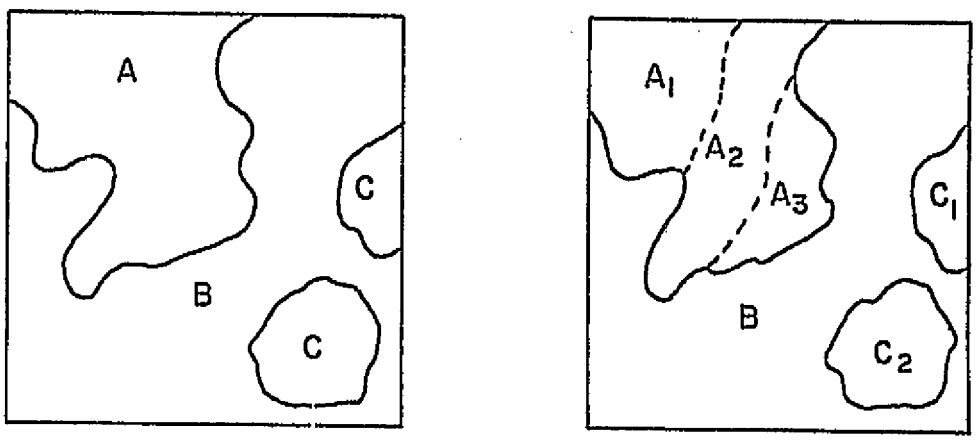


Figure 4.4 Multistage Clustering of a Geographic Area.

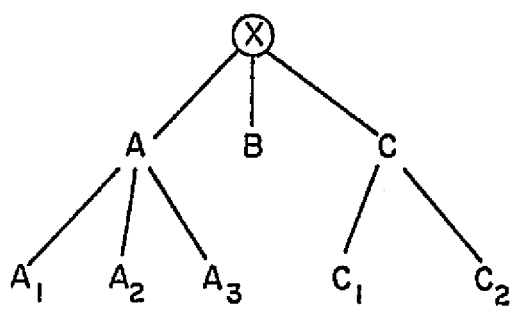


Figure 4.5 Node Structure of the Decision Tree Classifier Designed in Fig. 44.

respectively. Corresponding to this sequence of clustering, the structure of the decision tree classifier is shown in Fig. 4.5.

It is common for the first cluster map to have some mixture classes. Further subdivision of these subclasses allows them to be grouped together to provide the correct classes. Through this interactive approach, multistage clustering of a given area can lead to conformity with the map or photograph.

By utilizing the clustering algorithm [43] of LARSYS, the probability densities of the classes and subclasses can be approximated by multivariate normal distributions. The remaining classification problems thus become parametric. The maximum likelihood decision rule can easily be incorporated in the decision tree designed to classify unknown samples. Consequently, it is required that after each stage of clustering, the statistics of the clusters be calculated. These statistics will be necessary to specify the discriminant functions in the decision tree procedure. The set up of the decision tree procedure designed by this approach for digital computer implementation is shown in Fig. 4.6. This set up differs slightly from the previous one (Fig. 4.3) in the sense that each decision function is directly represented by the statistics of the classes (or clusters) to be classified.

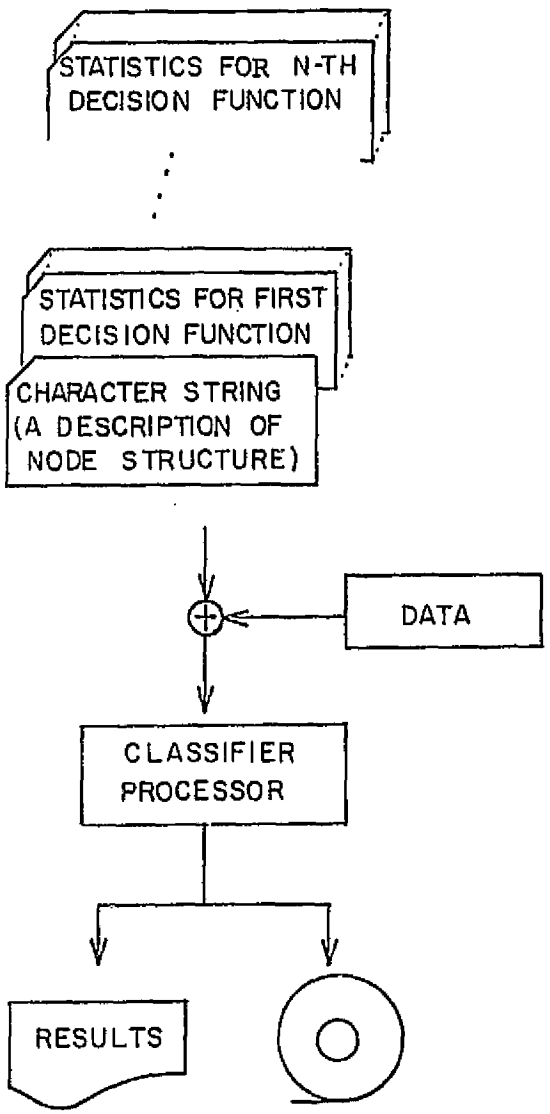


Figure 4.6 Input/Output Set Up of Decision Tree Procedure with Sequential Clustering Approach.

Although in the above discussion nonsupervised clustering is used to obtain the structure of each stage of classification, the supervised training method can also be used. For two dimensional imagery data, the spatial properties of the classes would be a major determining factor as to which method would be more appropriate.

Another major advantage of this approach is that after observing the classification results, if there is need for a certain class to be reclassified, this approach can be used to construct a multistage classifier which is used to classify data again; a change in the results will be observed only in those samples classified in that particular class. Thus, the advantage of the multistage classifier over the conventional one stage approach, where the classification results of other classes may also be changed by the addition of unrelated subclass to the classifier, is obvious.

4.3 The Decision Tree Optimization

The study in this section is aimed at a systematic approach to design a good decision tree classifier. The nature of the design problem would be very similar to that of the histogram approach introduced earlier. With sets of training samples of known classes being given, the design procedure will construct a good decision tree to classify unknown samples into these classes. The method described in the first section provides the fundamental idea of how to

solve this design problem. However, what is left unanswered is the question of how good the designed tree is when compared to other alternatives.

With the generality of the tree structure already discussed in Chapter 3, even for a small number of classes and features, numerous different tree structures can be constructed. Suppose there are m nonterminal (or decision) nodes in a given node structure and n features are available for classification. For each nonterminal node, $2^n - 1$ feature subsets can be used for the decision function. Thus, for this given nodes structure, $(2^n-1)^m \triangleq 2^{n \cdot m}$ different arrangements for the decision functions can be found. For the total number of possible trees N, we shall have:

$$N \triangleq \sum_{i=1}^{K} 2^{n \cdot m_i}$$

where K is the number of different nodes structures, and m_i is the number of nonterminal nodes in the i-th nodes structure. Although N is not explicitly evaluated in an exact expression (because the values of K and m are not determined), its value evidently can be very large.

The above consideration generally prevents the practice of constructing and evaluating all possible structures. For the purpose of having a systematic approach to design a "good" decision tree structure, methods of optimization are considered.

4.3.1 Objective of the Decision Tree Optimization

The objective of the decision tree optimization, as a result of the discussion in Chapter 2, would be to improve either the classification accuracy or the computational efficiency or both. The simultaneous optimization (maximizing) of both the accuracy and the efficiency would be impossible, because according the theorems in Section 2.2, for any given accuracy a bound on efficiency has to be satisfied. That is, a solution which maximizes both the accuracy and the efficiency without constraint simply does not exist. In trying to achieve the goal of maximizing just the accuracy, the decision tree procedure will be useful only if the optimal dimensionality is less than the feature dimensionality (because of the dimensionality problem discussed in Section 2.1). However, in many cases a user is willing to sacrifice some accuracy in order to gain efficiency, even if the dimensionality problem may not occur for maximum feature dimensionality. In these cases, the amount of tradeoff between accuracy and efficiency would be entirely up to that user.

With the above considerations, difference in the performance criteria leads to two different approaches to optimize the decision trees. One tries to maximize the accuracy and another the "overall performance".

4.3.2 The Accuracy Oriented Design Approach

4.3.2.1 A Class of Binary Tree Classifiers

A class of binary trees will be designed for the purpose of maximizing the classification accuracy. In a binary decision tree, each nonterminal node has exactly two immediate descendant nodes. For our special purpose this corresponds to a test of likelihood for a pair of classes using the optimal feature subset for that pair of classes. If the dimensionality problem (described in Section 2.1) does not occur for maximum dimensionality, the optimal feature subsets for all class pairs will be the same, i.e. the complete set. Hence the binary tree procedure is equivalent to the conventional one stage procedure which also performs series of tests to make a final decision. If the dimensionality problem does occur for maximum dimensionality, the optimal feature subsets for different class pairs can be different. In this case, the binary tree procedure is not equivalent to the conventional procedure.

An illustration of the binary tree procedure is shown in Fig. 4.7 for classifying an unknown into four classes $\{\omega_1,\omega_2,\omega_3,\omega_4\}$. In this figure the class of a terminal node is the final decision, and f(i,j) denotes the optimal feature subset used in the decision function for classifying classes ω_i and ω_j.

From this example it is clear that for N-class classification N-1 tests are necessary to reach a terminal

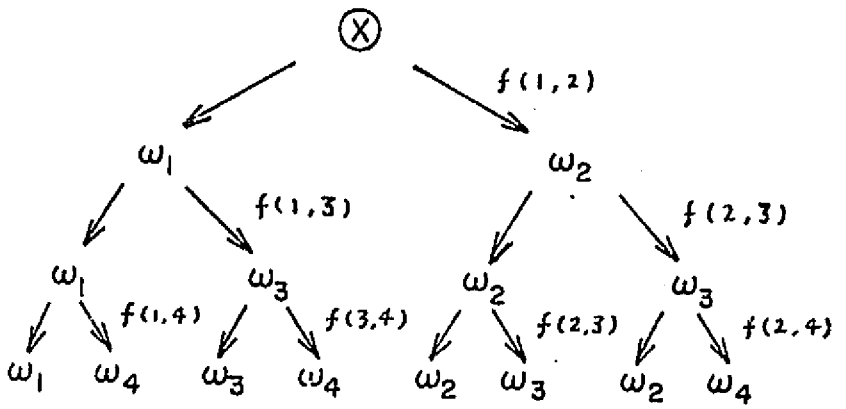


Figure 4.7 A Binary Tree Structure for Four Class Classification.

decision. Therefore, the decision rule of the binary tree procedure for optimal classification can be formally defined as follows: In a optimal binary tree procedure, to reach a terminal decision for N-class classification, a sequence of N-1 tests are performed; in each test a Baysian decision rule is used to classify a pair of classes (i.e. to discriminate one class from another), and the class rejected in the test is excluded from consideration in further tests.

The mathematical formulation of the binary tree procedure is also shown below:

Assuming D is the optimal decision function (with equal a priori probability and 0-1 loss function) for testing class pair ω_i and ω_j, and Ω is the decision of D, we have

$$\Omega = D(\omega_i, \omega_j) \tag{4.1}$$

with

$$\Omega = \begin{cases} \omega_i & \text{if } r_{ij} \geq 1 \\ \omega_j & \text{otherwise} \end{cases} \tag{4.2}$$

where

$$r_{ij} = \frac{P(x|\omega_i)}{P(x|\omega_j)} \tag{4.3}$$

is the likelihood ratio for two classes ω_i and ω_j.
With Ω and D defined above, the binary tree procedure can be put in the recursive form:

$$\Omega_i = D(\omega_i, \Omega_{i-1}) \quad i = 2, \ldots, N \qquad (4.4)$$

with $\Omega_1 = \omega_1$

where N is the number of classes. The recursive formula of Ω_i starts with Ω_2; and Ω_N is the final decision which determines to which class the unknown sample belongs.

A block diagram of the multistage decision procedure as described in Eq. 4.1 to Eq. 4.4 is shown in Fig. 4.8. There is no need to encode and store the entire tree structure with the method described in Chapter 3. When probability densities of all classes are estimated, the necessary information to specify the binary tree decision procedure uniquely would be the optimal feature subsets for all class pairs. Thus, the key step in designing the binary tree decision procedure is to find the optimal feature subsets for all class pairs based on the estimated statistics. Maximizing the Bhattacharyya distance [45,46] can be a reliable method for feature section. Some experimental results will be shown in Chapter 5.

4.3.2.2 Discussion

For the decision procedure described above, the classification accuracy is maximized since the optimal feature subsets are used for discriminating pairs of classes. The efficiency is generally lower than a conventional procedure using the same feature dimensionality because more conditional probabilities have to be calculated for Eq.

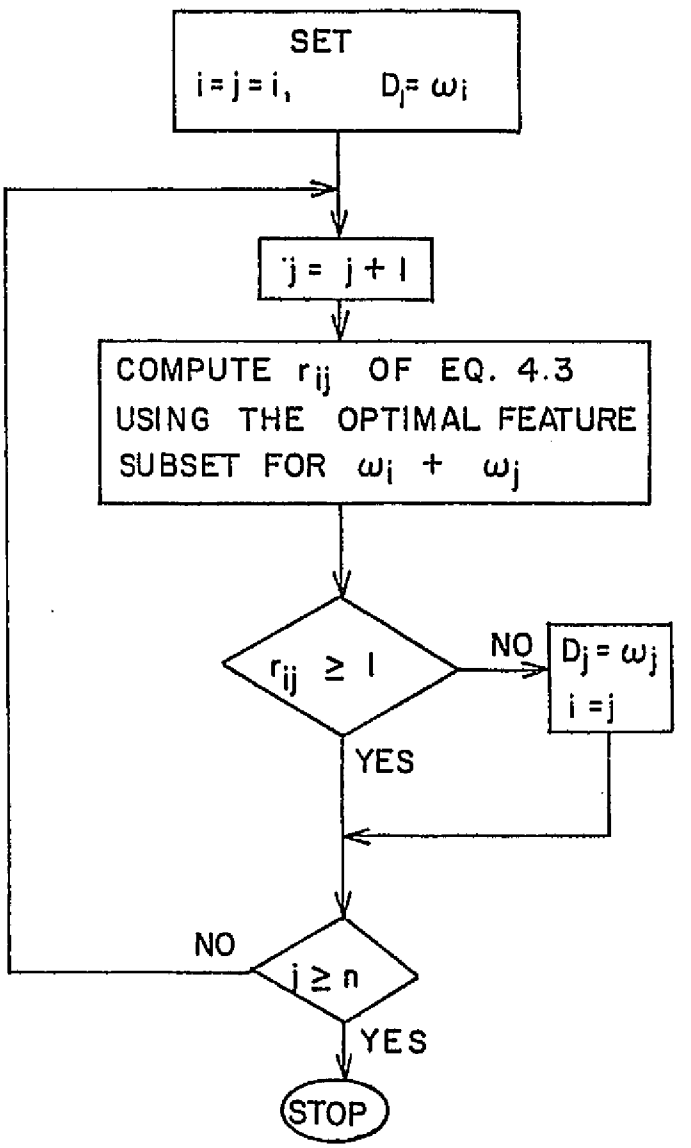


Figure 4.8 Flow Chart of the Binary Decision Tree Procedure.

4.3. If feature subsets for all class pairs are different the number of conditional probabilities calculated is twice the number normally calculated using only one feature subset.

If D is in fact the optimal decision function for classifying two classes, then following the recursive functional form of Eq. 4.4, it is clear that Ω_N is the optimal solution. In other words the procedure of Eq. 4.4 is an optimal procedure for a N-class classification. This is because the multistage decision process defined by Eq. 4.4 is in a recursive form; with D being the optimal decision function, once a true optimal solution ω_k is encountered at the k-th stage the decisions at later stages including the final decision will all be the same, i.e. ω_k. And an optimal solution will be achieved regardless the order of classes in the class sequence. This policy discussed here is described as well by Bellman's "Principle of Optimality" [44] for dynamic programming, which states that - "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

If the true densities are known and if the features are all independent variables the optimal feature subsets for all class pairs are the same, i.e. the complete feature set. As mentioned before, in this case the multistage decision procedure of Eq. 4.4 degenerates to the

conventional optimal procedure, the maximum likelihood procedure. However, if the densities are estimated with different optimal feature subsets for different class pairs the procedure of Eq. 4.4 may not be optimal, for the reason that the law of transitivity* can not be applied to the ordering of likelihood ratios measured in different feature subspaces. With the loss of optimality, contradiction of classification results might occur, if the sequence of classes used in tests is different from the sequence $\{\omega_1,\omega_2,\ldots,\omega_n\}$ used in Eq. 4.4. Different class sequence in the tests corresponds to a different tree structure. As an example the sequence $\{\omega_4,\omega_3,\omega_2,\omega_1\}$ will lead to the structure shown in Fig. 4.9, which is an alternative to the structure shown in Fig. 4.7. The different results for alternative structures is also illustrated by a simple example in Fig. 4.10, where the region ($x < 0$, $y < 0$, $z > 0$) in feature space will be assigned to two different classes due to two different arrangements as shown.

In practical cases, if the probability densities are fairly well represented by the training samples, the population of samples in the ambiguous regions in feature space can be very small. Therefore the difference in classification results due to different arrangements would be negligible. From this standpoint, it is clear that the binary tree approach as described is not optimal but close

*A binary relation R over a set S is said to be transitive if for s, t and u in S, sRt and tRu imply sRu.

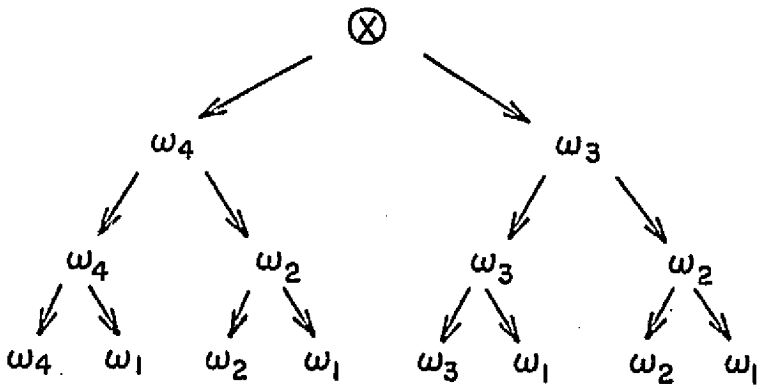


Figure 4.9 Another Binary Decision Tree for Four Class Classification.

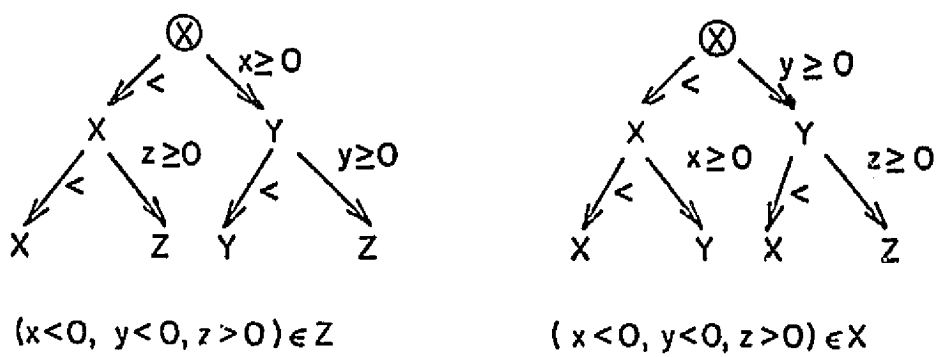


Figure 4.10 A Hypothetical Example Illustrating That Different Binary Trees Lead to Different Classification Results.

enough to an optimal approach to improve the accuracy for multiclass classification in the presence of the dimensionality problem. For an optimal approach, some back up process in a decision tree procedure would be necessary, and thresholds on the likelihood ratio would be used in order to decide whether to reject a class or not. Procedures similar to the Fano's Algorithm in sequential decoding [20] can be designed, the details of which are discussed in Reference [47].

4.3.3 The Search Approach to Optimize the Decision Tree

As mentioned in the end of Section 4.3.1, to maximize the overall performance of a decision tree is one of the goals of decision tree optimization. For this purpose, the designed tree structure must be as general as possible. The essential features for a general and practical tree structure can be stated as follows:

1) Any feature subset can be used in the decision function of a nonterminal node.

2) The number of immediate descendant nodes of a nonterminal node varies from two to the number of classes in that node.

3) The number of classes in a node is always greater than the number of classes in each of its immediate descendant nodes.

4) No two immediate descendant nodes of a nonterminal node contain the same set of classes.

With such generality numerous different structures are possible. There are basically two problems in optimizing the performance of a decision tree. One is the complexity of the tree structure. It has not been possible to describe the tree structure in terms of a set of variables, and then form a space in which each point stands for a unique tree structure. The second problem is that the overall performance of proposed classifier structure can not be predicted exactly accurately. Because of the first problem, most of the existing mathematical programming procedures can not be applied effectively. Hence, the heuristic search method will be used. In this method, the structure is constructed stage by stage, thus reducing the problem of representation. For the second problem, there is no exact solution at present. Attempts have been made to predict the performance as accurately as possible.

Generally speaking, the search method introduced here can be referred to as "guided search with forward pruning", a category in the methods of heuristic search [49,50]. This particular search method is also very close to the branch-and-bound method [51]. The essential concept of the branch-and-bound method is that it partitions solutions into subsolutions (branching) and after each branching, only feasible solutions are retained for further consideration.

4.3.3.1 The Search Procedure

This procedure first selects a set of feature subsets

to be searched. If m, the total number of features is small, all $2^m - 1$ feature combinations can be used. If m is large, feature selection methods can be used to select a set of "likely" feature subsets out of the $2^m - 1$ possibilities. The reduction in feature subsets increases the search efficiency.

These selected feature subsets are then searched in order to construct a stage of the decision tree structure. For each feature subset, with the given classes under consideration, a nonsupervised clustering is performed. With interclass separability as distance,* each class is treated as a single point in the space. As a result of clustering, several groups of classes are found. The candidate substructure (a stage in the tree) for each feature subset is then constructed; i.e. each group of classes represent a newly generated descendant node; the associated decision function has the corresponding feature subset chosen as features, and the statistical parameters for each outcome (descendant node) are the pooled statistics of the "representative classes"** in each group.

*We will assume that a "distance" has some, though not all, the properties of "metric". A metric is a real valued function δ defined on $S \times S$ ($\times$ indicates cartesian product) such that for arbitrary F, G, H in S

(a) $\delta(F,G) \geq 0$

(b) (1) $\delta(F,F) = 0$

(2) If $\delta(F,G) = 0$ then $F = G$

(c) $\delta(F,G) = \delta(G,F)$

(d) $\delta(F,G) + \delta(G,H) \geq \delta(F,H)$

**The "representative classes" are unique to one class group only in contrast to some overlapping classes which belong to more than one class group. Explanation of the clustering procedure with the associated method of extracting those "representative" classes will be given later.

When a candidate substructure is formed, it is evaluated by a function which reflects the cost of classification using that substructure. After all feature subsets are searched, the candidate substructure with the lowest cost will be selected as the substructure for that stage.

The above discussion describes the method of constructing one stage of the decision tree classifier. After this stage is constructed, some of the newly generated nodes may have more than one class. The same procedure is used in expanding those nodes, i.e. constructing the next stage. The search procedure terminates, indicating that the decision tree design is completed, when all terminal nodes contain only one class.

A flow chart and a simple example of the search method is shown in Fig. 4.11 and 4.12 respectively. In the example, six classes ω_i, $i = 1, \ldots, 6$ are to be classified with only four features f_i, $i = 1, \ldots, 4$ available. The search procedure searches through all the $2^4 - 1$ feature subspaces. With a given cost criterion the best structure shown in Fig. 4.12, where the encircled classes are the representative classes. This structure results from the clustering of those six classes based on the separabilities corresponding to feature subset $\{f_1\}$. Notice that the search procedure will be applied to the first and second nodes generated to construct the next stage, since they contain more than one class.

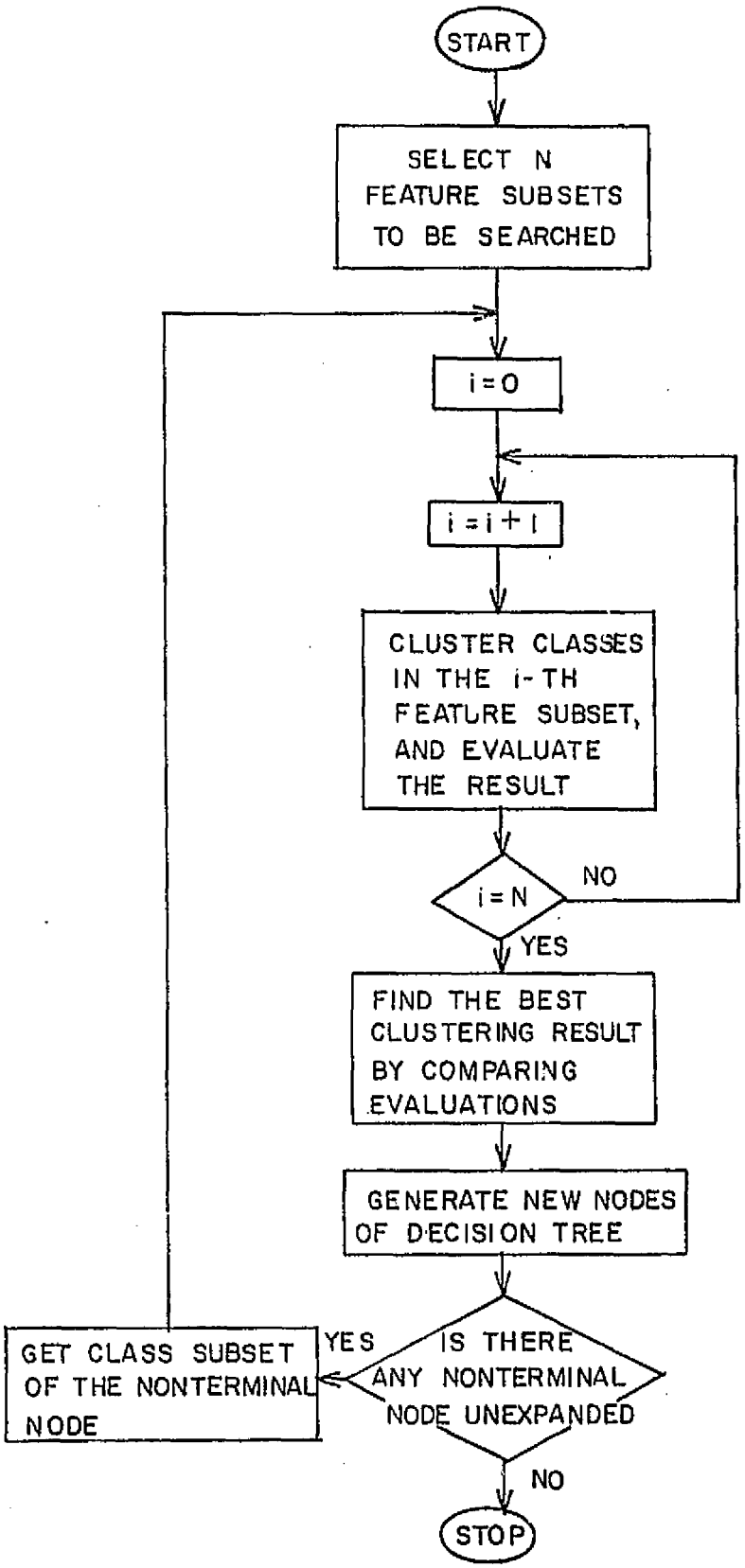


Figure 4.11 A Flow Chart of the Search Procedure.

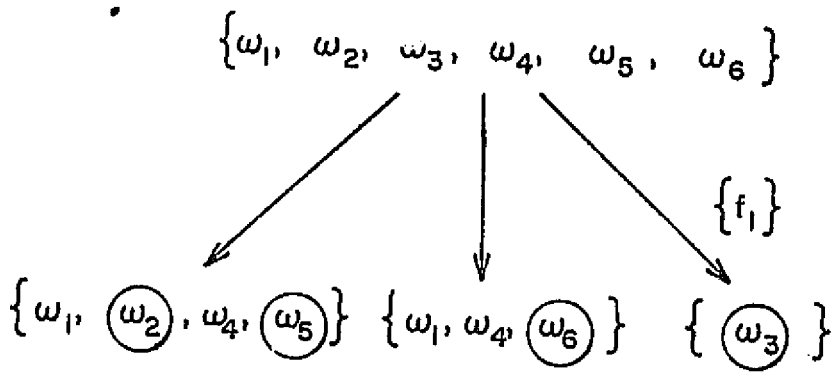


Figure 4.12 A Stage of the Tree Structure.

A set up of the search approach for digital computer application is shown in Fig. 4.13. The same diagram is also valid for the accuracy oriented approach discussed in Section 4.3.2.

In the search procedure described previously, the clustering and evaluation are two major steps. Some introduction to the clustering procedure will be given in next subsection. The form of the evaluation function and the discussion of optimality of the decision tree designed will be given in later subsections.

4.3.3.2 The Clustering Procedure

As mentioned previously, clustering classes into groups is an important step in the search procedure. A brief introduction of the clustering procedure will be given here (while the detailed mathematical verifications will be given in Appendix B).

The first step in the clustering procedure is to form a distance matrix for the points that are to be clustered. The second step is to find several nonoverlapped point subsets. These subsets have the property that only points from the same subset are considered similar, while points from different subsets can never be similar. Whether two points are similar or not is determined by a similarity criterion defined on the distance between these two points. After these distinct subsets are found, the same number of

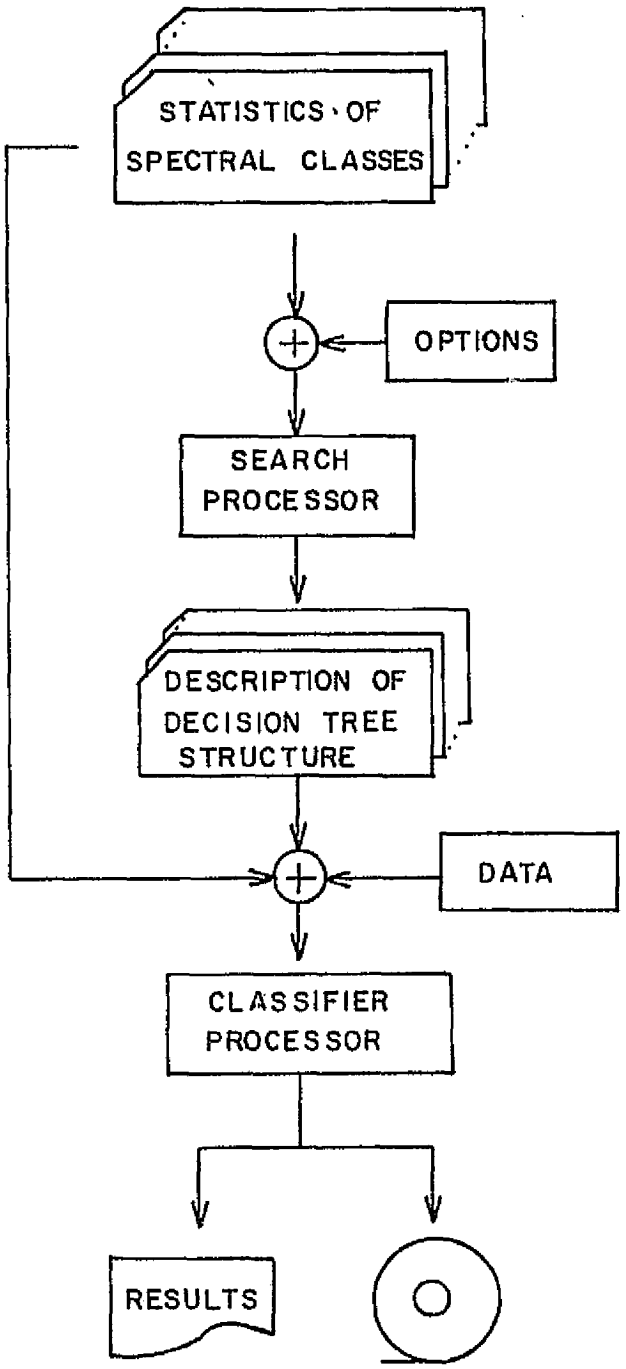


Figure 4.13 Input/Output Set Up of Decision Tree Procedure with Optimization Approach.

clusters are formed each being obtained by grouping all the points similar to the points in a subset previously selected.

The same procedure can be applied to classes, with each class being treated as a single point and the separability between two class distributions as a measure to determine whether these two classes are similar or not. By doing so, we may again form a similarity matrix. Using the clustering procedure, distinct and mutually dissimilar (for any two classes belonging to two different class subsets selected) class subsets can be selected. The classes in these distinct class subsets will be called the representative classes. Groups of classes selected later based on the first selected class subsets will then be clusters which are the proposed immediate descendant nodes of this stage. The significance of the representative classes has been mentioned in the previous section, i.e. the parameters of the decision functions are pooled statistics of those representative classes.

The idea of the clustering procedure is very simple. However, to sort out the distinct and mutually dissimilar point (or class) subsets is not easy, especially if the number of points (or class) is large. A method to simplify this cluster sorting procedure, as developed in this study is explained in Appendix B.

4.3.3.3 Form of Evaluation Function

The evaluation function is essential to the method of guided search as discussed in Section 4.3.3.1. The form of the function reflects the objective of optimization (that is to maximize the overall performance of accuracy and efficiency). To specify the performance criterion, the additive form of accuracy and efficiency will be used. This form is chosen because the additive form of the total cost has been widely accepted by statisticians [48].

As the overall performance of the decision tree is evaluated by the weighted sum of accuracy and efficiency, each stage of the tree will be evaluated by a similar criterion. The evaluation function $E(d_i)$ for each candidate structure following node d_i will be defined as follows:

$$E(d_i) = -T(d_i) - K \cdot \varepsilon(d_i) + \sum_{j=1}^{c_i} E(d_{\ell+j}) \qquad (4.5)$$

where the evaluation of the decision function for node d_i is given by the first two terms. The summation quantity is the predicted evaluation for further stages. The efficiency and accuracy are represented by the negative of the computation time $T(d_i)$ and negative of the error $\varepsilon(d_i)$ respectively; both quantities are measures for node d_i only. K is a weight constant which determines the relative importance of efficiency and accuracy, and its value will be assigned by the user. c_i is the number of immediate descendant nodes of d_i, and $d_{\ell+j}$ are those nodes, with $E(d_{\ell+j})$ as their associated evaluations. To be more specific, we have

$$E(d_i) = \frac{1}{T(m,n)} \left[P_i \cdot T(m_i, c_i) + \sum_{j=1}^{c_i} P_{\ell+j} \cdot T(m, n_{\ell+j})\right]$$

$$- K \cdot \varepsilon(d_i) + K \cdot C \qquad (4.6)$$

where T(a,b) stands for the computation time of a maximum likelihood procedure for an a-feature b-class classification. m,n are the number of features and classes used in the conventional one stage procedure. P_k is the probability that a path of classification will pass through node d_k. m_i and c_i are the number of features and decisions, respectively, of the decision function proposed for node d_i. $n_{\ell+j}$ is the number of classes contained in the descendant node $d_{\ell+j}$. And C is a constant to be explained in the next paragraph. The meanings of some of the notations appearing in Eq. 4.6 are also illustrated in Fig. 4.14.

The term $T(d_i)$ in Eq. 4.5 is expressed by $P_i \cdot T(m_i, c_i)$ in Eq. 4.6. $K \cdot \varepsilon(d_i)$ in Eq. 4.5 is not changed in Eq. 4.6. $E(d_{\ell+j})$ in Eq. 4.5 is expressed by $P_{\ell+j} \cdot T(m, n_{\ell+j})$ which is the computation time of the one stage procedure (with m features and $n_{\ell+j}$ classes) being designed for node $d_{\ell+j}$; and $\varepsilon(d_{\ell+j})$, the expression for error is not included in Eq. 4.6. The reason that this simplified form for $E(d_{\ell+j})$ is used is because structures for further stages have not yet been determined, and efficiency and accuracy are difficult to predict; thus the conservative single stage procedure evaluations are proposed for each of the immediate descendant nodes $d_{\ell+j}$ of node d_i. And the sum of error quantity

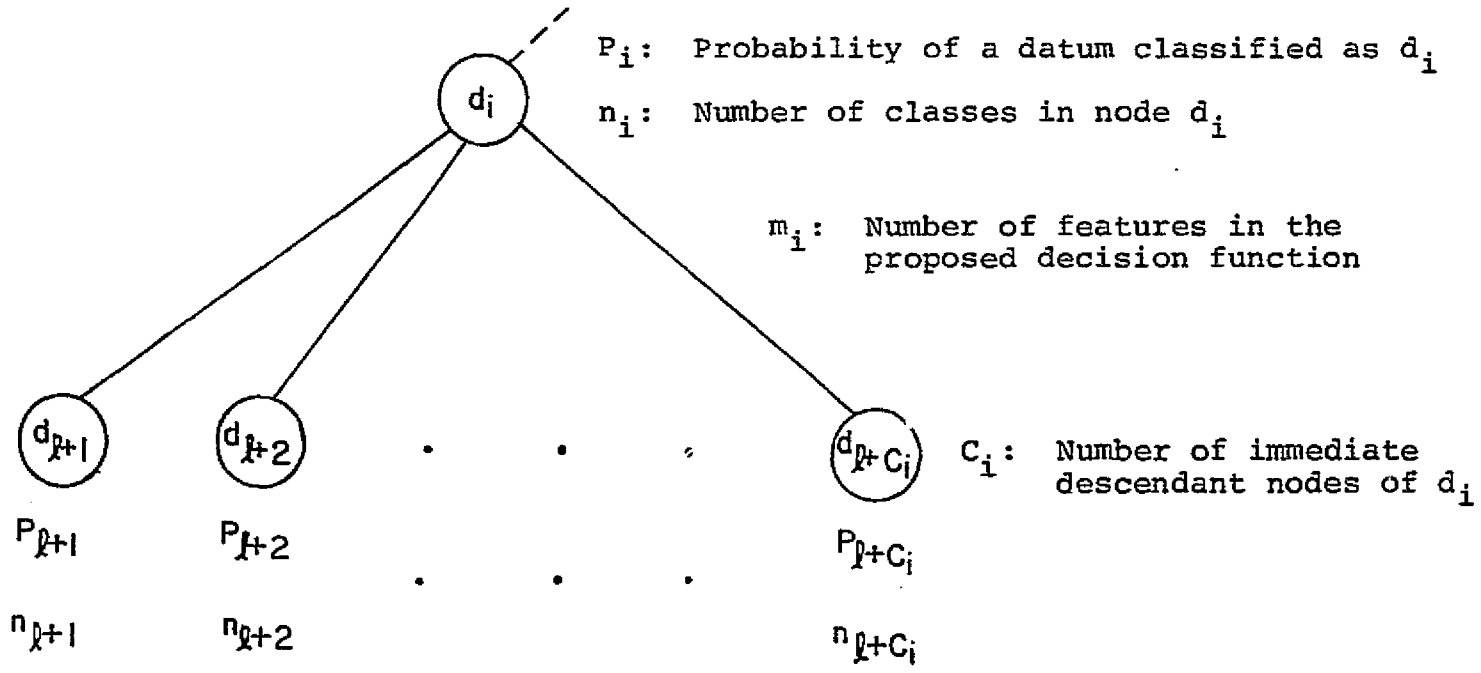


Figure 4.14 A Stage of the Decision Tree Classifier

$\varepsilon(d_{\ell+j})$ is expressed by a bias constant C. The terms associated with efficiency are normalized by the computation time of one stage conventional procedure, i.e. T(m,n), so that they can be compared to the terms associated with accuracy which are expressed in terms of error rate.

To improve the performance of the designed classifier, a constraint on $E(d_i)$ is applied, which is the evaluation of a conventional procedure to be used for node d_i such that the evaluation of a selected substructure can not be less than this constraint. In other words, a conventional procedure will be used for node d_i, if the evaluations of all candidate structures are no greater than this constraint. Another interpretation of the constraint is that a conventional one stage structure is also added as a candidate substructure to be evaluated. The constraint $E_o(d_i)$ for node d_i is given by Eq. 4.7

$$E_o(d_i) = -\frac{1}{T(m,n)} [P_i \cdot T(m,n_i)] - K \times \varepsilon_o(d_i) \qquad (4.7)$$

where n_i is the number of classes in node d_i.

Since $E_o(d_i)$ is a constant term for all $E(d_i)$, it is convenient to substract $F_o(d_i)$ from the expression for $E(d_i)$. The constraint for this modified $E(d_i)$ will be zero for all d_i. The modified form of evaluation is then given by Eq. 4.8.

$$E'(d_i) = \frac{1}{T(m,n)} \{P_i \cdot T(m,n_i) - [P_i \cdot T(m_i,c_i) + \sum_{j=1}^{c_i} P_{\ell+j} \cdot T(m,n_{\ell+j})]\} + K \times [C' - \varepsilon(d_i)] \qquad (4.8)$$

Since $\epsilon_o(d_i)$ is a constant, it is absorbed in the bias constant C'.

In Eq. 4.8 all the quantities of $T(\cdot,\cdot)$ are known quantities, since the computation time of classification for a given number of classes and features can be measured. The remaining quantities can not be calculated precisely. In Eq. 4.8, they are P_i, $P_{\ell+j}$ and $\epsilon(d_i)$. However, with a good separability measure, these probabilities can be estimated reasonably well. The empirical method of estimating probabilities is given in Appendix C.

4.3.3.4 Discussion of the Optimality of the Design

Equation 4.8 is used to optimize a stage of the structure. How this relates to the optimization of the overall performance of the decision tree is explained as follows:

In a designed tree structure, assume there are totally N nonterminal nodes. The summation of the evaluations of these N nonterminal nodes, d_i, $i = 1, \ldots, N$ is given below

$$E = \sum_{i=1}^{N} E'(d_i)$$

$$= \frac{1}{T(m,n)} \sum_{i=1}^{N} [P_i \cdot T(m,n_i) - \sum_{j=1}^{c_i} P_{\ell+j} \cdot T(m,n_{\ell+j})]$$

$$- \frac{1}{T(m,n)} \sum_{i=1}^{N} P_i \cdot T(m_i,c_i) + K \cdot NC' - K \cdot \sum_{i=1}^{N} \epsilon(d_i) \quad (4.9)$$

In Eq. 4.9, terms in the first underlined summation will cancel each other except for the first term $P_1 \cdot T(m,n_1)$ of the root node, which is equivalent to $T(m,n)$. The second summation is the expression of computational time of the decision tree procedure. And the last summation is the total error rate. Let T_o, ε_o and T, ε be the computation time, and error rate of the conventional procedure and the tree procedure respectively, as defined below:

$$T_o = T(m,n)$$

$$T = \sum_{i=1}^{N} P_i \cdot T(m_i,n_i)$$

$$\varepsilon = \sum_{i=1}^{N} \varepsilon(d_i)$$

With these expressions, Eq. 4.9 is rewritten as

$$E = \frac{1}{T_o}[T_o - T] - K \times \varepsilon + K \times NC' \qquad (4.10)$$

Eq. 4.10 can be viewed as the difference in performances of the tree procedure and the conventional procedure. i.e.

$$E = \left(-\frac{T}{T_o} - K \times \varepsilon\right) - \left(-\frac{T_o}{T_o} - K \times \varepsilon_o\right) + C'' \qquad (4.11)$$

where $$C'' = K \times NC' - K \times \varepsilon_o \text{ is a constant.} \qquad (4.12)$$

Through the above derivations, the consistency of the evaluation $E(d_i)$ and the overall evaluation is evident. In other words maximizing $E(d_i)$ individually increases the value E as expected. The value of constant C' in Eq. 4.8 is difficult to determine. Ideally, C' should be set close to the value of ε_o/N such that C" vanishes in Eq. 4.11, but N is unknown before a tree is designed. Indeed, one can simply set C' as zero; this is equivalent to raising the constraint of $E(d_i)$ by a positive amount (because the value for which C' stands is positive).

This solution to the design is suboptimal to the objective of optimization. The reasons are summarized as follows:

1) Not all possible tree structures are evaluated.

2) The evaluation is an approximated quantity.

3) Maximizing each $E(d_i)$ does not imply that the overall evaluation $-\frac{T}{T_o} - K \times \varepsilon$ is maximized.

Although the search is suboptimal, with a carefully formulated evaluation function--Eq. 4.8--, net improvement in classifier performance is achieved. The search procedure itself is very efficient, thus its practical usefulness is enhanced. Some experimental results related to the search method for decision tree optimization will be shown in Chapter 5.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Introduction

Several experimental results related to the dimensionality problem will be presented first. Experiments were performed on both real and simulated data sets. Next presented are results of decision tree classifiers based upon various design approaches. Emphasis has been placed on the optimization approach. The reason is twofold: One is to verify the validity of the optimization procedure since several empirical methods are involved; the other is to gain confidence in the performance of a design which is the result of an "automatic" design approach.

The Bayesian decision rule with assumptions of 0-1 loss function, equal a priori probabilities and multivariate normal distributions is used as the decision rule in all experiments when classification is involved.

Two separability measures, the transformed divergence D_T [53] and the transformed Bhattacharyya distance B_T [54], are also introduced here, for they will be used frequently in later experiments as criteria for feature selection.

$$D_T = 2000 \times (1 - e^{-D/8}) \qquad (5.1a)$$

where D has been defined in Eq. 2.9, the divergence of two normal distributions.

$$B_T = 2000 \times [1 - \mathrm{erf}_c(\sqrt{2B})] \qquad (5.1b)$$

where

$$B = \frac{1}{8}(M_2-M_1)^T\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{-1}(M_2-M_1) + \frac{1}{2}\log\frac{|(\Sigma_1+\Sigma_2)/2|}{|\Sigma_1|^{1/2}|\Sigma_2|^{1/2}} \qquad (5.2)$$

and

$$\mathrm{erf}_c(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \qquad (5.3)$$

5.2 Dimensionality Problem in Multispectral Pattern Recognition

In this section, the dimensionality problem will be experimentally studied. There are two major objectives of conducting these experiments. One is to further demonstrate the existence of this problem in multispectral pattern recognition; and the other is to verify the hypothetical explanation of this problem, which is discussed in Section 2.1.

5.2.1 Experiments on Real Data

The following two experiments are mainly for the purpose of observing the dimensionality problem in multispectral pattern recognition. The first experiment is a repetition

of the one shown in the end of [1], except that the training and test data sets are different. The specific purpose of repeating this experiment is to confirm the previous results which demonstrated that in classifying multispectral remotely sensed data the optimal dimensionality can be rather low.

Experiment 5.1 Five classes of crops, oats, soybeans, corn, red clover and wheat are selected from multispectral scanner (hereafter referred to as MSS) data of the 1966 C-1 Flight Line*. Part of the selected data is used for training and a much larger portion is used for testing (detailed field descriptions are listed in Appendix D.1). The number of features used for classification varies from one to twelve. And the feature subsets were selected based on the averaged pairwise transformed divergence D_T (Eq. 5.1a) in conjunction with the condition that a feature subset with lower dimensionality is always a subset of another with higher dimensionality. With feature subsets selected in this manner, although each one may not be the optimal with respect to each dimensionality (but is close to optimal), however the effect of additional features can clearly be observed as classification dimensionality increases. The classification results in terms of overall error rates (averaging by the total number of test samples) are plotted in Fig. 5.1 and also tabulated in Table 5.1. Notice the error rate of the complete feature

*An experimental flight line over west central Tippecanoe County, Indiana. Also described in Ref. [1].

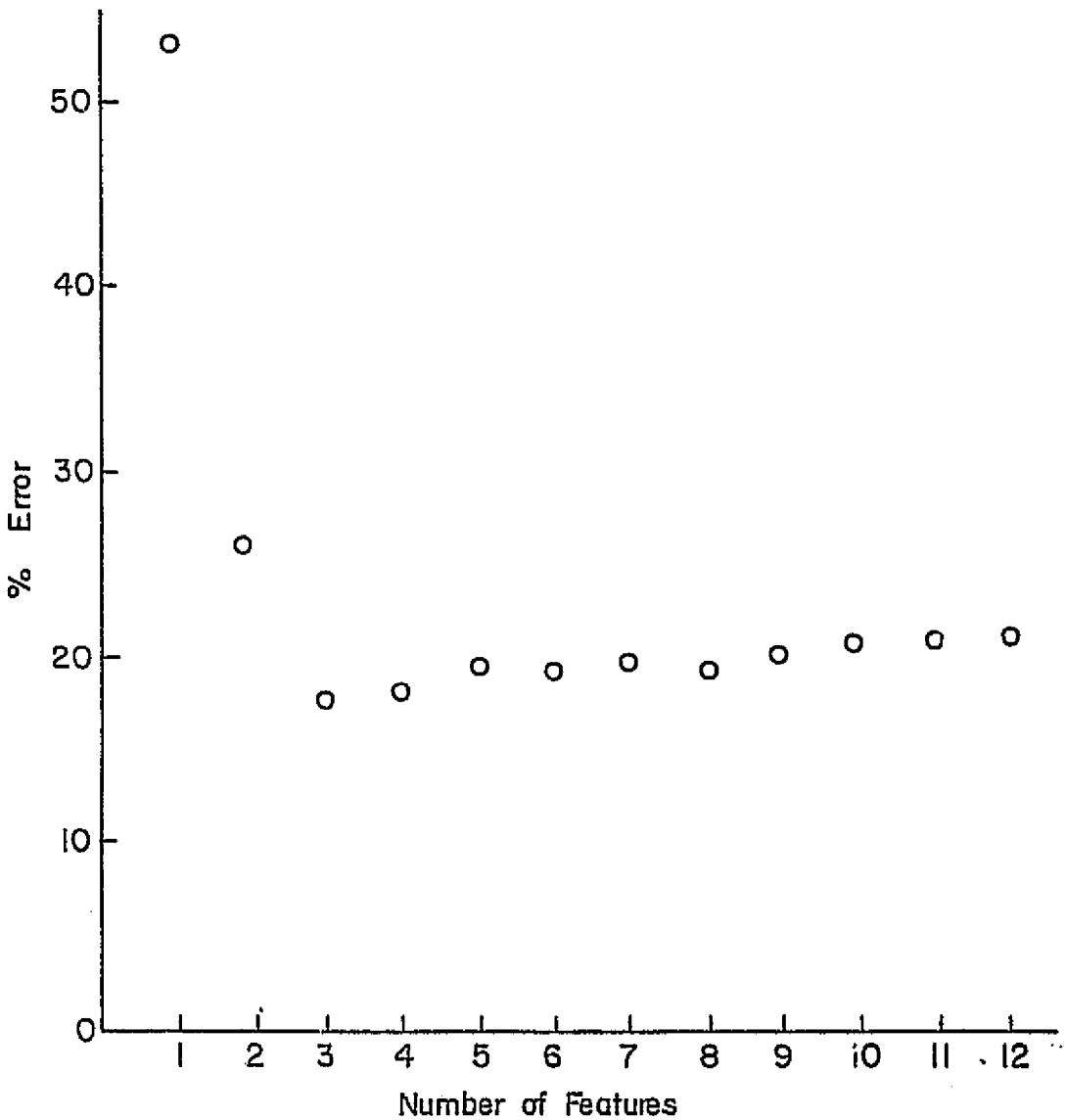


Figure 5.1 Error Rate Versus Dimensionality for the Five Class Test in Experiment 5.1.

Table 5.1 Feature Subsets and Associated Error Rates for the Five Class Test in Experiment 5.1.

FEATURE SUBSET	OVERALL ERROR (%)
1	53.4
1, 10	26.4
1, 10, 12	18.1
1, 9, 10, 12	18.5
1, 6, 9, 10, 12	20.3
1, 6, 9, 10, 11, 12	20.1
1, 6, 8, 9, 10, 11, 12	20.4
1, 5, 6, 8, 9, 10, 11, 12	20.0
1, 5, 6, 7, 8, 9, 10, 11, 12	20.4
1, 4, 5, 6, 7, 8, 9, 10, 11, 12	20.5
1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12	20.9
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	20.9

set is about three percent higher than the best result which is obtained by using three features.

The above experiment is for five-class classification. For a closer look at the problem, an experiment on two-class classification was conducted.

Experiment 5.2 For a two-class classification, crops of corn and soybeans are selected and classified (detailed description of data and results are listed in Appendix D.2). The results are plotted in Fig. 5.2, together with an upper bound ε_o (Ref. [5], p. 70) on error probability, which is calculated by using Eq. 5.4 based on the estimated densities.

$$\varepsilon_o = [P(\omega_1) \cdot P(\omega_2)]^{1/2} \exp(-B) \qquad (5.4)$$

where B is the Bhattacharyya distance defined by Eq. 5.2; and $P(\omega_1)$ are the a priori probabilities estimated by the numbers of test samples for two classes.

From the results of these two experiments, the dimensionality problem in multispectral pattern recognition is clearly observed. It is also noticed that the trend of calculated error bounds based on estimated statistics does not fit the trend of real error rates in this case, i.e. the former goes downward and the latter goes upward. In principle, the error bound ε_o given by Eq. 5.4 will never increase with additional features. Contradiction in the above example occurs because

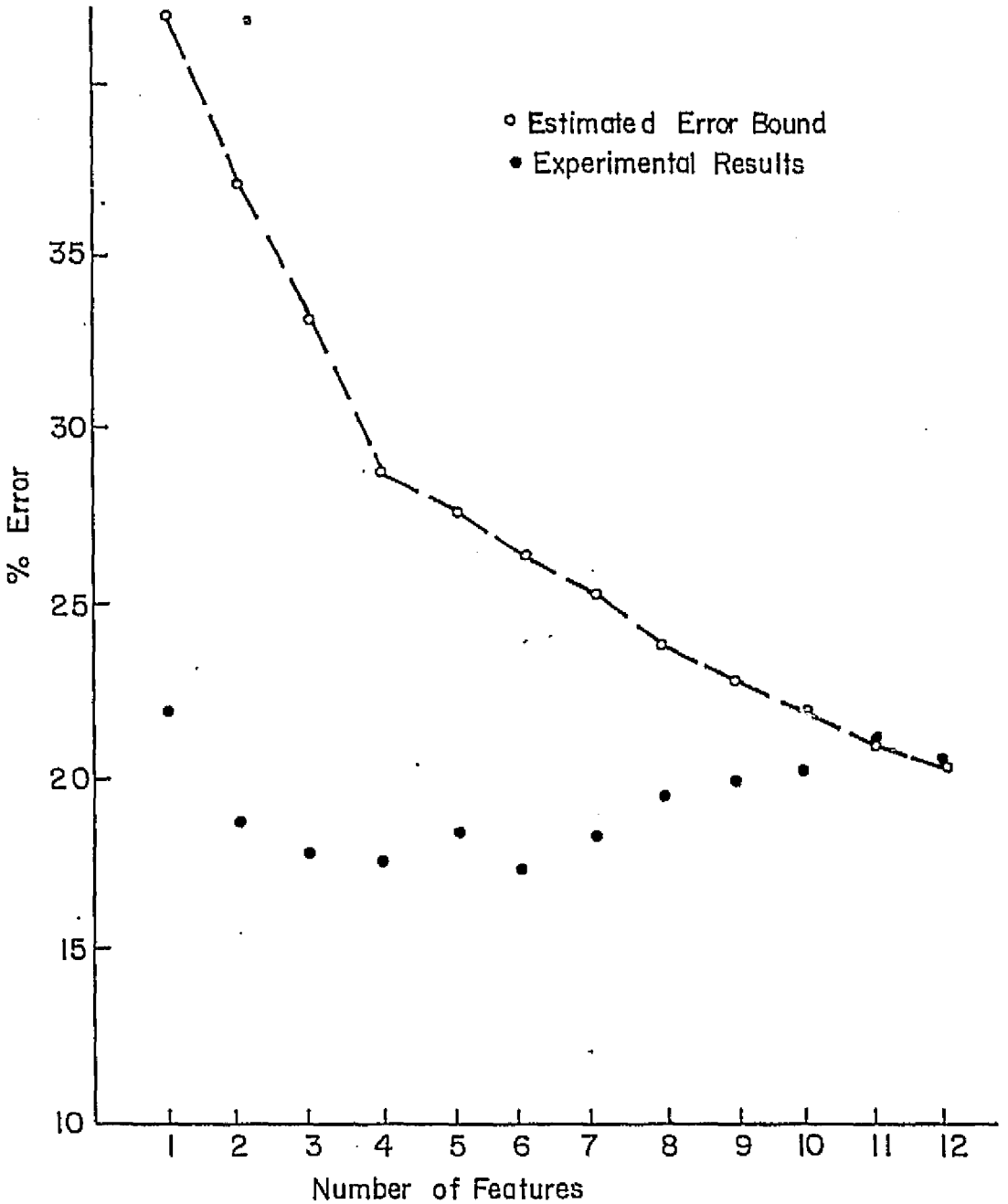


Figure 5.2 Error Rate and Its Upper Bound versus Dimensionality for the Two Class Test in Experiment 5.2.

the densities used to calculate ε_o are not the true densities.

5.2.2 Experiments on Simulated Data

As mentioned in Section 2.1, the dimensionality problem is closely related to the number of training samples in probability density estimation. The following experiments were conducted mainly for the purpose of observing this relationship. Simulated multispectral remotely sensed data have been used for the reason that it is possible to select an arbitrary number of independent samples (real data are more or less spatially correlated). In the simulation, multivariate normally distributed data were generated based on the second order statistics of real remotely sensed data. And the Hasting Formula [55] were used to approximate the inverse of the error function (Eq. 2.12) to transform a random variable from a uniform distribution into a normal distribution.

Experiment 5.3 10,000 samples for each class were randomly generated according to the normal distribution with means and covariances calculated in Experiment 5.2 (Appendix D.2). Totally there were 20,000 samples generated for two classes. Four sets of classifications were performed on all 20,000 samples, using successively 20, 40, 100 and 10,000 training samples per class respectively. In each set the dimensionality varied from one to its upper limit

which is twelve. For each dimensionality the same feature subset was used for all four sets; and the feature subsets were the same as those which were used in Experiment 5.2.

Four sets of results are shown in Fig. 5.3 (Results for 400 training samples were made but are not plotted in Fig. 5.3, because they are very close to the results for 10,000 training samples). The dimensionality problem and its relationship to the number of training samples is apparent. That is, the optimal dimensionality decreases as the number of training samples decreases.

Attempts at theoretically relating the number of training samples to the amount of degradation in accuracy have not been successful, due to the difficulties mentioned in Section 2.1.2. One of the difficulties, the lack of analytical means to estimate errors, can be eased (such that Eq. 2.11 can be used) if both classes are known to have approximately equal covariance matrices. To demonstrate this theoretical result, the following experiment is made on simulated data of two normal distributions with equal covariances.

Experiment 5.4 Two multivariate normal distributions $N(M_1,\Sigma)$, $N(M_2,\Sigma)$ are assumed for two classes of data, where M_1, M_2, are the same as the means of the two classes in Experiment 5.3, and Σ is the covariance of the first class

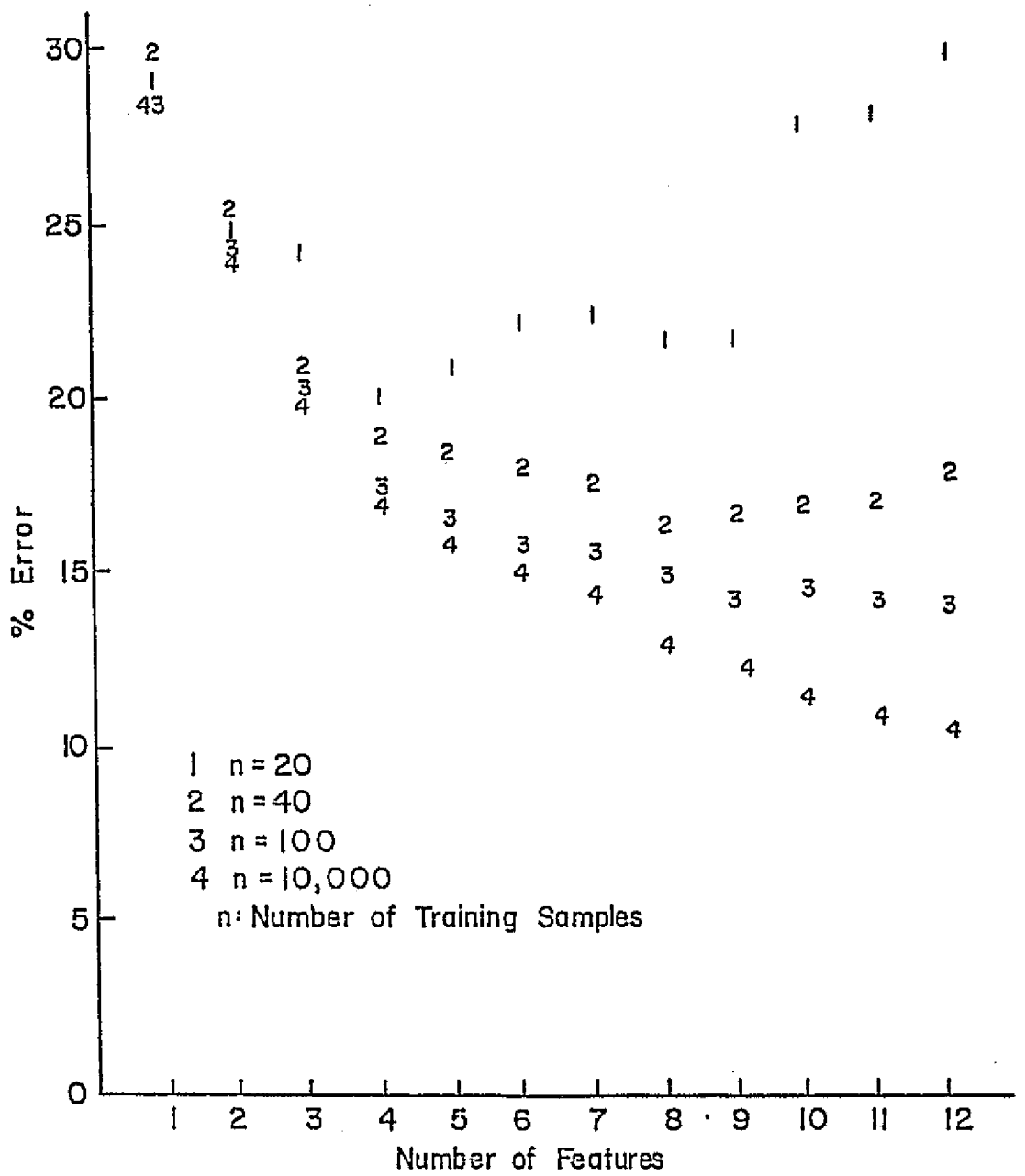


Figure 5.3 Effect of Number of Training Samples on Error Rate in Classifying Two Multivariate Normal Distributions.

in that experiment. 10,000 samples were generated for each class according to the above defined distributions. Again, statistics calculated from 20, 40, 100 and 10,000 samples were used to classify these two classes. The fact of equal covariance in these two distributions was not explicitly used during the experiment, so the procedure of this experiment was the same as Experiment 5.3, except that the feature subsets selected were based on $N(M_1, \Sigma)$ and $N(M_2, \Sigma)$. The classification results are plotted in Fig. 5.4. The theoretical error rates, calculated according to Eq. 2.11 (and Eq. 2.8) with given numbers of samples n, dimensionality m and divergence D (calculated from the true distributions), are also included and are connected by dotted lines in Fig. 5.4.

It is noticed in Fig. 5.4, that the experimental and theoretical results match best, as expected, for the case with $n = 10,000$. Discrepancies between experimental and theoretical results for other cases, in general, occur within three percent. For $n = 20$, the underestimation is probably because small quantities with variances of the order $1/n^2$ are neglected in deriving Eq. 2.8. Despite these discrepancies, the trend of the theoretical results corresponds well with that of the experimental results. One must also recall that Eq. 2.8 is a problem averaged expression for the error in likelihood ratio estimation,

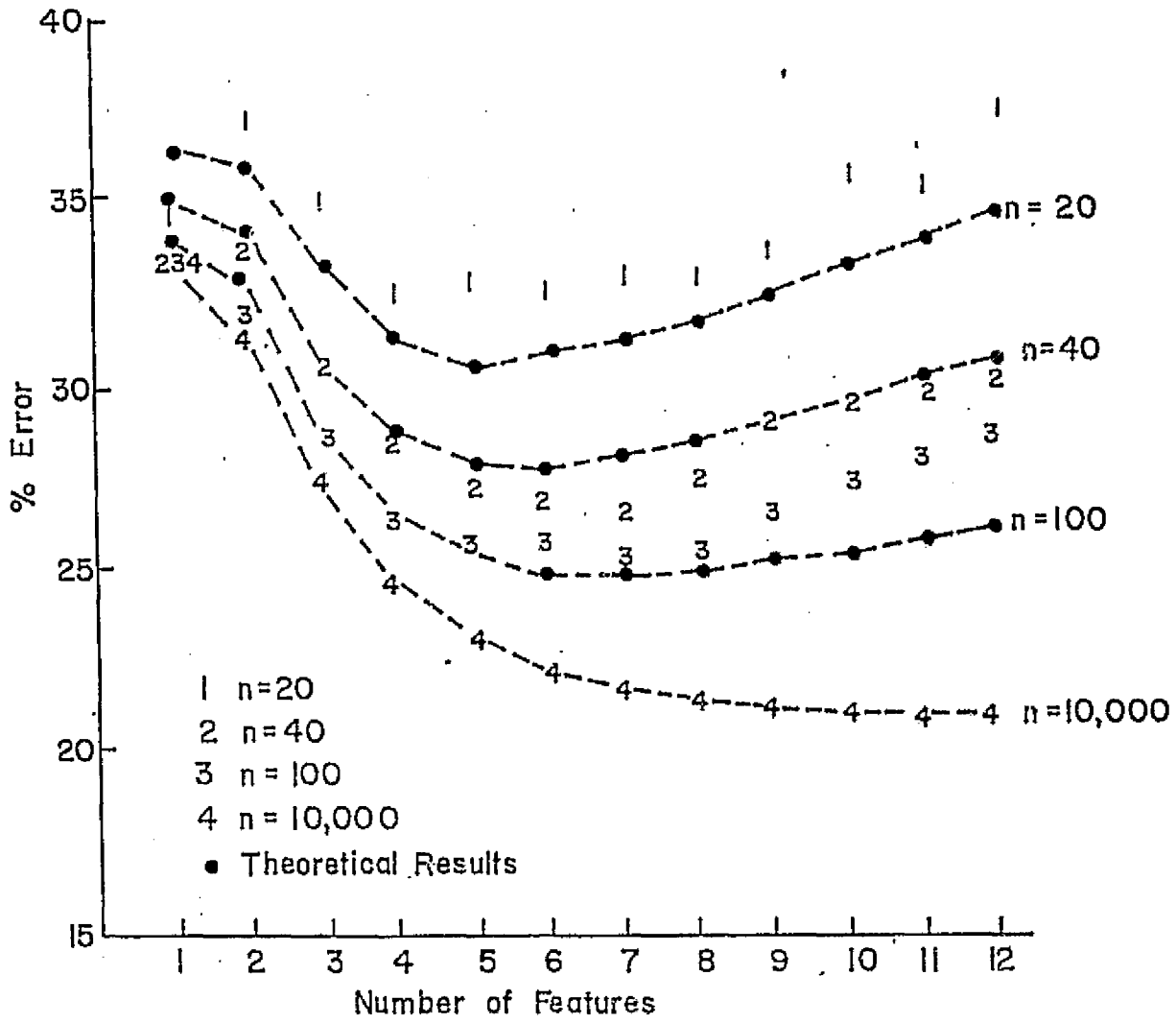


Figure 5.4 Measured and Theoretical Classification Results in Classifying Two Normal Distributions with Equal Covariance.

thus random deviation of the results of a single experiment is to be expected, especially when n is small.

For practically predicting the optimal dimensionality, besides the difficulty of lacking analytic means to predict error accurately, another is that the value of D calculated with a small number of training samples is not accurate enough to be used to estimate the degradation, e.g. Eq. 2.8. An example is shown in Fig. 5.5 where the values of divergence calculated based on statistics used in the classifications in Experiment 5.4 are plotted. For a given dimensionality, there is a general tendency that as the number of samples decreases the divergence increases from its true value. For the equal covariance case such phenomenon can be explained as follows: first, any error in estimating the covariance will lead to the calculated D being greater than its true value (the first term in Eq. 2.9 is never negative for positive definite Σ_1 and Σ_2, but it is zero for the original case of equal covariances); second, there are m (dimensionality) degrees of freedom for the error in estimating the mean vectors, but there is only one possibility (out of m) that D will decrease, which corresponds to the distance between two estimated means decreasing along its true direction (the direction of a vector joining two true means in feature space). For cases with unequal covariances, the second argument still holds and the phenomenon mentioned is expected.

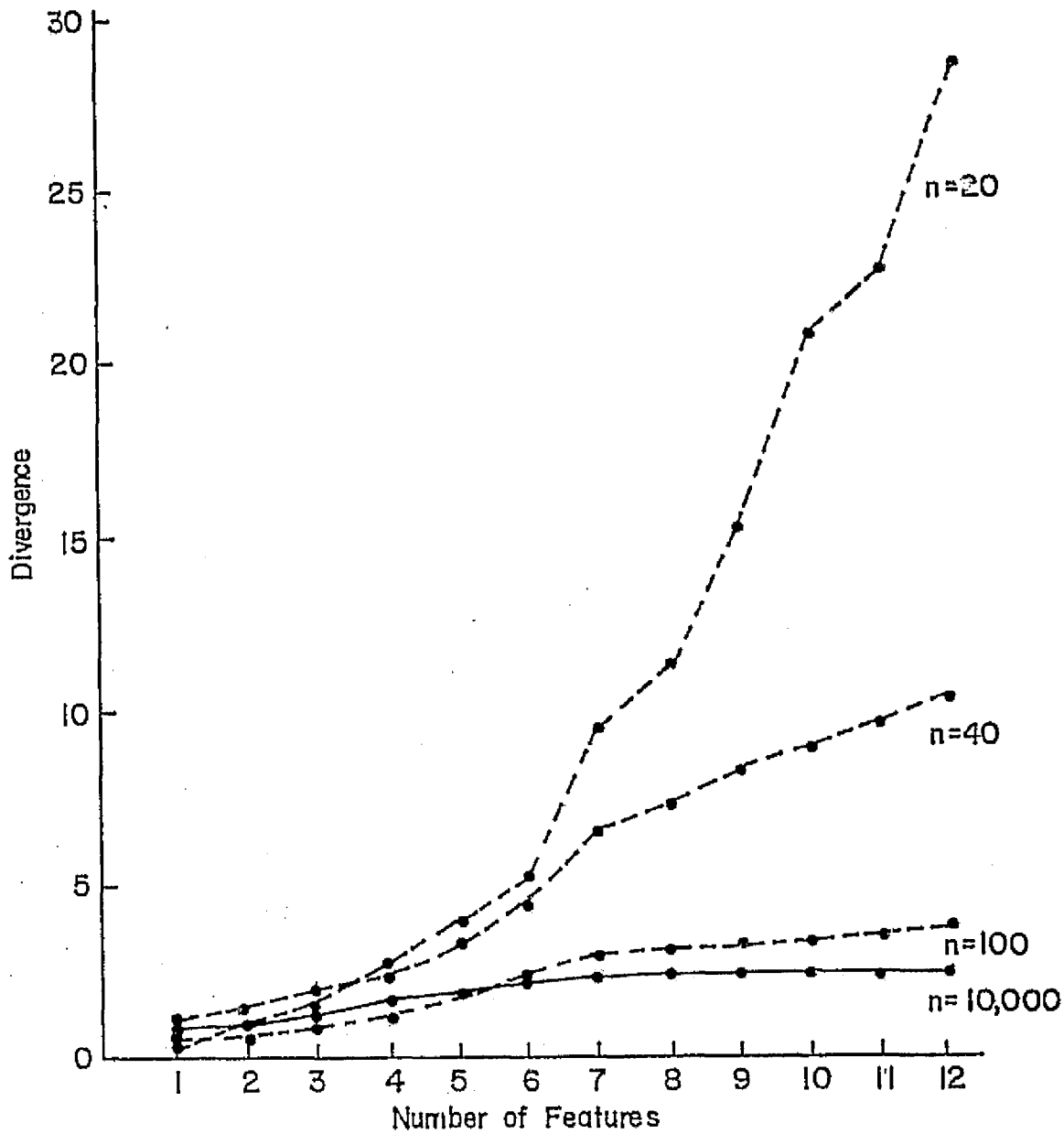


Figure 5.5 Estimated Divergence Based on Sample Statistics for the Two Class Test in Experiment 5.4.

A similar phenomenon is also observed in the Bhattacharyya distance measure. In Fig. 5.6a, estimated upper bounds on error rate for Experiment 5.3 are plotted. The bound is calculated according to Eq. 5.4 by using the Bhattacharyya distance based on the estimated statistical parameters used in Experiment 5.3. It appears as though when n decreases one may expect lower error rate. However, this is only because B tends to be overestimated more for small samples than for larger ones. The true situation is suggested by the real classification results shown in Fig. 5.6b (which is the same as Fig. 5.3 except that the vertical scale is reduced in order to be comparable to Fig. 5.6a), where the performance associated with small n is worse than that with large n. In fact, some of the real results actually exceed the estimated bounds in Fig. 5.6a.

5.2.3 Summary

From the results of the above experiments, it is evident that the optimal dimensionality for classification may be smaller than the dimensionality of the complete feature set, when there are a limited number of training samples for estimating the probability densities. Because the practical method of predicting optimal dimensionality has not been achieved, and because the distance measure may be misleading in case of too few training samples,

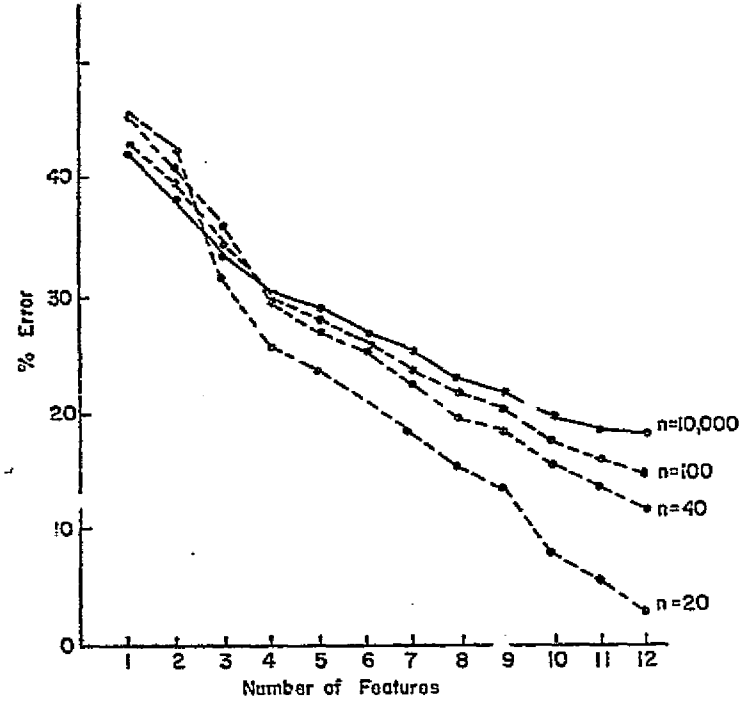


Figure 5.6a Estimated Error Bound Based on Sample Statistics for the Two Class Test in Experiment 5.3.

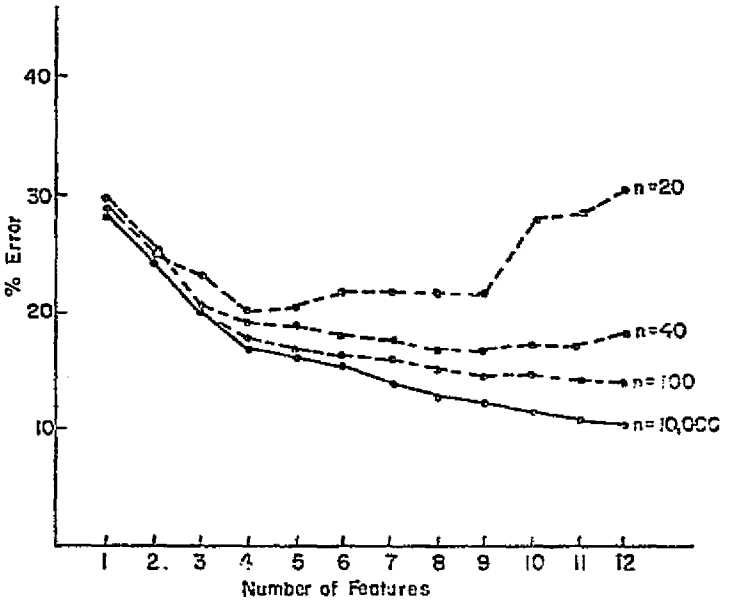


Figure 5.6b Real Classification Results of Experiment 5.3.

to achieve reliable classification results one has to have enough training samples. The results reported in Experiment 5.2 and 5.3 also suggest that any prediction on classifier performance based on a limited number of training samples can be erroneous. One must be aware of this fact and therefore be cautious in selecting features as well as numbers of training and test samples.

5.3 Classification Results of Decision Tree Classifiers

The following are results obtained by utilizing the various approaches to the design of a decision tree classifier discussed in Chapter 4.

5.3.1 Classifier Designed by Utilizing the Histogram Approach

Experiment 5.5 In this experiment, the objective of the classification was to map water bodies in strip mined areas by using aircraft MSS data. Thirteen meaningful spectral classes* were selected, including subcategories of water and other representative coverage types. By examining the coincident spectral plot (in a form similar to the one shown in Fig. 4.3), a decision tree was designed as shown in Fig. 5.7, where the sets labeled by "CH" are sets of spectral channels used in that stage of classification, and the symbol in the parenthesis is the

*Data sets were provided by courtesy of Luis A. Bartolucci.

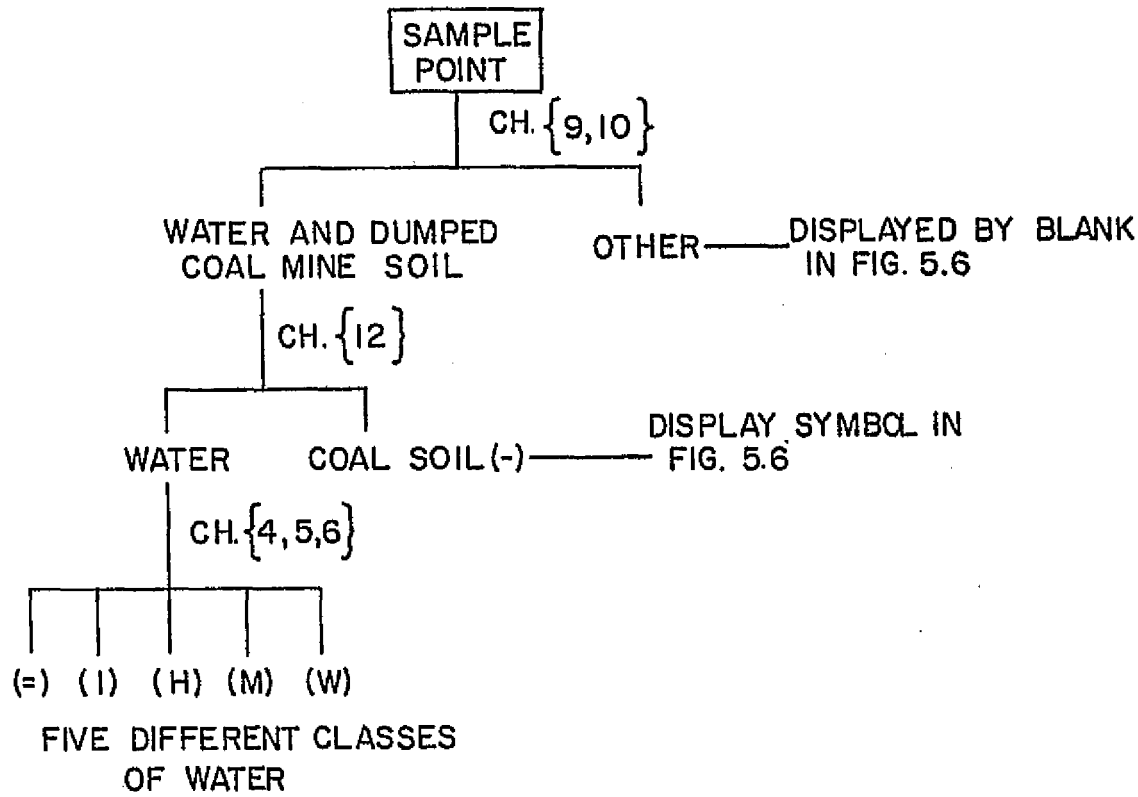


Figure 5.7 A Decision Tree Classifier for Water Mapping.

symbol used to display that particular class in the printed output.

Fig. 5.8a shows the classification results for an area by using the classifier shown in Fig. 5.7. Some of the scattered points classified as water are tree shadows according to aerial photographs; these data points have spectral response similar to water and hence are misclassified.

With the same set of statistics and symbols, Fig. 5.8b shows the classification results by using the conventional one stage procedure with five of the six features* that were used in the decision tree shown in Fig. 5.7. In this set of results another type of error occurs, some areas of water are classified as nonwater. It is difficult to draw conclusions as to which classifier provides more accurate results, however the decision tree procedure is about six times faster in computation than the conventional procedure.

5.3.2 Classifiers Designed by Utilizing the Sequential Clustering Approach

Experiment 5.6 In this experiment, the objective of classification was to detect the change in size of a lake. MSS data gathered from the same area in two different seasons were overlayed The data gathered on one date which was associated with high water level were first clustered into

*One of the three features CH.{4,5,6} of Fig. 5.7, for water subclass classification was not used.

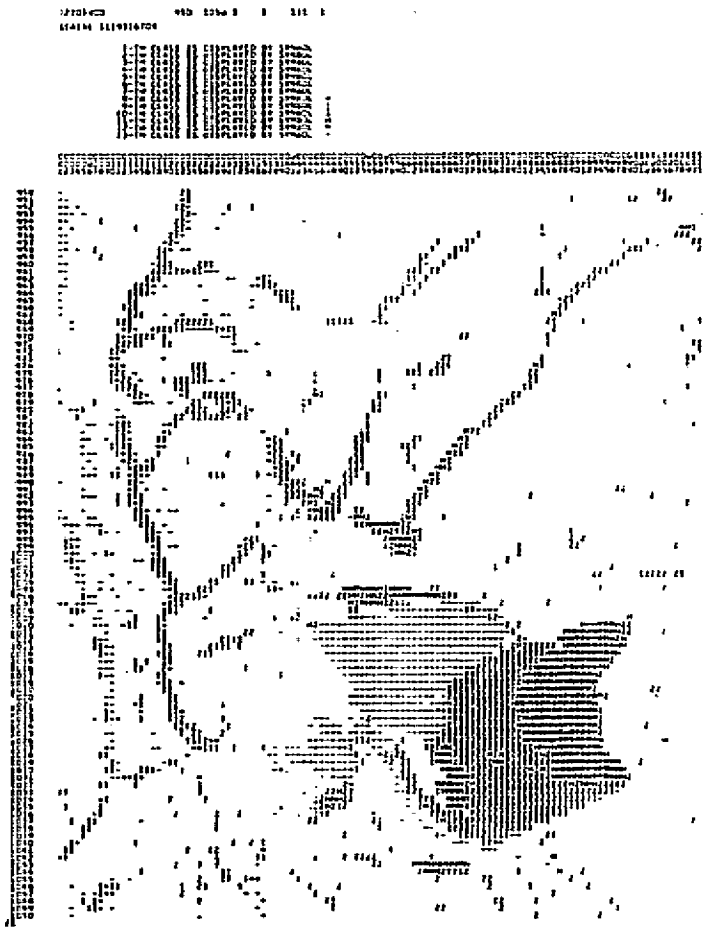


Figure 5.8a Classification Results Using the Classifier Shown in Fig. 5.7.

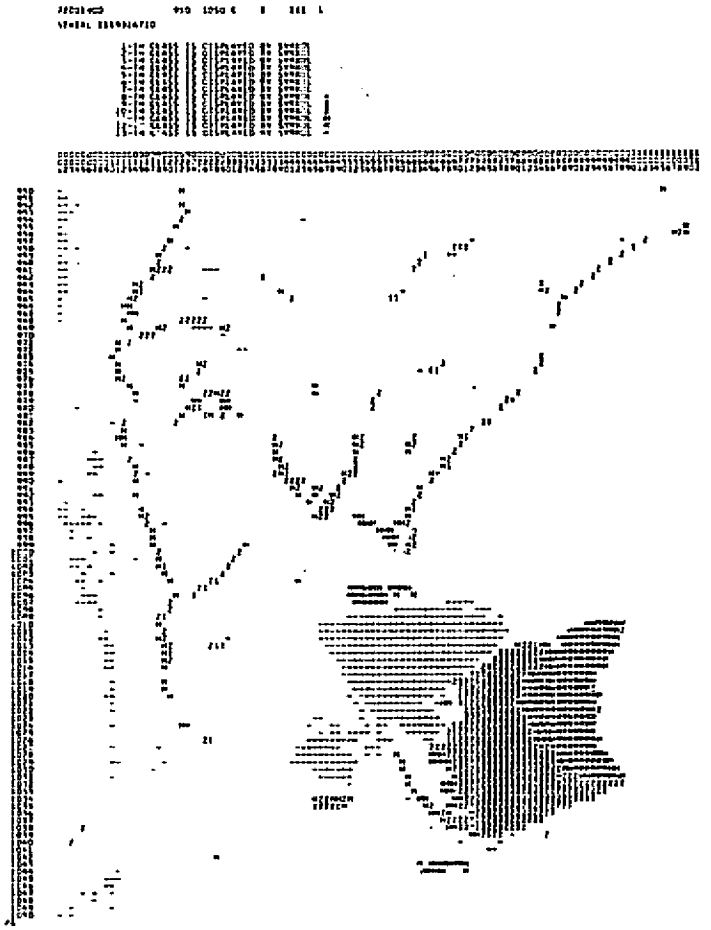


Figure 5.8b Classifier Results Using a Conventional Classifier (5 Features).

three spectral classes. Through the clustering map, one of the cluster classes was observed to correspond accurately to the lake area. Next, a part of the area identified as lake was further clustered into three spectral classes using data gathered on another date; the area selected for clustering was known to have been partially covered by water at that time. After these two steps, a two level decision tree classifier was designed. The results of classification are shown in Fig. 5.9a, where the three classes displayed are water, wet soil and bare soil; the other categories are displayed as blank. The same results are displayed in Fig. 5.9b, where the changed area of water (bare soil) is displayed by dots, the unchanged (includes water and wet land) is displayed by character "W", and the other unrelated areas are displayed as blank. This experiment shows one of the applications of the decision tree classification approach.

As mentioned in Chapter 4, instead of clustering, a supervised learning scheme can also be used to obtain the statistics of spectral classes and so to construct the decision tree in a sequential manner. Results similar to those shown in Fig. 5.9 can be obtained by using this supervised design approach.

5.3.3 Classifiers Designed by Utilizing the Optimization Approach

As a result of the discussion in Section 4.3.1, with a

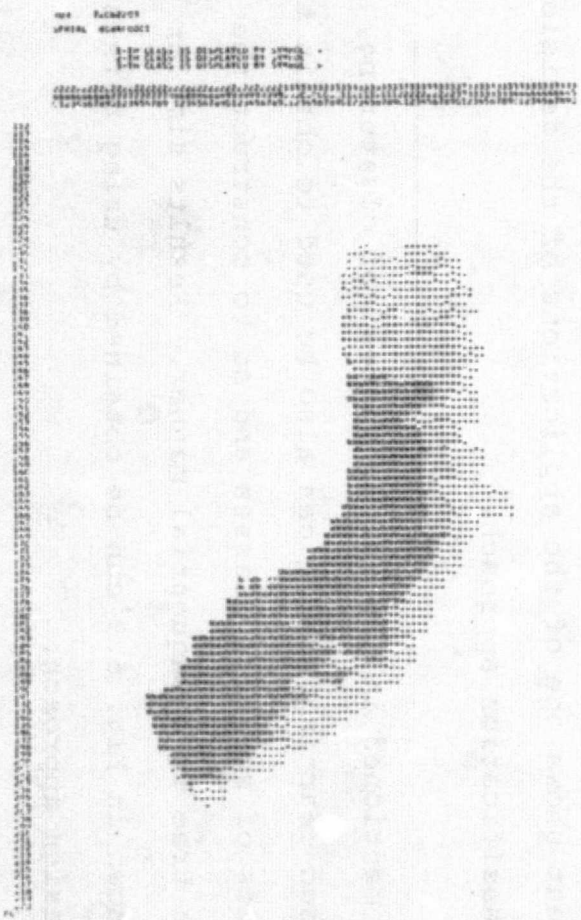


Figure 5.9a Three Spectral Classes of a Lake in Dry Season. .: Bare soil, =: Wet land, W: Water.

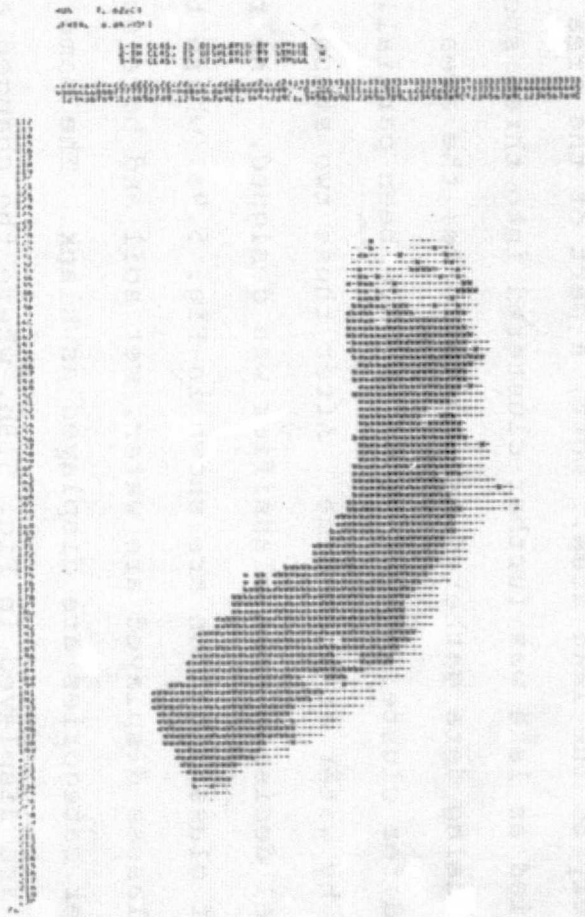


Figure 5.9b Change of Water Covered Area of the Lake in Dry Season. .: Nonwater, W: Water.

given set of classes to be classified, there are two design approaches to optimize the performance of a decision tree classifier. The usefulness of these two approaches was experimentally studied and the results are reported in the following subsections.

5.3.3.1 Binary Decision Trees to Improve the Accuracy

The decision making procedure and the structure of the binary decision tree have been discussed in Section 4.3. The key step to the classifier design then is to find the optimal feature subset for each pair of classes. In the following two experiments, a "without replacement search procedure" [56,57] has been used to select feature subsets for class pairs. The procedure first selects the best single feature from the total set of M features in accordance with a given criterion. Then the remaining (M-1) features are scanned for the next best single feature which results in the best pair when combined with the previously chosen best single feature, and so on. The performance criterion used is to maximize the separability of two probability densities. The reason for using this "without replacement search" approach for feature selection is to test the effectiveness of this suboptimal approach which uses considerably less amount of computation time than the exhaustive search method.

Experiment 5.7 The data set of Experiment 5.1 was used in this experiment. Classification results (in terms of % error) associated with the binary decision tree classifiers designed with different feature selection methods are listed in the last three columns of Table 5.2. Classifiers with three, four and five features to classify a pair of classes were constructed. This number is listed as the "Dim." (an abbreviation of Dimensionality) in the table. The first two columns under the "Binary Decision tree procedure" are results associated with the "without replacement search" method for feature selection, and the effectiveness of both the divergence D and the Bhattacharyya distance B as separability criteria have been tested. The last column lists the results associated with the exhaustive search method for feature selection*, with the Bhattacharyya distance as a separability criterion.

Also listed in Table 5.2 are results obtained by using the conventional maximum likelihood decision rule. For dimensionality three, four and five, results associated with features selected according to maximum average transformed divergence D_T (Eq. 5.1a) and maximum average transformed Bhattacharyya distance B_T (Eq. 5.2a) are listed in the first and second columns under the item "Maximum Likelihood Procedure" respectively. Results listed in the third column

*For a given dimensionality and a pair of classes, the method searches through all possible feature subsets, and finds the one with the highest separability measure.

Table 5.2 Results (% Error) of Five Class Classification by Using Conventional Maximum Likelihood Procedures and Binary Decision Tree Procedures.

DIM.	MAXIMUM LIKELIHOOD PROCEDURE			BINARY DECISION TREE PROCEDURE		
	MAXIMUM AVERAGE D_T	MAXIMUM AVERAGE B_T	BEST RESULTS	SEARCH WITHOUT REPLACEMENT		EXHAUSTIVE SEARCH
				D	B	B
3	22.8	18.1	18.1	21.4	21.1	17.7
4	20.2	18.5	18.5	20.0	17.8	18.3
5	20.3	20.3	18.7	19.9	18.2	20.6
6	19.7					
7	20.4					
8	20.0					
9	20.4					
10	20.5					
11	20.9					
12	20.9					

of the ML Procedure are the best results ever found by using the conventional procedure, which were obtained by testing several other feature subsets associated with close to maximum transformed distance values.

The best results of dimensionality 3, 4 and 5 using the binary decision tree method are 17.7, 17.8 and 18.2; they are plotted in Fig. 5.10 as three circles. The dots in Fig. 5.10 are results using conventional classifiers. The three dots jointed by solid curves are results in the third column (the column of dimensionality is not counted) of Table 5.2, and the others (except the result with Dim.=2) are from the lower portion of the first column.

The results plotted in Fig. 5.10 clearly indicate that for this case the optimal feature dimensionality for the conventional classification procedure is three. A binary tree classifier with this dimensionality for each test does achieve the highest accuracy.

Experiment 5.8 A commonly used data set [58, pp. 6-7] described in Appendix D.3, which is also selected from C-1 Flight Line, is used in this experiment. There are nine spectral classes, two of which are subclasses of wheat. The misclassifications between these two classes are not counted as errors. The procedure of the experiment is simpler than that of Experiment 5.7, i.e. classifications associated with the third and the sixth columns of Table

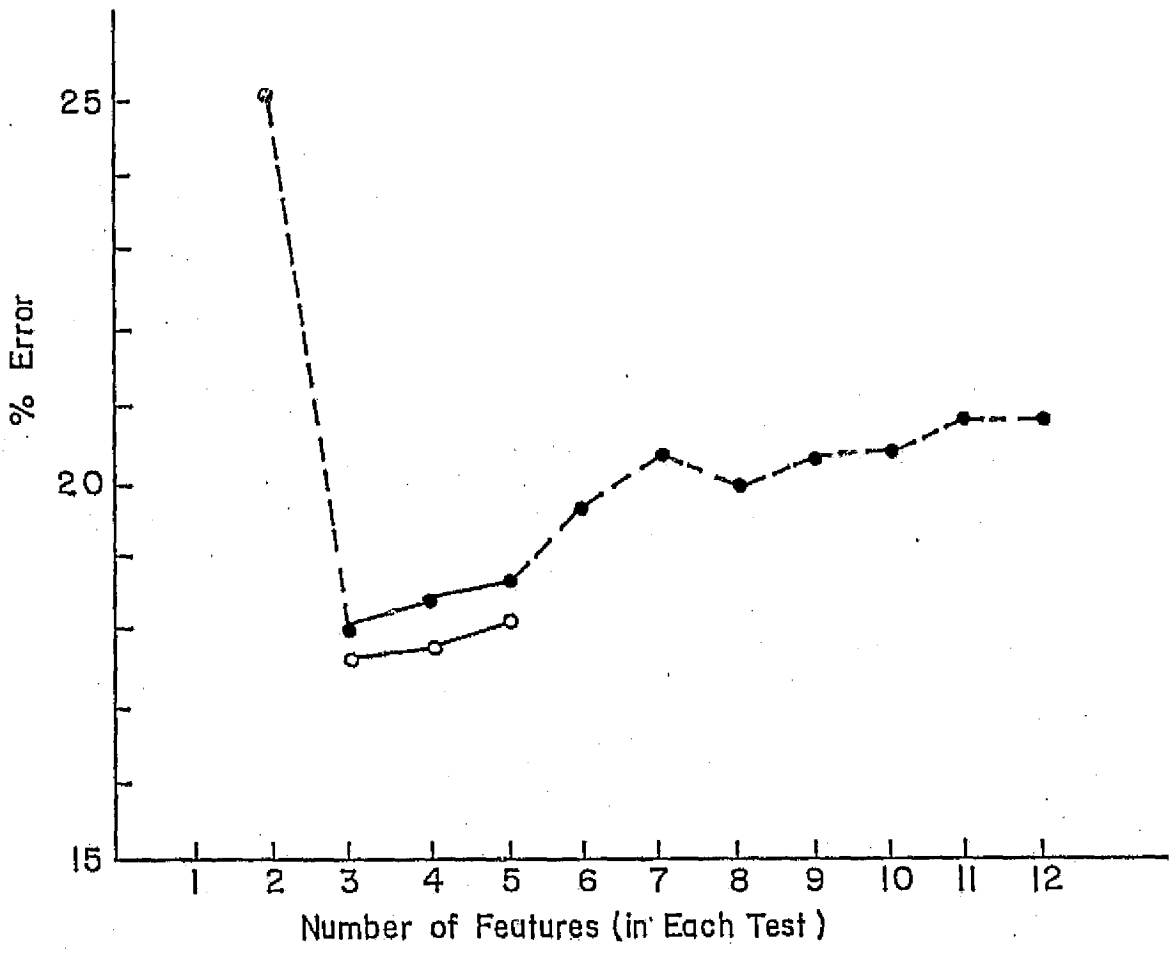


Figure 5.10 Classification Results of Conventional ML Procedures and Binary Decision Tree Procedures for the Five Class Test in Experiment 5.7.

5.2 are not performed. The classification results of this experiment are summarized in Table 5.3, and the values of column 1 and column 4 are plotted in Fig. 5.11.

The net improvement in classification accuracy by utilizing some binary decision tree classifiers is demonstrated in both experiments. Especially in Experiment 5.8, the binary tree classifier achieves the accuracy which can not be achieved by any conventional means. As far as the method of feature selection is concerned, these results suggest that Bhattacharyya distance is better than divergence as a separability criterion for a pair of classes, an inference which can also be drawn from the report by Whitsitt and Landgrebe [54]. For many classes, the performances of average B_T and D_T are comparable, probably because the variance of error rate, which is larger for a given D_T than a B_T in the corresponding range, for an average D_T value is reduced by the averaging process.

5.3.3.2 Classifiers Designed Through the Search Approach

The search approach as described in Section 4.3.3 is for the purpose of designing decision tree classifiers with better overall performances as compared to the conventional classifier. The following experiments are designed to verify whether this objective can be achieved. Experiments on aircraft MSS data will be reported first. Simulated aircraft MSS data are then used to test the validity of the search procedure. Experiments on satellite MSS data are also reported.

Table 5.3 Results (% Error) of Nine Class Classification by Using Conventional Maximum Likelihood Procedures and Binary Decision Tree Procedures.

DIM.	MAXIMUM LIKELIHOOD PROCEDURE		BINARY DECISION PROCEDURE	
	MAXIMUM AVERAGE	MAXIMUM AVERAGE	SEARCH WITHOUT REPLACEMENT	
	D_T	B_T	D	B
3	18.0	22.8	8.2	6.7
4	8.0	8.1	7.2	7.0
5	7.6	7.5	6.7	6.7
6	7.7			
7	7.2			
8	7.2			
9	7.0			
10	7.2			
11	7.2			
12	7.1			

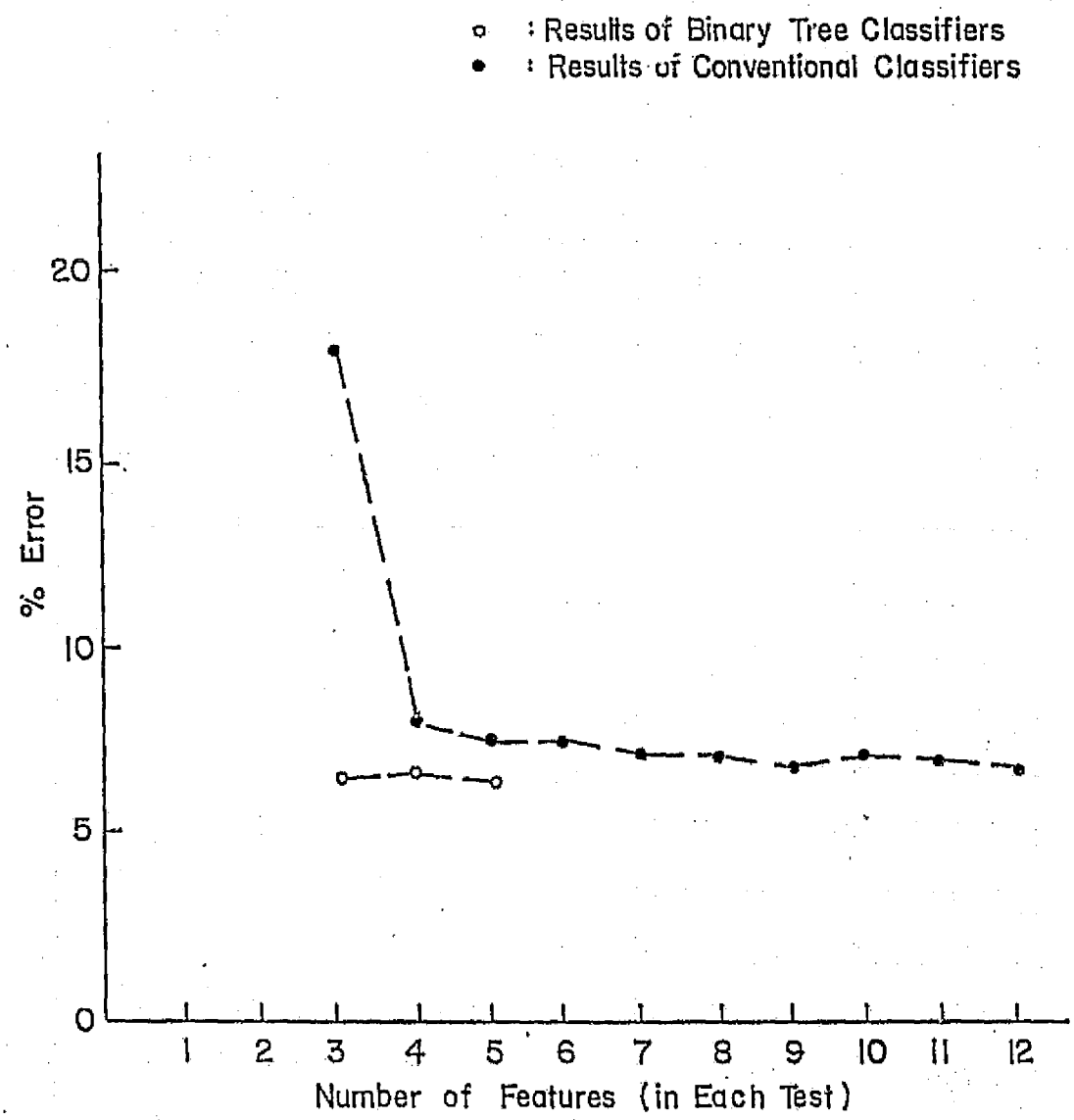


Figure 5.11 Classification Results of Conventional ML Procedures and Binary Decision Tree Procedures for the Nine Class Test in Experiment 5.8.

In each experiment different classifier structures are constructed by varying the design options. These options are listed as follows:

1) The maximum number of features m used in each stage of decision.
2) Distance criterion used in the clustering procedure described in Section 4.3.3.2. Two distance measures have been used: the transformed divergence D_T and the transformed Bhattacharyya distance B_T.
3) Threshold value T_h to determine class similarity (see Appendix B).
4) Tradeoff constant K of Eq. 4.5, which determines the relative importance of accuracy to efficiency.

Experiment 5.9 The data sets of Experiment 5.8 were used in this experiment. Feature subsets for the search were selected using two approaches: One approach was to find a good feature subset first, then form all possible combinations of features from this subset. In this experiment, the criterion of maximizing average D_T used to select a good feature subset. The dimensionality was chosen as four, and features {1,6,10,11} were selected. This resulted in a total of $2^4 - 1 = 15$ feature subsets formed for search. The other approach uses the "without replacement search" method, seeking good feature subsets with dimensionality from one to four for all class pairs. With a total of twelve features, approximately six times as many feature subsets (78 and 79

in Table 5.2) were formed by using the second approach as compared to the first one.

Several different threshold values were used. $T = 1900$ was the starting value for the threshold applied to B_T or D_T. This value was chosen from the past experience that good classifiers can be constructed with thresholds equal to or higher than 1900. Therefore this starting value was used throughout these experiments.

The classification results of the classifiers designed are tabulated in Table 5.4, where columns labeled by "E(%)" are classification results in terms of overall error rate; T/T_o indicates the ratio of the classification time (of central processing unit) associated with the decision tree classifier to that of the conventional classifier with $m = 4$. And m, B_T (or D_T), T_h and K are the four options described previously. The second column labeled "Feature Subsets" are the number of feature subsets searched in designing a decision tree classifier, and the numbers in the fifth column labeled "ID" are to distinguish different classifier structures; classifiers having the same "ID" have the same node structure.

An example of how the tradeoff constant K effects the classifier structure is shown in Fig. 5.12 (only the node structures are shown). With K the only variable option, it is observed that as K increases the structure approaches the one stage conventional classifier. This is expected because using a larger value of K accuracy is emphasized more than efficiency; if the dimensionality

Table 5.4 Decision Tree Design Parameters and Associated Classification Results of Experiment 5.9.

DESCRIPTION					CLASSIFICATION RESULTS			
DISTANCE CRITERION	FEATURE SUBSETS	PARAMETERS		ID	m = 4		m = 3	
		T_h	K		T/To	E(%)	T/To	E(%)
B_T	15+	1900	10.0 20.0	1	0.72	10.3	0.53	17.0
			40.0	2	0.85	7.8	0.62	10.0
			100.0	3*	1.0	8.0	0.69	18.0
		1950	20.0 40.0	4	0.86	7.8	0.62	10.0
		2000	-	3	1.0	8.0	0.69	18.0
	78++	1900	10.0	5	0.60	13.7	0.47	16.8
			20.0	6	0.79	7.8	0.58	16.7
			40.0	7	0.81	8.9	0.58	17.5
			100.0	8	1.0	8.1	0.69	18.0
		1950	20.0 40.0	9	0.72	10.4	0.53	11.2
		2000	-**	8	1.0	8.1	0.69	18.0
D_T	15+	1900	10.0 20.0 40.0	1	0.72	10.3	0.54	19.6
			100.0	3	1.0	8.0	0.69	18.0
		1950	20.0 40.0	1	0.72	10.3	0.54	19.6
		2000	-	10	0.93	8.1	0.67	18.2
	79++	1900	10.0 20.0 40.0	11	0.77	11.0	0.53	20.0
			100.0	8	1.0	8.1	0.69	18.0
		1950	20.0 40.0	11	0.77	11.0	0.53	20.0
		2000	-	12	0.93	8.2	0.67	18.2

* Classifiers with T/To = 1 are same as the conventional ML classifier.

** When T_h = 2000, any positive K will result in the same classifier structure (because $c(d_i) = 0$, see Appendix C).

\+ Feature subsets are combinations of four features {1, 6, 10, 11}

++ Feature subsets are selected from all twelve available features with the "without replacement search" method.

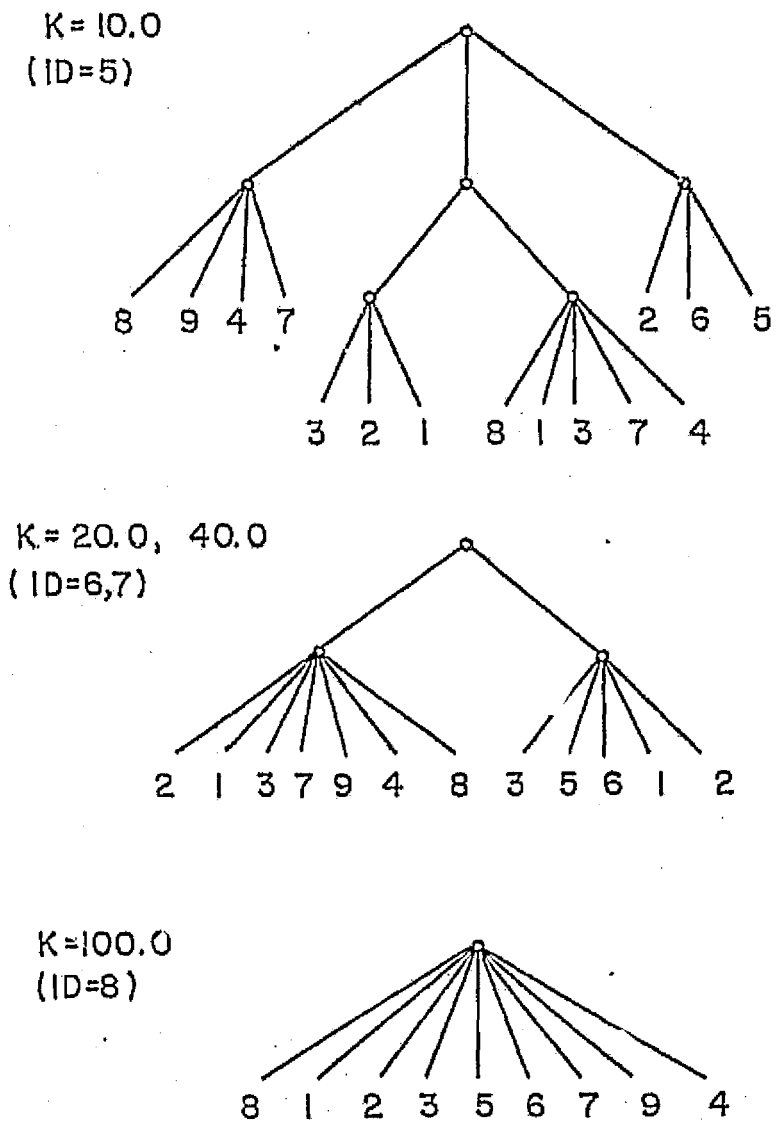


Figure 5.12 Change of Drcision Tree Structure with Respect to the Change of Tradeoff Constant K.

problem does not arise, the conventional procedure with the complete feature set is optimal in accuracy. The performance of the decision tree classifiers are plotted as dots and circles in Fig. 5.13 (Triangles are results of next experiment). The performance of conventional classifiers with $m = 3,4$ are also plotted for comparison purpose: they are the two squares as indicated.

Polynomial curve fitting has not been used for the results plotted in Fig. 5.13 (nor for later experiments), because experiments at this stage are mainly for the purpose of observing which set of parameter values give desirable results; thus it is not very meaningful to discuss the results in terms of "mean" performance of error rate versus the efficiency. It is observed from Table 5.4, that B_T as a separability measure is more effective than D_T; and with B_T as the distance, $T_h = 1950$ can be better than $T_h = 1900$. Another observation is that for a fixed level of accuracy, the classification time can be reduced by using properly designed decision trees, i.e. the efficiency is improved relative to that of the conventional classifier.

It is also important to verify the validity of the search procedure, especially when empirical methods, such as calculation of classification probabilities through statistical distances (Appendix C), are involved. Simulated

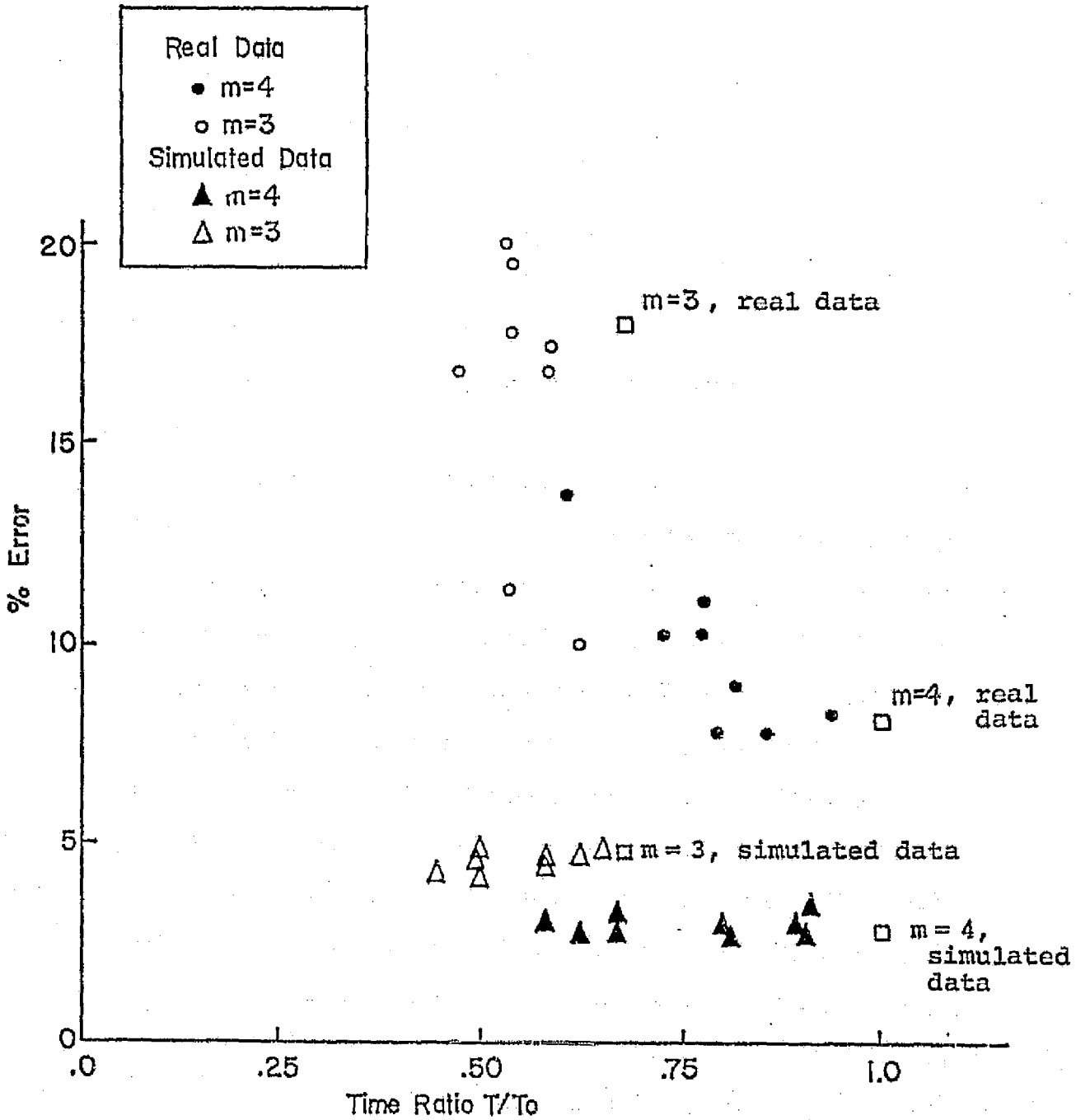


Figure 5.13 Performance of Decision Tree Classifiers in Classifying Real and Simulated Data Sets.

data sets of multivariate normal distributions have been used for an "accurate" evaluation, because real data of each class (after unimodal refinement) are not exactly normally distributed.

Experiment 5.10 Nine classes of data with 1,000 samples for each class were generated according to multivariate normal distributions with means and variances the same as those calculated for the classes in Experiments 5.8 and 5.9. Classifiers designed in Experiments 5.9 were used to classify this simulated data set. The results in terms of efficiency and accuracy are plotted as triangles in the lower portion of Fig. 5.13.

The probability P_i that a classification path will pass through node d_i has been estimated during the design (Eq. 4.6). As a result the total amount of computation time for a given design per sample can be estimated by summing up the products of probabilities and computation time of all stages. For all classifiers designed in Experiment 5.10, the estimated units of computation time are plotted versus the measured units in Fig. 5.14. The estimated values are generally a few percent lower than measured values; this is because in a real case P_i is a sum of the probabilities of correct and misclassifications, but in the empirical method described in Appendix C the probability of misclassification is not included in P_i for the reason of simplicity and this leads to the underestimation.

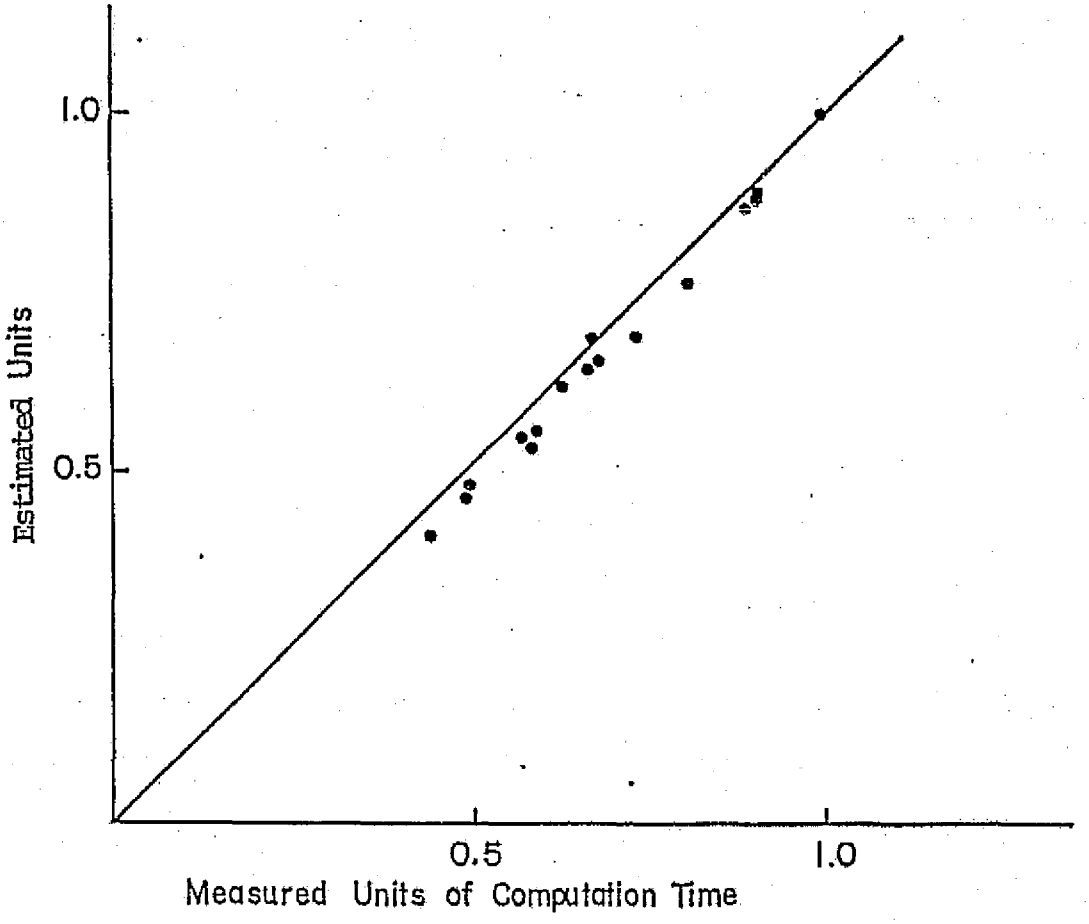


Figure 5.14 Estimated and Measured Classification Time of Decision Tree Classifiers in Classifying Simulated Data Sets.

The accuracy of classifying simulated data being higher than that of real data is expected because real data are not exactly normally distributed. For a given dimensionality, it is noticed that with increasing efficiency the error rate is essentially kept at the same level. This suggests that the sequential partitioning of the feature space by the designed decision trees is very effective. The results plotted in Fig. 5.14, which demonstrate the closeness of the predicted and the measured results, reflect the validity of the method in approximating the classification probability, which is an important step in the search approach. Because the error rates do not change much, the effects of the tradeoff constant K can hardly be observed; this will be studied in later experiments.

The following two experiments are performed on ERTS-1 Satellite MSS data. The spectral dimensionality of this data is four, and all fifteen feature combinations have been selected for search.

Experiment 5.11 Twenty six spectral classes were obtained in a forest area by means of an Eucledean Distance clustering algorithm [43]. These classes were then grouped into five groups: conifer, deciduous, agricultural area, water and bare rock, which represent the basic coverage types in that forestry area. The statistics* of these twenty six classes were used to classify an area of 12,467

*Data sets were provided by courtesy of Michael Fleming.

samples. One hundred and twenty four test fields with a total of 773 samples were selected from available ground truth information for testing purposes.

The input to the search procedure are options and the class group information which modifies the zero-one error matrix and also helps to determine whether further classification of a set of classes in a node is necessary. The assumption of equal a priori probabilities for all spectral classes was also used in the design.

By utilizing the search procedure, a number of decision tree classifiers were designed. A typical tree structure is shown in Fig. 5.15. In the upper figure, the numbers in brackets are features, and the others are class designations. For the nonterminal nodes, the classes in the upper row are the representative classes; their pooled statistics are used to represent that node they are in. In the lower portion of Fig. 5.15, the tree structure shown above is drawn in terms of symbols, each of which indicates a subgroup of classes. The mapping of classes, symbols and groups is indicated in Table 5.5.

Classification results of the classifiers designed along with their options are listed in Table 5.6. The form of this table is similar to that of Table 5.4, except items labeled by δN are added. The number δN for each classifier was determined by counting the points classified differently by using the designed classifier with respect to the results

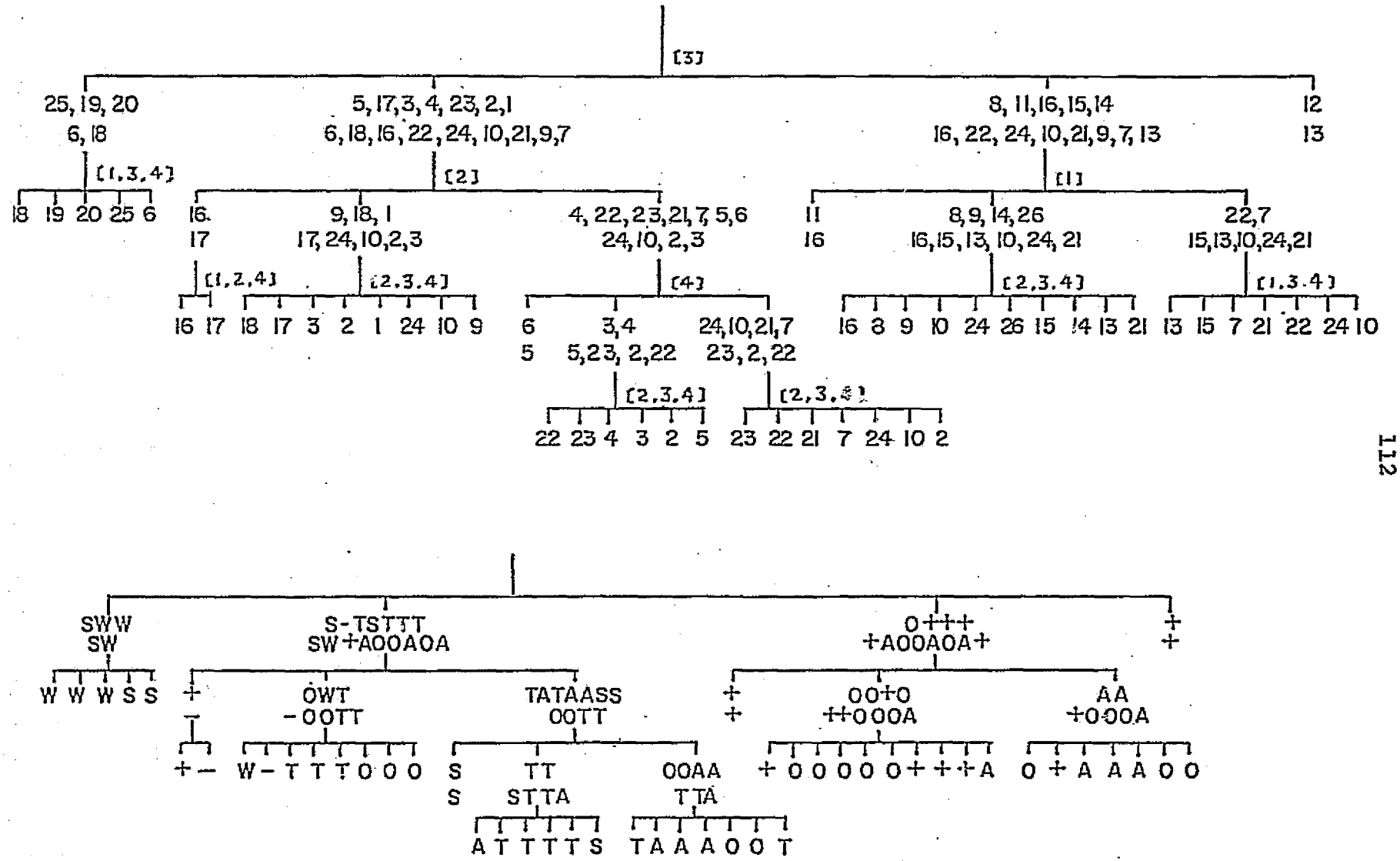


Figure 5.15 An Example of Decision Tree Classifier Designed for 26 Class Classification in Experiment 5.11.

Table 5.5 Class Group Information of the Twenty Six Spectral Classes in Experiment 5.11.

GROUP	SUBGROUP	SYMBOL	SPECTRAL CLASSES COMPRISED
CONIFER	PINE	T	1, 2, 3, 4, 23
	SPRUCE, FIR	S	5, 6, 25
DECIDROUS	ASPEN	A	7, 21, 22
	OAK	O	8, 9, 10, 24, 26
AGRICULTURAL		+	11, 12, 13, 14, 15, 16
BARE		-	17
WATER		W	18, 19, 20

Table 5.6 Decision Tree Design Parameters and Associated Classification Results of Experiment 5.9.

DESCRIPTIONS				CLASSIFICATION RESULTS					
DISTANCE	PARAMETERS			m = 4			m = 3		
CRITERION	T_h	K	ID	T/To	E(%)	SN	T/To	E(%)	SN
B_T	1900	5.0	1	0.33	6.3	245	0.23	5.8	532
		10.0	2	0.26	5.3	191	0.20	5.4	382
		20.0	3	0.42	5.2	99	0.30	5.4	346
		40.0	4	0.46	5.2	78	0.32	4.9	378
		100.0 200.0	5	0.61	5.2	13	0.41	4.9	371
		1000.0	6	1.0	5.2	0	0.64	6.6	412
	1950	5.0	7	0.42	5.0	110	0.30	6.0	367
		10.0 20.0	8	0.35	5.0	104	0.26	6.0	353
		40.0	9	0.43	5.2	91	0.31	6.0	353
		100.0 200.0	5	0.61	5.2	13	0.41	5.5	371
		1000.0	10	0.87	5.2	2	0.57	6.6	416
	1999	–	11	0.57	5.0	11	0.40	4.8	369
	CONVENTIONAL		6	1.0	5.2	0	0.64	6.6	412
D_T	1900	5.0	12	0.35	5.4	76	0.26	5.7	359
		10.0 20.0	13	0.34	5.0	33	0.27	4.5	379
		40.0 100.0 200.0	14	0.46	5.0	31	0.32	4.5	393
		1000.0	15	0.54	5.2	21	0.36	6.6	423
	1950	5.0 10.0	16	0.36	5.0	34	0.26	4.5	379
		20.0	17	0.37	5.0	33	0.27	4.5	379
		40.0	18	0.39	5.0	35	0.28	4.5	372
		100.0 200.0	19	0.44	5.0	31	0.31	4.5	394
		1000.0	20	0.53	5.3	29	0.37	6.9	423
	1999	–	21	0.59	5.2	9	0.41	4.9	375
	CONVENTIONAL		6	1.0	5.2	0	0.64	6.6	412

of using a conventional procedure with all four features. The accuracies and efficiencies (measured by ratios T/T_o) of Table 5.6 are plotted in Fig. 5.16a and 5.16b for B_T and D_T respectively. Various values of the change of classification δN versus tradeoff constant K are plotted in Fig. 5.17. And finally the efficiencies T/T_o with respect to the tradeoff constant K are plotted in Fig. 5.18a and Fig. 5.18b according to the values in Table 5.6 for B_T and D_T respectively.

Comparing Fig. 5.16a and Fig. 5.16b, there does not appear to be any significant difference in the performance of classifiers designed by using D_T or B_T as the separability criterion. Some of the results in Fig. 5.16b are better than those of Fig. 5.16a. And this observation is contradictory to the results shown in Experiment 5.7, there B_T is shown to be better than D_T as separability criterion in finding optimal feature subsets for a pair of classes. This contradiction can be explained by the nearly equal effectness of average B_T and D_T, which has been mentioned in the end of Section 5.3.3.1, because in this experiment the numbers of classes in terminal decisions are often much greater than two. It is also noted from Table 5.6 that results of decision tree classifiers with a maximum of three features in terminal decisions are better than the results using a conventional classifier with all four features, or the results using decision tree classifiers with four features in terminal decisions.

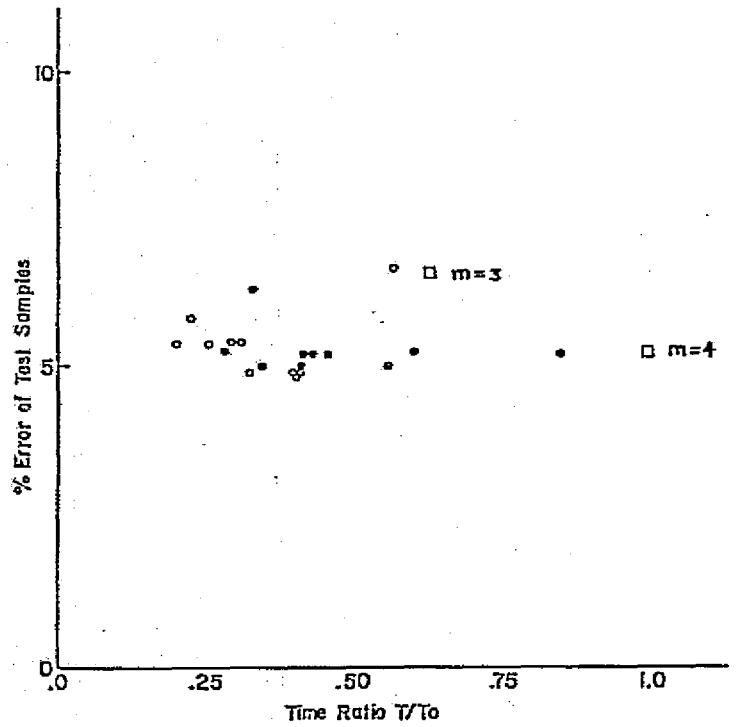


Figure 5.16a Performance of Decision Tree Classifiers Designed with E_T. •: m=3, •: m=4, (□ : Conventional Classifiers, m=3,4).

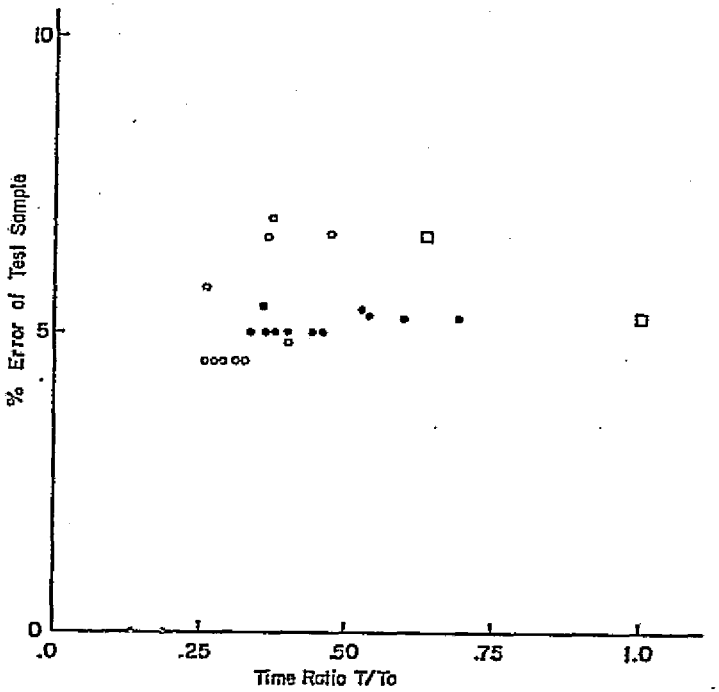


Figure 5.16b Performance of Decision Tree Classifiers Designed with D_T, o: m=3, • : m=4.

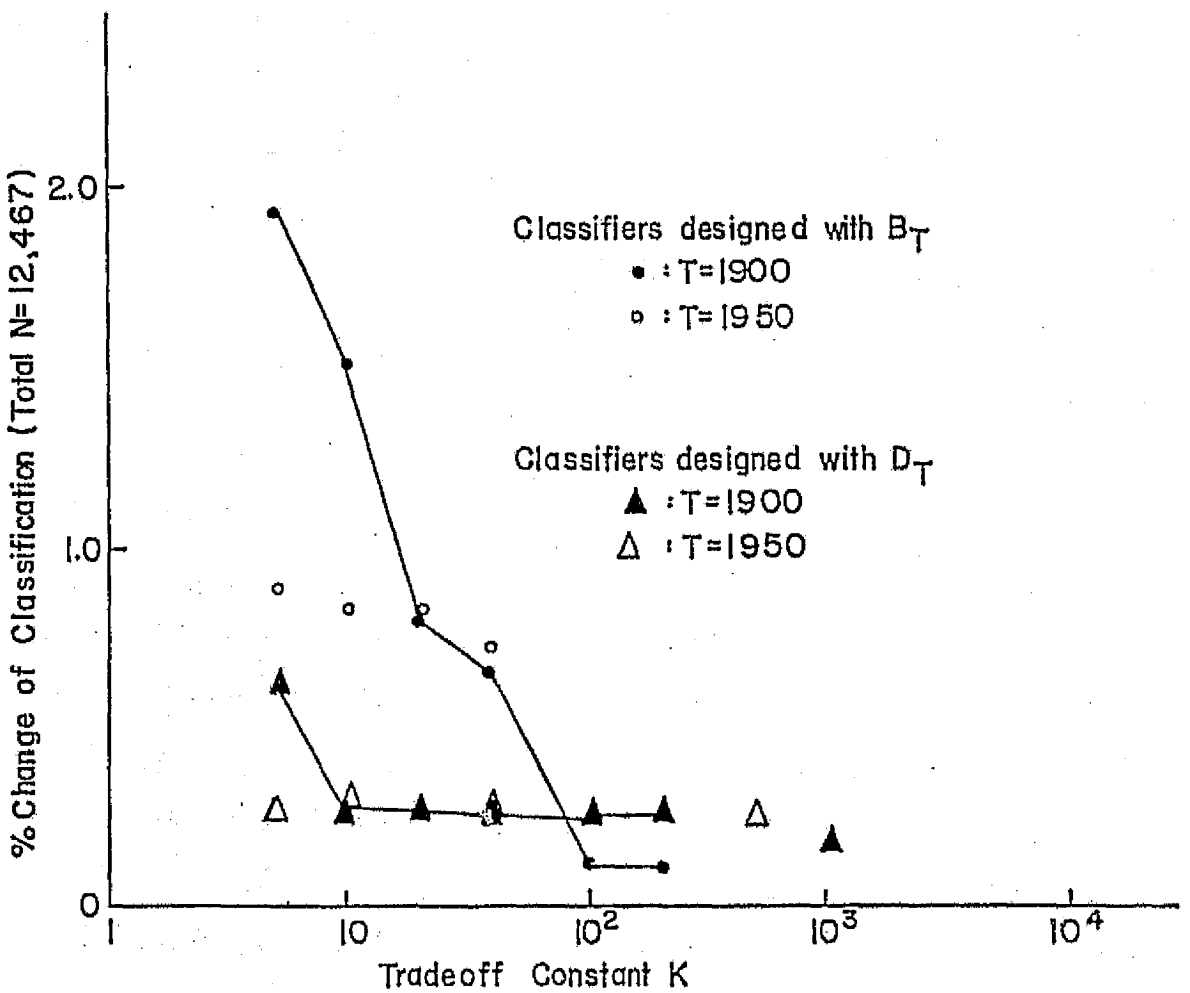


Figure 5.17 Change of Classification (%) Versus Tradeoff Constant K.

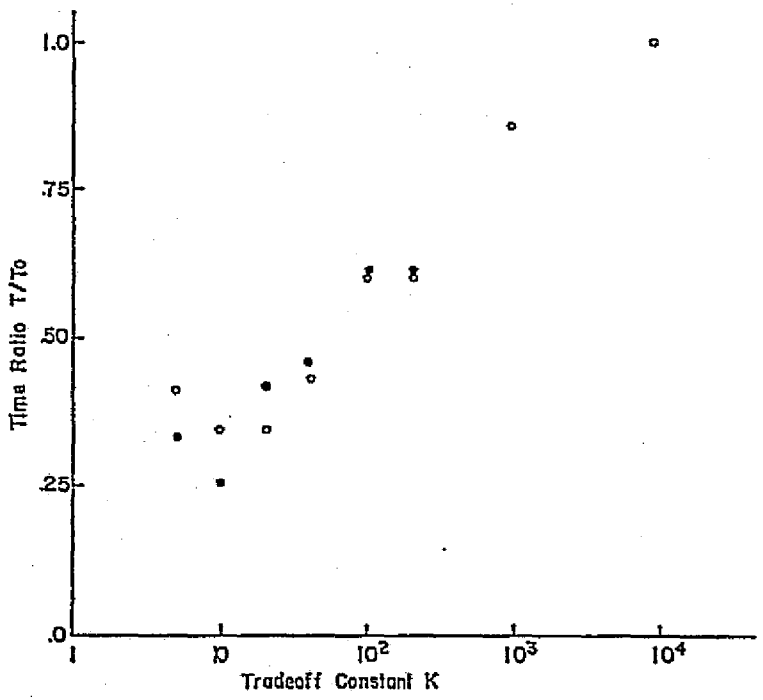


Figure 5.18a Time Ratio Versus K for the Classifiers Designed with B_T in Experiment 5.11.
∘: T_h=1950, •: T_h=1900.

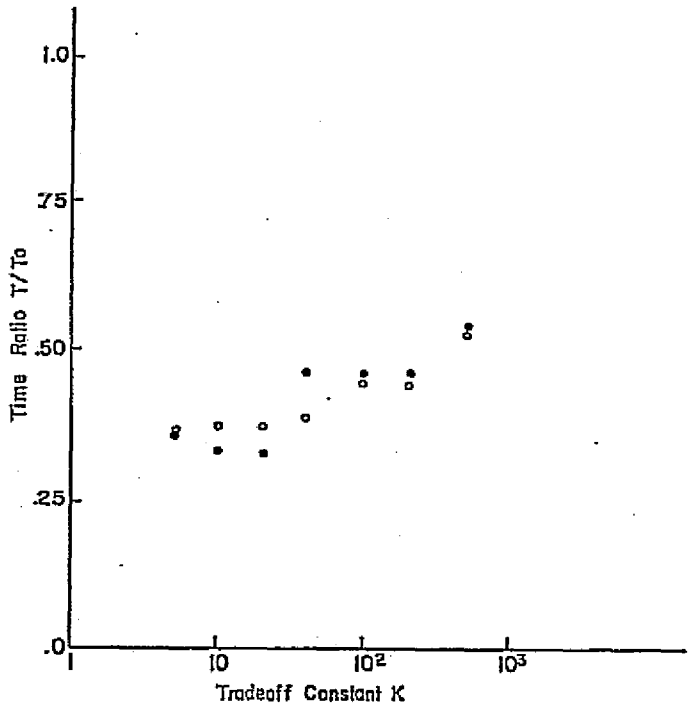


Figure 5.18b Time Ratio Versus K for the Classifiers Designed with D_T in Experiment 5.11.
∘: T_h=1950, •: T_h=1900.

This fact is probably due to the presence of the dimensionality problem. The slight improvement (within one percent) in the accuracy is not considered significant because only 773 samples have been tested. Nevertheless, the fact that the overall performance of the classifiers designed by the search method can be better than the overall performance of conventional classifiers is again demonstrated.

Another way to evaluate the performance of designed classifiers is by checking the classification results point by point (as compared to the results made by a conventional classifier). Here we actually assume that the decision boundaries of a conventional procedure using all features are optimal. The purpose of the check is to observe whether the boundaries of a decision tree classifier coincide with the optimal boundaries. Since the class group information is part of the input in designing a decision tree classifier, only misclassification between different groups are counted. These results are listed in Table 5.6, and are plotted in percentage quantity in Fig. 5.17 (only results of $m = 4$ are plotted for comparison) with respect to the tradeoff constant K. The percentages of change are all very small as shown. It is also observed that the different δN for design with B_T as a separability criterion is relatively more sensitive to the change of K than those designed with D_T.

The efficiencies with respect to the input tradeoff constant K are also plotted (Fig. 5.18a and Fig. 5.18b). Again

those designed using B_T appear to be relatively more sensitive to the change of K.

Experiment 5.12 Twenty eight spectral classes are found in the satellite MSS data in the San Jose urban area. These classes are then grouped into eight meaningful groups as shown in Table 5.7 according to ground truth information. The procedure for this experiment is same as in Experiment 5.11, except that no test samples are available, so that only the resulting changes (δN) can be determined. In all 10,040 samples are classified. The classification results of classifiers designed by the search procedure are listed in Table 5.8. (The notations in this figure are same as those used in Table 5.6.)

The fact that the performance of the decision tree classifiers can be better than that of the conventional classifiers is again demonstrated in this experiment. That is for a negligible change in classification results, the computation time can be greatly reduced; or for the same amount of δN (or less than 37° of the conventional classifier with $m = 3$) the computation time measures for decision trees are in most cases le[illegible] than that of the conventional classifier ($m = 3$).

5.3.3.3 Discussion

For the class of binary decision trees, feature selection using the Bhattacharyya distance has been found

Table 5.7 Class Group Information of the Spectral Classes in Experiment 5.12.

FUNCTIONAL LAND - USE	SYMBOL	SPECTRAL CLASSES COMPRISED*
COMMERCIAL - INDUSTRAL	1	1, 2, 3, 14
MOBILE HOMES	V	5
RESIDENTIAL	M	6, 9, 10, 13, 15, 16, 17 18, 19, 20, 21
PARKING LOTS	•	8, 22
UNIMPROVED OPEN SPACE (BARE)	-	11
UNIMPROVED OPEN SPACE (TREES)	/	23, 24, 25, 26, 28, 29, 30
IMPROVED OPEN SPACE (IRRIGATED)	+	12
WATER	O	27

*CLASSES 4, 7 are deleted.

Table 5.8 Decision Tree Design Parameters and Associated Classification Results of Experiment 5.12.

DESCRIPTIONS				CLASSIFICATION RESULTS			
DISTANCE	PARAMETERS		ID	m = 4		m = 3	
	T_h	K		T/To	SN	T/To	SN
B_T	1900	5.0	1	0.34	136	0.25	453
		10.0 20.0	2	0.40	85	0.29	410
		40.0	3	0.44	77	0.31	446
		100.0 200.0	4	0.53	77	0.36	446
	1950	5.0 10.0 20.0	5	0.39	32	0.28	374
		40.0 100.0	6	0.46	24	0.32	389
		200.0	7	0.52	24	0.36	389
	1999	0.0	8	0.80	0	0.54	203
	CONVENTIONAL		9	1.00	0	0.65	370
D_T	1900	5.0	10	0.39	89	0.28	413
		10.0	11	0.42	88	0.30	412
		20.0 40.0 100.0	12	0.59	56	0.40	409
		200.0	13	0.79	130	0.52	458
	1950	5.0 10.0 20.0	14	0.85	118	0.28	435
		40.0	15	0.40	112	0.29	415
		100.0 200.0	16	0.41	6	0.35	366
	1999	0.0	17	0.50	37	0.57	390
	CONVENTIONAL		18	1.00	0	0.65	429

to be more effective than using the Divergence. For decision tree classifiers designed by the search method, the two transformed separability criteria B_T and D_T seem to be of comparable effectiveness for feature selection. Since less computation is required in calculating Divergence (for normal distributions), this makes the transformed Divergence D_T preferable to the transformed Bhattacharyya distance B_T.

By observing the results of previous experiments, for general classification the recommended threshold value T for the search can be set as 1950, and the tradeoff constant K can be set at 20.0. If T is set as 1999 or its maximum value (i.e., 2000), the classification results of the designed decision trees are almost the same as the results of conventional classifiers; net improvement in efficiency is also observed in these cases.

The cost of search is another important factor in determining the usefulness of the search procedure. It is roughly proportional to the number of feature subsets searched and the number of classes. In Experiment 5.9 using nine classes, to design a tree the average computation time using a large computer (IBM 360/67) is about ten seconds for fifteen feature subsets. In Experiment 5.11 and 5.12, the average computation time to design a tree is about forty seconds.

CHAPTER 6

CONCLUSION

6.1 Summary of Results

The dimensionality problem in multiclass and multivariate classification has been studied both theoretically and experimentally. The results confirm the existence of this phenomenon; thus one must come to the conclusion that one must be cautious in choosing the feature dimensionality for classification when there are only a limited number of training samples available to estimate data distributions. Although reliable methods which enable one to predict the optimal dimensionality have not been found, the basic study presented in this report provides additional knowledge to pattern recognition researchers and users concerning the effect of insufficient number of training samples on classification accuracy.

The major objective of the entire work is to develop multistage decision tree classifiers. The above study is one of the efforts in understanding the utility of such classifiers. Another meaningful result from these efforts is the derivation of the upper bounds on logic efficiency in multiclass classification. In a practical problem these bounds usually can not be attained, but they imply that

some type of classification procedure can be more efficient than the conventional procedure, i.e., the usual pointwise maximum likelihood decision rule; and one of the suggested procedures is the decision tree procedure. The study of logic efficiency and the dimensionality problem actually leads to some necessary conditions on efficient classification.

To design decision tree classifiers, several design approaches have been proposed in Chapter 4. In the first two approaches, human interaction is heavily involved in many aspects. The performance of the designed classifier thus will depend heavily on the experience of the person who designs the classifier. In the optimization approach the decision tree classifiers are designed by a preprogrammed process. Man-machine interactions are minimized, so that the need for a highly trained analyst is reduced, although the analyst is still required to supply certain parameters and training sets.

There are two separate design procedures in the optimization approach. One is aimed specifically at classifiers with higher accuracy. The design procedure is very straightforward. The other design procedure uses a heuristic search strategy. Due to the difficulty in representing the tree structure and the lack of theoretically verified method to predict the classifier performance, several empirical methods have been incorporated in the search strategy. And the strategy as can be noticed involves many different procedures. Both of these facts raise difficulty in verifying the validity of the search strategy. The basic point is that when both

a practical solution and theoretical perfection can not be achieved simultaneously, then one tends to choose the former. Through the experimental results, the fact that the performance of classifiers designed by the search procedure are better in most cases than that of the conventional procedure is demonstrated. Also one can observe the fact that performance does change with respect to different input parameters in a predictable manner.

6.2 Suggestions for Further Research

Predicting the optimal feature dimensionality is an important step for optimal classification. Other approaches which have not been investigated in this work, such as analyzing the principle components, can be pursued.

The bound on logic efficiency suggests another type of efficient procedure. That is the block or sample classifier. At Purdue University, several kinds of sample classifiers for remote sensing classification have been studied [59-61] or are currently under investigation. Generally, they classify many resolution elements at a time; and in general the classification accuracy is improved because sample statistics provide more information than a single data vector. A systematic approach to design block classifiers which focus on higher classification efficiency also can be proposed for further investigation.

Several approaches towards the design of decision tree classifiers have been studied in this report. All of

the designed classifiers are point classifiers, and context information has been ignored in classifying unknown samples. Since the class designations of successive samples in multispectral remotely sensed data generally are not independent, context information is certainly very helpful in further improving the classification accuracy. Thus, how to extract the context information and then utilize it in point classifiers (one-stage or multistage) can really be a very interesting and rewarding research project.

LIST OF REFERENCES

[1] K. Fu, D. Landgrebe, and S. Phillips, "Information Processing of Remotely Sensed Data," Proceedings of the IEEE, Vol. 57, pp. 639-653.

[2] G. Nagy, "Digital Image-Processing Activities in Remote Sensing for Earth Resource," Proc. of IEEE, Vol. 60, pp. 1177-1200, October, 1972.

[3] H. Raiffa, Decision Analysis, p. 10, Addison-Wesley, Reading, Massachusetts, 1970.

[4] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley, 1973.

[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1972.

[6] W. S. Meisel, Computer-Oriented Approaches to Pattern Recognition, Academic Press, 1972.

[7] K. S. Fu and P. H. Swain, "On Syntatic Pattern Recognition," in Software Engineering, Vol. II, pp. 155-182, J. T. Tou, ed. Academic Press, New York, 1971.

[8] Y. C. Ho and A. K. Agrawala, "On Pattern Classification Algorithms Introduction and Survey," Proc. of IEEE, Vol. 56, pp. 2101-2114, December, 1968.

[9] L. N. Kanal, "Interactive Pattern Analysis and Classification Systems: A Survey and Commentary" Proc. of IEEE, Vol. 60, pp. 1200-1215, October, 1972.

[10] G. Nagy, "State of the Art in Pattern Recognition," Proc. IEEE, Vol. 56, pp. 836-861, May, 1968.

[11] A. Wald, Sequential Analysis, Wiley, New York, 1947.

[12] K. S. Fu, Sequential Method in Pattern Recognition and Machine Learning, Academic Press, 1968.

[13] R. L. Mattson and J. E. Dammann, "A Technique for Determining and Coding Subclasses in Pattern Recognition Problem," IBM Journal, July, 1965.

[14] W. S. Meisel and D. A. Michalopoulos, "A Partitioning Algorithm with Application in Pattern Classification and the Optimization of Decision Tree," IEEE Trans. on Computer, Vol. C-22, pp. 93-103, January, 1973.

[15] M. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, Cambridge, Massachusetts, 1969.

[16] N. J. Nilsson, Learning Machines, McGraw-Hill, New York, 1965.

[17] W. G. Eppler, "An Improved Version of the Table Look-Up Algorithm for Pattern Recognition," Summaries 9th Inter. Symp. on Remote Sensing of Environment (U. of Michigan, Ann Arbor), pp. 86-88, April, 1974.

[18] H. Fukunaga and D. R. Olsen, "Piecewise Linear Discriminant Functions and Classification Errors for Multiclass Problems, IEEE Trans. Information Theory, Vol. IT-16, pp. 99-100, January, 1970.

[19] M. Nadler, "Error and Reject Rates in Hierarchical Pattern Recognizer," IEEE Trans. on Computer, Vol. C-20, December, 1971.

[20] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," IEEE Trans. Information Theory, Vol. IT-9, pp. 64-74, April, 1963.

[21] F. Jelinek, "A Fast Sequential Decoding Algorithm Using a Stack," IBM J. Res. Develop., Vol. 13, pp. 675-685, November, 1969.

[22] O. L. Mangasarian, "Linear and Nonlinear Separation of Patterns by Linear Programming," Operations Research, 13, pp. 444-452, May-June, 1965.

[23] Y-C. Ho, and R. L. Kashyap, "An Algorithm for Linear Inequalities and its Applications," IEEE Trans. on Elec. Comp., Vol. EC-14, pp. 683-688, October, 1965.

[24] K. S. Fu, Y. T. Chien, and G. P. Cardillo, "A Dynamic Programming Approach to Sequential Pattern Recognition," IEEE Trans. on Computer, December, 1967.

[25] J. R. Slagle and R. C. T. Lee, "Application of Game Tree Searching Techniques to Sequential Pattern Recognition," Communications of the ACM, Vol. 14, No. 2, February, 1971.

[26] S. E. Estes, "Measurement Selection for Linear Discriminants Used in Pattern Classification," IBM Corporation, San Jose, Calif., Research Report RJ-331, April, 1956.

[27] D. C. Allais, "The Selection of Measurements for Prediction," Technical Report No. 6103-9, Stanford Electronics Laboratories, November, 1964.

[28] G. F. Hughes, "On the Mean Accuracy of Statistical Pattern Recognizers," IEEE Trans. on Information Theory, Vol. IT-14, No. 1, pp. 55-63, January, 1968.

[29] K. Abend and T. J. Harley, Jr., "Comments 'On the Mean Accuracy of Statistical Pattern Recognizers'," IEEE Trans. Info. Theory, Vol. IT-15, pp. 420-421, May, 1969.

[30] B. Chandrasekaran and T. J. Harley, Jr., "Comments 'On the Mean Accuracy of Statistical Pattern Recognizers'," IEEE Trans. Info. Theory, Vol. IT-15, pp. 421-423, May, 1969.

[31] B. Chandrasekaran, "Independence of Measurements and the Mean Recognition Accuracy," IEEE Trans. Info. Theory, Vol. IT-17, pp. 452-456, July, 1971.

[32] L. M. Kanal and B. Chandrasekaran, "On Dimensionality and Sample Size in Statistical Pattern Classification," Proc. NEC, 24, 2-7, 1968; also in Pattern Recognition, 3, 225-234, October, 1971.

[33] K. S. Fu and P. J. Min, "On Feature Selection in Multiclass Pattern Recognition," Tech. rept. No. TR-EE68-17, School of Electrical Engineering, Purdue University, Lafayette, Indiana, July, 1968.

[34] A. G. Wacker and D. A. Landgrebe, "The Minimum Distance Approach to Classification," Ph.D. Dissertation, Purdue University, 1972.

[35] E. Parzen, "On Estimation of a Probability Density Function and Mode," Ann. Math. Stat., 33, pp. 1065-1076, September, 1962.

[36] A. Wald and J. Wolfowitz, "Optimum Character of the Sequential Probability Ratio Test," Ann. Math. Statist. 19, pp. 326-339, 1948.

[37] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, 1949.

[38] R. Gallager, Information Theory and Reliable Communication, Wiley, N.Y., 1968.

[39] D. A. Landgrebe, "Systems Approach to the Use of Remote Sensing," To be published as a chapter in "Remote Sensing of Environment" edited by J. Lintz, Jr. and Davis S. Simonett, Addison-Wesley, 1974.

[40] R. M. Hoffer, "Agricultural and Forest Resource Surveys from Space," LARS Information Note 100972, Purdue University, October, 1972. Also an invited paper presented at the 23rd International Astronautical Congress, Vienna, Austria, October 8-15, 1972.

[41] C. W. Marshall, Applied Graph Theory, p. 18, Wiley-Interscience, 1971.

[42] J. E. Hopcraft and J. D. Ullman, Formal Languages and Their Relation to Automata, Reading, Mass., Addison-Wesley, 1969.

[43] T. L. Phillips, ed., LARSYS User's Manual, Laboratory for Applications of Remote Sensing, Purdue U., W. Lafayette, Indiana, 1973.

[44] R. E. Bellman and S. E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, N.J., 1962.

[45] A. Bhattacharyya, "On a Measure of Divergence between Two Statistical Populations Defined by their Probability Distributions," Bulletin of the Calcutta Mathematical Society, V. 35, No. 3, pp. 99-110, September, 1943.

[46] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," IEEE Trans. on Communication Technology, Vol. Com-15, pp. 52-60, February, 1967.

[47] J. Wozencraft and J. Jacobs, Principles of Communications Engineering, Wiley, N.Y., 1965.

[48] M. H. DeGroot, Optimal Statistics Decisions, p. 142, McGraw-Hill, 1970.

[49] N. J. Nilsson, Problem Solving Methods in Artificial Intelligence, McGraw-Hill, 1971.

[50] J. R. Slagle, Artificial Intelligence: The Heuristic Programming Approach, McGraw-Hill, 1971.

[51] E. L. Lawler and D. E. Wood, "Branch-and-Bound Methods: A Survey," Operation Research, 14, pp. 699-719, July-August, 1966.

[52] H. Jeffreys, Theory of Probability, p. 158, Oxford University Press, 1948.

[53] P. H. Swain and R. C. King, "Two Effective Feature Selection Criteria for Multispectral Remote Sensing," Proceeding on the First International Joint Conference on Pattern Recognition, October, 1973.

[54] S. J. Whitsitt and D. A. Landgrebe, "Simulation Techniques for Estimating Error in the Classification of Normal Patterns," LARS Information Note 040174, Laboratory for Applications of Remote Sensing, Purdue U., W. Lafayette, Indiana, April, 1974.

[55] C. Hastings, Approximations for Digital Computers, pp. 191-192, Princeton University Press, Princeton, N.J., 1955.

[56] C. Y. Chang, "Dynamic Programming as Applied to Feature Subset Selection in a Pattern Recognition System," IEEE Trans. on S.M.C., Vol. SMC-3, pp. 166-171, March, 1973.

[57] A. W. Whitney, "A Direct Method of Nonparametric Measurement Selection," IEEE Trans. Computer (Short Notes), Vol. C-20, pp. 1100-1103, September, 1971.

[58] D. A. Landgrebe, "Description and Results of the LARS/GE Data Compression Study," LARS Information Note 021171, LARS/PURDUE, November, 1971.

[59] T. Huang, "Per Field Classifier for Agricultural Applications," LARS Information Note 060569, LARS/PURDUE, May, 1969.

[60] T. V. Robertson, "Extraction and Classification of Objects in Multispectral Images," Proceeding of the Conference on Machine Processing of Remotely Sensed Data, Purdue U., W. Lafayette, Indiana, October, 1973, pp. (3b)27-31.

[61] J. N. Gupta, R. L. Kettig, D. A. Landgrebe and P. A. Wintz, "Machine Boundary Finding and Sample Classification of Remotely Sensed Agricultural Data," Proc. Conf. on Machine Processing of Remotely Sensed Data, Purdue U., W. Lafayette, Indiana, October, 1973, pp. (4b)25-30.

APPENDIX A

A DERIVATION ON DIMENSIONALITY PROBLEM

A.1 Derivation of the Mean Square Error of the Likelihood Ratio

If the probability densities are estimated quantities, the likelihood ratio which is a random variable of the random sample X does not equal to its true value. The mean square error of the ratios calculated based on the estimated densities will be approximated in the following derivation.

Assuming X_i, i=1,...,n are independent identically distributed (i,i,d) random vectors from an unknown multivariate normal distribution $N(M,\Sigma)$, the unknown density function N can be estimated through the statistics $\hat{M}$ and $\hat{\Sigma}$. They are

$$\hat{M} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{2.1a}$$

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \hat{M})(X_i - \hat{M})^T \tag{2.1b}$$

where X_i is a (mx1) vector

and n is the number of samples from a known category, assuming

$$n > m \tag{A.1}$$

The probability density function $f(\hat{M})$ of $\hat{M}$ is:

$$f(\hat{M}) = N(M, \frac{1}{n}\Sigma) \tag{A.2}$$

The distribution of $\hat{\Sigma}$ can be derived from the Wishart Distribution [A1] by writing

$$\hat{\Sigma} = \frac{1}{n-1} A$$

then the density of $A = \sum_{i=1}^{n} (X_i-\hat{M})(X_i-\hat{M})^T$ is

$$f(A) = \frac{|A|^{1/2(n-m-1)} \quad e^{-1/2\ trA\Sigma^{-1}}}{2^{1/2nm}\ \pi^{m(m-1)/4}\ |\Sigma|^{n/2} \prod_{i=1}^{m} \Gamma[1/2(n+1-i)]} \qquad (A.2)$$

for A positive definite and 0 otherwise

For two classes with equal a priori probability, the estimated log likelihood ratio is:

$$\hat{r}_{12}(X) = \log \frac{\hat{P}(X|\omega_1)}{\hat{P}(X|\omega_2)}$$

where $\hat{P}(X|\omega_i) = N(\hat{M}_i, \hat{\Sigma}_i)$ is the estimated probability density function of class ω_i, with i=1,2. The true value of $\hat{r}_{12}(X)$ is

$$r_{12}(X) = \log \frac{P(X|\omega_1)}{P(X|\omega_2)}$$

where $P(X|\omega_i) = N(M_i, \Sigma_i)$ are the true densities. The mean square error of $\hat{r}_{12}(X)$ is written as

$$E[(\Delta r)^2] = E[(\hat{r}_{12}(X)-r_{12}(X))^2] \qquad (A.4)$$

The integral expression of Eq. A.5 is given by

$$E[(\Delta r)^2] = 1/2 \int_{\hat{\Omega}} P(\hat{M},\hat{\Sigma}) \int_X [P_1(X)\{\log\frac{P_1(X)}{P_2(X)} - \log\frac{\hat{P}_1(X)}{\hat{P}_2(X)}\}^2 + P_2(X)\{\log\frac{P_2(X)}{P_1(X)} - \log\frac{\hat{P}_2(X)}{\hat{P}_1(X)}\}^2]\,dx\,d\hat{\Omega}$$

$$= 1/2 \int_{\hat{\Omega}} P(\hat{M},\hat{\Sigma}) \int_X [P_1(X)+P_2(X)][\log\frac{P_1(X)}{\hat{P}_1(X)} - \log\frac{P_2(X)}{\hat{P}_2(X)}]^2\,dx\,d\hat{\Omega} \qquad (A.5)$$

where $\hat{\Omega}$ indicates the estimated parameters, $\hat{M}$ and $\hat{\Sigma}$ denote the estimated mean and covariance respectively. And $P_i(X)$ stands for $P(X|\omega_i)$. The factor one half is included because of an assumption:

$$P(\omega_1) = P(\omega_2) = 1/2 \qquad (A.6)$$

Eq. A.5 can also be written as Eq. A.7, in terms of the cross product and the square of the logarithmic quantities, i.e.

$$E[(\Delta r)^2] = E_c + E_s \qquad (A.7)$$

where

$$E_c = E_{\hat{\Omega},X}[-2\cdot(\log\frac{P_1(X)}{\hat{P}_1(X)})\cdot(\log\frac{P_2(X)}{\hat{P}_2(X)})] \qquad (A.8a)$$

$$E_c = E_{\hat{\Omega},X}[(\log\frac{P_1(X)}{\hat{P}_1(X)})^2 + (\log\frac{P_2(X)}{\hat{P}_2(X)})^2] \qquad (A.8b)$$

and $E_{\hat{\Omega},X}$ indicates the expectation which is averaged over the space of X and $\hat{\Omega}$; the integral expression of $E_{\hat{\Omega},X}$ has been given in Eq. A.5.

With $\hat{\Omega}_1$ being independent of $\hat{\Omega}_2$, Eq. A.8a can be written as the product of two expectations. Further with $\hat{\Sigma}_i$ being independent of $\hat{M}_i$ (because the distribution of $\hat{\Sigma}_i$ is independent of M_i) and with $\hat{\Sigma}_i^{-1}$ being approximated by $\Sigma_i^{-1} - \Sigma_i^{-1}\delta\Sigma_i\Sigma_i^{-1}$ ($\delta\Sigma_i = \hat{\Sigma}_i - \Sigma_i$), the expectation of log $(P_i(X)/\hat{P}_i(X))$ yields the approximated value m/2n. Thus E_c is derived as:

$$E_c \doteq -\frac{m^2}{2n^2} \tag{A.9}$$

where n,m are the number of samples and features respectively.

However, the evaluation of E_s given by Eq. A.8b is more difficult. Theoretically, a closed form solution of E_s can be obtained because the density functions of $\hat{M}_i$ and $\hat{\Sigma}_i$ are known (Eq. A.2 and Eq. A.3, and the density function of $|\hat{\Sigma}_i|$ can also be derived from Eq. A.3), and the average over $[P_1(X)+P_2(X)]$ can be calculated by first factorizing the covariance matrices and then using the moment generating functions of $P_1(X)$ and $P_2(X)$. It can be seen that the final solution of this integration is very complicated. Instead of carrying out this exact derivation, an approximation (error quantities with variances lower than the order of 1/n are dropped) of E_s is calculated. First, we have

$$\begin{aligned}\log\frac{P_1(X)}{\hat{P}_1(X)} &= \log P_1(X|\Omega) - \log P_1(X|\hat{\Omega}) \\ &= 1/2 \log |\hat{\Sigma}_1| + 1/2(X-\hat{M}_1)^T\hat{\Sigma}_1^{-1}(X-\hat{M}_1) \\ &\quad - 1/2 \log |\Sigma_1| - 1/2(X-M_1)^T\Sigma_1^{-1}(X-M_1)\end{aligned} \tag{A.10}$$

Rewrite $\hat{M}_i$, $\hat{\Sigma}_i$, with i=1,2, as

$$\hat{M}_i = M_i + \delta M_i \tag{A.11a}$$

$$\hat{\Sigma}_i = \Sigma_i + \delta\Sigma_i \tag{A.11b}$$

With $\hat{M},\hat{\Sigma}$ being unbiased estimators as defined by Eq. 2.1, the delta-quantities in Eq. A.11 have the following properties (suffix i in Eq. A.11 has been dropped)

$$E[\delta M] = 0 \tag{A.12a}$$

$$E[\delta M \delta M^T] = 1/n\ \Sigma \tag{A.12b}$$

$$E[\delta\sigma_{ij}] = 0 \tag{A.13a}$$

$$E[(\delta\sigma_{ii})^2] = 2/n\ \sigma_{ii}^2 \tag{A.13b}$$

where $\delta\sigma_{ij}$ are elements of the matrix $\delta\Sigma$

With approximation on $\hat{\Sigma}^{-1}$ given below

$$(\Sigma+\delta\Sigma)^{-1} \cong \Sigma^{-1} - \Sigma^{-1}\,\delta\Sigma\Sigma^{-1} \tag{A.14}$$

Eq. A.10 can be expanded as follows:

$$\log\frac{P_1(X)}{\hat{P}_1(X)} = 1/2(X-M_1-\delta M_1)^T(\Sigma_1+\delta\Sigma_1)^{-1}(X-M_1-\delta M_1)$$
$$-1/2(X-M_1)^T\Sigma_1^{-1}(X-M_1)-1/2\ \log\frac{|\Sigma_1|}{|\Sigma_1+\delta\Sigma_1|}$$

$$\cong 1/2(X-M_1-\delta M_1)^T(\Sigma_1^{-1}-\Sigma_1^{-1}\delta\Sigma_1\Sigma_1^{-1})(X-M_1-\delta M_1)$$
$$-1/2(X-M_1)^T\Sigma_1^{-1}(X-M_1)\ -1/2\ \log\frac{|\Sigma_1|}{|\Sigma_1+\delta\Sigma_1|}$$

$$= (X-M_1)^T\Sigma_1^{-1}\delta M_1 + 1/2\ \delta M_1\Sigma_1^{-1}\delta M_1$$

$$-1/2(X-M_1-\delta M_1)\Sigma_1^{-1}\delta\Sigma_1\Sigma_1^{-1}(X-M_1-\delta M_1)$$

$$-1/2\ \log\frac{|\Sigma_1|}{|\Sigma_1+\delta\Sigma_1|} \qquad (A.15)$$

Assuming n is large, in approximating the expectation of the square of Eq. A.15, only products of the lowest orders of the delta-quantities are retained. Thus we have

$$\underset{\hat{\Omega},X}{E}\left[\left(\log\frac{P_1(X)}{P_1(X)}\right)^2\right] \cong E_1 + 1/4E_2 + 1/4E_3 + O(\frac{1}{n^2}) \qquad (A.16)$$

where

$$E_1 = \underset{\hat{\Omega},X}{E}\left[(X-M_1)^T\Sigma_1^{-1}\delta M_1\delta M_1^T\Sigma_1^{-1}(X-M_1)\right]$$

$$E_2 = \underset{\hat{\Omega},X}{E}\left[\{(X-M_1)^T\Sigma_1^{-1}\delta\Sigma_1\Sigma_1^{-1}(X-M_1)\}^2\right]$$

$$E_3 = \underset{\hat{\Omega},X}{E}\left[\left(\log\frac{|\Sigma_1+\delta\Sigma_1|}{|\Sigma_1|}\right)^2\right]$$

Notice the cross products of terms in Eq. A.15 are not included because they have zero expectations. The quantities E_1, E_2 and E_3 are then evaluated in the following manner:

$$E_1 = \underset{\hat{\Omega},X}{E}\left[(X-M_1)^T\Sigma_1^{-1}\delta M_1\delta M_1^T\Sigma_1^{-1}(X-M_1)\right]$$

$$= \underset{X}{E}\left[(X-M_1)^T\Sigma_1^{-1}\{\underset{\hat{\Omega}}{E}[\delta M_1\delta M_1^T]\}\Sigma_1^{-1}(X-M_1)\right]$$

$$= 1/n\ \underset{X}{E}\left[(X-M_1)^T\Sigma_1^{-1}(X-M_1)\right]$$

$$= 1/n \times 1/2\ \int_X[P_1(X)+P_2(X)]\left[(X-M_1)^T\Sigma_1^{-1}(X-M_1)\right]dx$$

For $X \varepsilon \omega_1$, the quadratic term in parenthesis is of chi-square distribution with m degrees of freedom.

So we have

$$\int_X P_1(X)\,[(X-M_1)^T \Sigma_1^{-1}(X-M_1)]\,dx = m \qquad \text{(A.17)}$$

For $X \varepsilon \omega_2$, the result of integration will be

$$\int_X P_2(X)\,[(X-M_1)^T \Sigma_1^{-1}(X-M_1)]\,dx$$

$$= 2 \log \frac{|\Sigma_2|}{|\Sigma_1|} + \operatorname{tr} \Sigma_2 \Sigma_1^{-1} + (M_1-M_2)^T \Sigma_1^{-1}(M_1-M_2) \qquad \text{(A.18)}$$

which is derived through the use of moment generating function. The method is described in Ref [5], pp. 63-65.

Combining Eq. A.17 and Eq. A.18, we have

$$E_1 = \frac{1}{2n}\left[m + 2\log \frac{|\Sigma_2|}{|\Sigma_1|} + \operatorname{tr}\Sigma_2\Sigma_1^{-1} + (M_1-M_2)^T\Sigma_1^{-1}(M_1-M_2)\right] \qquad \text{(A.19)}$$

The quantity E_2 will be calculated by first introducing the orthonormal matrix Φ which satisfies:

$$\Phi_1^T \Sigma_1 \Phi_1 = \Lambda_1 \quad \text{and} \quad \Phi_1^T \Phi_1 = I \qquad \text{(A.20)}$$

where Λ_1 is a diagonal matrix. Using Φ as the matrix for linear transformation, let

$$(X'-M_1') = \Phi_1 (X-M_1) \qquad \text{(A.21a)}$$

$$\delta\Lambda_1 = \Phi_1^T \delta\Sigma \Phi_1 \qquad \text{(A.21b)}$$

Inserting the unit matrix of Eq. A.20 into E_2, with the newly defined terms of Eq. A.21, E_2 can be written as

$$E_2 = \underset{\hat{\Omega},X}{E}\left[\{(X'-M_1')\Lambda_1^{-1}\delta\Lambda_1\Lambda^{-1}(X'-M_1')\}^2\right]$$

$$= \underset{\hat{\Omega},X}{E}\left[\{\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(x_i'-m_{1i}')(x_j'-m_{1j}')\delta\lambda_{1ij}}{\lambda_{1i}\lambda_{1j}}\}^2\right]$$

Eq. A.13a implies $E[\delta\lambda_{1ij}] = 0$. With each $\delta\lambda_{1ij}$ being independent with another, the above equation is simplifed as shown below

$$E_2 = \underset{\hat{\Omega},X}{E}\left[\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(x_i'-m_{1i}')^2(x_j'-m_{1j}')^2\delta\lambda^2_{1ij}}{\lambda_{1i}^2\lambda_{1j}^2}\right] \qquad (A.22)$$

The expectation of $\delta\lambda^2$ is derived [Ref. 5, pp. 250-251] as

$$E[(\delta\lambda_{ij})^2] = \frac{1}{n}(\lambda_i^2\delta_{ij} + \lambda_i\lambda_j)$$

where δ_{ij} is the Kronecker delta-function which equals 1 if i=j, and 0 otherwise. The suffix 1 of $\delta\lambda$ and λ as used in Eq. A.22 is dropped in the above expression. Substitute the above expression into Eq. A.22, we get

$$E_2 = \frac{1}{n}\underset{X}{E}\left[\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{(x_1'-m_{ij}')^2(x_j'-m_{ij}')^2}{\lambda_{1i}\lambda_{1j}} + \sum_{i=1}^{m}\frac{(x_i-m_{1i}')^4}{\lambda_{1i}^2}\right]$$

$$= \frac{1}{n}\underset{X}{E}\left[\left(\sum_{i=1}^{m}\frac{(x_i'-m_{1i}')^2}{\lambda_{ij}}\right)^2 + \sum_{i=1}^{m}\frac{(x_i'-m_{1i}')^4}{\lambda_{1i}^2}\right] \qquad (A.23)$$

When quantities in the bracket of Eq. A.23 are averaged with respect to $P_1(X)$, the integration can easily be evaluated. This is because the first summation is of chi-square distribution with m degrees of freedom; and in the second summation each term is the square of a random variable of chi-square distribution with one degree of freedom (for $X\varepsilon\omega_1$,

with the orthonormal transformation, x_i' is now uncorrelated with x_j' for $i \neq j$). Thus

$$E_2 = \frac{1}{2n}[(m^2+2m) + m(1+2)] + E_2'$$

$$= \frac{1}{2n}(m^2+5m) + E_2' \qquad (A.24)$$

where

$$E_2' = \frac{1}{2n} \int_{x'} P_2(X')[(\sum_{i=1}^{m} \frac{(x_i'-m_{ij}')^2}{\lambda_{1i}})^2 + \sum_{i=1}^{m} \frac{(x_i'-m_{1j}')^4}{\lambda_{2i}^2}]dx' \qquad (A.25)$$

The integration of Eq. A.25 is rather difficult to carry out; to simply the calculation the assumption of approximately equal covariances has been made, i.e.

$$\Sigma_1 \cong \Sigma_2 \qquad (A.26)$$

Eq. A.25 when solved with $\Sigma_1 = \Sigma_2$, gives

$$E_2' = \frac{1}{2n}[(m^2+2m+4D'+D'^2+2mD') + (3m+6D'+\sum_{i=1}^{m} \frac{(m_{1i}'-m_{2i}')^4}{\lambda_i^2})]$$

$$= \frac{1}{2n}[m^2+5m+10D'+2mD'+D'^2+\sum_{i=1}^{m} \frac{(m_{1i}'-m_{2i}')^2}{\lambda_i^2}] \qquad (A.27)$$

where λ_i are the eigenvalues of the common covariance Σ

$m_{\ell i}'$, $\ell=1,2$, are the components of mean vectors in the transformed space.

and $D' = (M_1-M_2)^T\Sigma^{-1}(M_1-M_2)$ is the divergence of two normal distributions with equal covariance.

With the assumption of Eq. A.26, an approximation can be made for E_2' of Eq. A.25, i.e.

$$E_2' \cong \frac{1}{2n}(m^2+5m+10D+2mD+D^2) \qquad (A.28)$$

where D is the Divergence of two multivariate normal distributions as defined in Eq. 2.9. Notice the last term of Eq. A.27 has been dropped because the summation is less than D^2 and most of the other terms of Eq. A.28.

Substituting E_2' into Eq. A.24, E_2 is now expressed as follows:

$$E_2 = \frac{1}{2n}(2m^2+10m+2mD+10D+D^2) \qquad (A.29)$$

Finally, for E_3 assuming the delta-quantities $\delta\lambda$ are small compared with λ, we have

$$\log\frac{|\Sigma_1+\delta\Sigma_1|}{|\Sigma_1|} \cong \log \frac{\prod_{i=1}^{m}(\lambda_{1i}+\delta\lambda_{1i})}{\prod_{i=1}^{m}\lambda_{1i}}$$

$$= \sum_{i=1}^{m} \log \frac{\lambda_{1i}+\delta\lambda_{1i}}{\lambda_{1i}}$$

$$\cong \sum_{i=1}^{m} \frac{\delta\lambda_{1i}}{\lambda_{1i}}$$

Substituting above into E_3,

$$E_3 \cong \underset{\hat{\Omega},X}{E} \left[\left(\sum_{i=1}^{m} \frac{\delta\lambda_{1i}}{\lambda_{1i}}\right)^2\right]$$

$$= \underset{\hat{\Omega}}{E} \left[\sum_{i=1}^{m} \left(\frac{\delta\lambda_{1i}}{\lambda_{1i}}\right)^2\right]$$

$$= \sum_{i=1}^{m} \frac{2}{n} \qquad (A.30)$$

$$= \frac{2m}{n}$$

Eq. A.30 is obtained because $\delta\lambda_i$ and $\delta\lambda_j$ are uncorrelated for $i \neq j$, and with $E[\delta\lambda_i]=0$, $E[\delta\lambda_i^2]=\frac{2}{n}\lambda_i$ according to Eq. A.22.

With E_1, E_2 and E_3 approximated, Eq. A.16 can now be expressed in terms of n,m and statistics parameters. An expression similar to Eq. A.16 can be obtained for the expected value of the square of $\log[P_2(X)/\hat{P}_2(X)]$. By adding them together, we get

$$E[(\Delta r)^2] = \underset{\hat{\Omega},X}{E}\left[\left(\log\frac{P_1(X)}{\hat{P}_1(X)}\right)^2 + \left(\log\frac{P_2(X)}{\hat{P}_2(X)}\right)^2\right] - \frac{m^2}{2n^2}$$

$$\cong \frac{1}{2n}(3m+2D) + \frac{1}{4}\cdot\frac{1}{n}(2m^2+10m+2mD+10D+D^2)$$

$$+ \frac{1}{4}\cdot\frac{4m}{n} + 0\left(\frac{1}{n^2}\right)$$

$$= \frac{1}{4n}(2m^2+20m+2mD+14D+D^2) + 0\left(\frac{1}{n^2}\right) \qquad \text{(A.31)}$$

Eq. A.31 is the approximate averaged mean square error of the estimated likelihood ratio $\hat{r}_{12}$. Several assumptions on which the approximation of Eq. A.31 is based are summarized as below:

1) $P(X|\omega_i) \sim N(M_i,\Sigma_i)$

2) $P(\omega_1) = P(\omega_2) = \frac{1}{2}$

3) $n_1 = n_2 = n$ and $n > m$

4) $\Sigma_1 \cong \Sigma_2$

In case the covariances are known, i.e. $\hat{\Sigma}_i=\Sigma_i$, better approximate solutions can be derived. The derivations are given as follows:

With $\delta\Sigma_i=0$, Eq. A.15 can be written as

$$\log \frac{P_i(X)}{\hat{P}_i(X)} = (X-M_i)\Sigma_1^{-1}\delta M_i + \frac{1}{2}\,\delta M_i\,\Sigma_1^{-1}\,\delta M_i$$

Substituting the above expression into Eq. A.8, the value of E_c equals $-\frac{2m^2}{n^2}$. The expression for E_s is now evaluated as

$$E_s = \underset{\hat{\Omega},X}{E}\left[(\log \frac{P_1(X)}{\hat{P}_1(X)})^2 + (\log \frac{P_2(X)}{\hat{P}_2(X)})^2\right] - \frac{2m^2}{n^2}$$

$$= \frac{1}{2n}(3m+2D) + \frac{1}{4}\sum_{i=1}^{2} \underset{\hat{\Omega}}{E}[(\delta M_i^T \Sigma_i^{-1}\delta M_i)^2]$$

Since δM_i has the density function $N(0,\frac{1}{n}\Sigma_i)$, the expected values of above equation can be evaluated. So we get

$$E_s = \frac{1}{2n}(3m+2D) + \frac{1}{2}\left(\frac{m^2+2m}{n^2}\right)$$

Substituting E_s and E_c into Eq. A.7 we have

$$E[(\Delta r)^2]_{\hat{\Sigma}=\Sigma} = \frac{3m+2D}{2n} + \frac{m^2+2m}{2n^2} - \frac{m^2}{2n^2}$$

$$= \frac{3m+2D}{2n} + \frac{m}{n^2} \qquad \text{(A.32)}$$

Eq. A.32 is the exact expression for $E[(\Delta r)^2]$ with known covariances.

A.2 Two Class Classification with Equal Covariance

With equal a priori probability, the logarithmic value of the likelihood ratio of two normal distributions with equal covariance is given by:

$$r_{12}(X) = \log \frac{P(X|\omega_1)}{P(X|\omega_2)}$$

$$= (M_1-M_2)^T\Sigma^{-1}X - \frac{1}{2}(M_1+M_2)^T\Sigma^{-1}(M_1-M_2) \qquad (A.33)$$

The maximum likelihood decision rule is set to be

$$r_{12}(X) \begin{cases} \geq 0 & X\varepsilon\omega_1 \\ < 0 & X\varepsilon\omega_2 \end{cases} \qquad (A.34)$$

Since X of a given class is of multivariate normal distribution, from the expression of Eq. A.33 it is clear that $r_{12}(X)$ is also normally distributed. The mean and variance of $r_{12}(X)$ are calculated as below:

$$E[r_{12}(X)|\omega_1] = -E[r_{12}(X)|\omega_2] = D/2 \qquad (A.35a)$$

$$V[r_{12}(X)|\omega_1] = V[r_{12}(X)|\omega_2] = D \qquad (A.35b)$$

where $D = (M_1-M_2)^T\Sigma^{-1}(M_1-M_2)$

In this special case, the probability of misclassification ε can be predicted according to

$$\varepsilon = \text{erf}\ (-\frac{\sqrt{D}}{2})$$

where $\text{erf}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{\alpha^2}{2}} d\alpha$

When density estimation is involved, the estimated value of $r_{12}(x)$ becomes

$$\hat{r}_{12}(x) = (\hat{M}_1-\hat{M}_2)\hat{\Sigma}^{-1}X-\frac{1}{2}(\hat{M}_1+\hat{M}_2)^T\hat{\Sigma}^{-1}(\hat{M}_1-\hat{M}_2) \qquad (A.37)$$

where the estimated parameters $\hat{M}_i$ and $\hat{\Sigma}_i$ are computed according to Eq. 2.1. The mean of Eq. A.37 can be calculated as

$$E[\hat{r}_{12}(X)|\omega_1] = -E[\hat{r}_{12}(X)|\omega_2] = D/2 \qquad (A.38)$$

Assumption that $\hat{\Sigma}_i$ is independent of $\hat{M}_i$ is made to obtain the above expression. The variance of Eq. A.37 with the mean given by Eq. A.38 can be calculated as

$$V[\hat{r}_{12}(x)|\omega_i] = V[\hat{r}_{12}(x)|\omega_i] + E[(\Delta r)^2|\omega_i] \qquad (A.39)$$

For equal covariance, we have

$$E[(\Delta r)^2|\omega_1] = E[(\Delta r)^2|\omega_2]$$

With the above expression being written as $E[(\Delta r)^2]$, substituting Eq. A.36b into Eq. A.39, we have

$$V[\hat{r}_{12}(X)|\omega_1] = V[\hat{r}_{12}(X)|\omega_2] = D+E[(\Delta r)^2]$$

So finally we arrive at

$$\varepsilon = \text{erf}\left(-\frac{D/2}{\{D+E[\Delta r^2]\}^{1/2}}\right)$$

That is

$$\varepsilon = \text{erf}\left(-\frac{1}{2}\left\{\frac{1}{D} + \frac{E[\Delta r^2]}{D^2}\right\}^{-\frac{1}{2}}\right) \qquad (2.11)$$

Reference

[A1] T. W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley, N.Y., 1958.

APPENDIX B

A NONSUPERVISED CLUSTERING PROCEDURE

B.1 Clustering Procedure

The nonsupervised clustering procedure which was used in the search method to design a decision tree classifier belongs to the class of graph theoretical methods for cluster analysis [B1]-[B4]. In the graph theoretical method, starting with a similarity graph which is in the form of a binary matrix $B = [b_{ij}]$ (such that $b_{ij}=1$ means elements i and j are similar), a sort strategy generally is incorporated to find sets of subgraphs which satisfy certain given criterion. If matrix elements $b_{ij}=1$ are scattered in the binary symmetric matrix B in a random fashion, the procedure for sorting will be very complicated. However, if elements of value 1 are all condensed along the diagonal of matrix B, the cluster sorting procedure can be simplified; that is, one can simply locate the "bottlenecks" along the belt of 1's and thereby extract cluster information.

Assuming the binary matrix B is obtained by applying a threshold to distances, to transform the original matrix into the one with elements 1's condensed along the diagonal is the same as rearranging the points into a new sequence such that points within a cluster and neighbors in the sequence. To

achieve that objective, the procedure for rearranging the point sequence is described below:

Step 1. For n points, one may form an (nxn) distance matrix $D=[d_{ij}]$ with a prespecified distance function.

Step 2. Set i=1. Initialize an n-vector $u(i)=[u_1(i), u_n(i)]$ such that

$$u_j(1)=0 \quad \text{for all } j=1,\ldots,n \tag{B.1}$$

and define $Q=(q_1,\ldots,q_n)$ as the initial index sequence, with

$$q_j=j \quad \text{for all } j=1,\ldots,n$$

Step 3. Find index K out of $1,\ldots,n$ such that the k-th rowsum of matrix D is a maximum. After K is found exchange the values of q_1 and q_k, so one will have $q_1=k$ and $q_k=1$ for later steps.

Step 4. Increase i by one. Set

$$u_j(i) = \alpha\, u_j(i-1) + d_{q_{i-1}q_j} \quad j=i,\ldots,n \tag{B.2}$$

where α is a constant $0\leq\alpha\leq 1$.

Step 5. Find index K, such that

$$u_k(i) = \min_{i\leq j\leq n} u_j(i) \tag{B.3}$$

Exchange the values of q_k and q_i as step 3 did for q_1 and q_k.

Step 6. If i is less than n, repeat step 4, otherwise proceed to next step.

Step 7. Rewrite $D=[d_{ij}]$ according to the newly arranged index sequence Q. That is

$$D' = [d'_{ij}]$$

$$\text{with} \qquad d'_{ij} = d_{q_iq_j} \tag{B.4}$$

Step 8. Apply threshold to d'_{ij} to obtain a binary matrix $B=[b_{ij}]$ such that

$$b_{ij} = \begin{cases} 1 & \text{if } d'_{ij} \leq T_n \\ 0 & \text{otherwise} \end{cases} \qquad \text{(B.5)}$$

where T is the threshold value.

The resulted binary matrix B has the property that elements with the value of 1 tend to cluster along the diagonal, thus simplifying the sorting procedure. An example of such a binary matrix B is shown in Fig. B.1b while the original distance matrix is shown in Fig. B.1a. The method of extracting clusters may differ for different cluster criteria, for our purpose to partition the feature space, the scheme will be explained as follows:

Suppose the binary matrix B of Eq. B.5 is "condensed", i.e.

$$\text{if } b_{ij}=1 \text{ then } b_{k\ell}=1 \text{ for all } i\leq k\leq j,\ i\leq \ell\leq j \qquad \text{(B.6)}$$

Then pairs of distinct vertices {i,j} (i<j) are selected for each occurance of

$$b_{ij}=0$$
$$\text{and} \quad b_{i,j-1}=b_{i-1,j-1}=b_{i+1,j}=1 \qquad \text{(B.7)}$$

After m such pairs (a_i,b_i), i=1,...,m are selected, order those pairs such that $a_i > a_j$ for all i>j. Examine those pairs with i from 2 to m, delete some pairs to make the remaining pairs (a'_j,b'_j), j=1,...,ℓ (ℓ<m) satisfy:

	1	2	3	4	5	6	7	8	9	10
1	0									
2	1601	0								
3	829	1446	0							
4	175	1010	322	0						
5	1723	40	645	1253	0					
6	1880	126	970	1583	42	0				
7	2000	1287	1938	1993	1342	1219	0			
8	1486	336	222	961	276	489	1830	0		
9	817	2000	1943	1547	2000	2000	2000	1995	0	
10	89	1187	434	14	1402	1689	1997	1109	1351	0

Figure B.1a A Distance Matrix for Ten Objects.

THRESHOLD = 1700

CONSTANT ALPHA = 0.6

	9	1	10	4	3	8	5	2	6	7
9	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0
10	1	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	1	1	0
8	0	1	1	1	1	1	1	1	1	0
5	0	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1
6	0	0	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	1	1	1	1

Figure B.1b The Binary Matrix (Similarity Graph) Obtained by Rearranging the Order of Objects and Applying Threshold on Distances.

$$a'_j < b'_j \qquad \text{for all } j=1,\ldots,\ell$$
$$\text{and} \quad b'_j < a_{j+1} \qquad \text{for all } j=1,\ldots,\ell-1 \tag{B.8}$$

By adding two numbers 1 and m, we may form new $\ell+1$ pairs from the old ℓ pairs. These newly formed pairs are

$$(1,a'_1),(b'_1,a'_2),\ldots(b'_\ell,m$$

each pair given above form a set of core points, e.g. a pair (i,j) gives the set $q_i,q_{i+1},\ldots,q_j$ where q_i is an element of the index sequence Q mentioned previously. Finally, for each core a cluster is formed by grouping points similar to at least one of the points in the core.

Referring back to the example in Fig. B.1b pair of indices (1,5), (2.9), (6,10) satisfy the condition in Eq. B.7. And according to Eq. B.8, the selected pairs* (1,5) and (6,10) give a new set of pairs (1,1), (5,6) and (10,10). The corresponding points for each pair can be found in the sequence Q (on top of the matrix in Fig. B.1b) as (9), (3.8) and (7). So, essentially three cores can be found from the binary matrix shown in Fig. B.1b. For each core, a cluster can be formed by having all the points associated with the core elements. The three clusters formed are (9,1,10,4), (1,10,4,3,8,5,2,6) and (5,2,6,7).

A flowchart of the clustering procedure follows in Figure B.2.

*The pair (2,9) has been omitted because the row number 2 is less than the column number 5 in the proceeding coordinate; but (2,9) itself gives the set of cores (1,2) and (9,10) which correspond to the points (9,1) and (6,7) as previously stated.

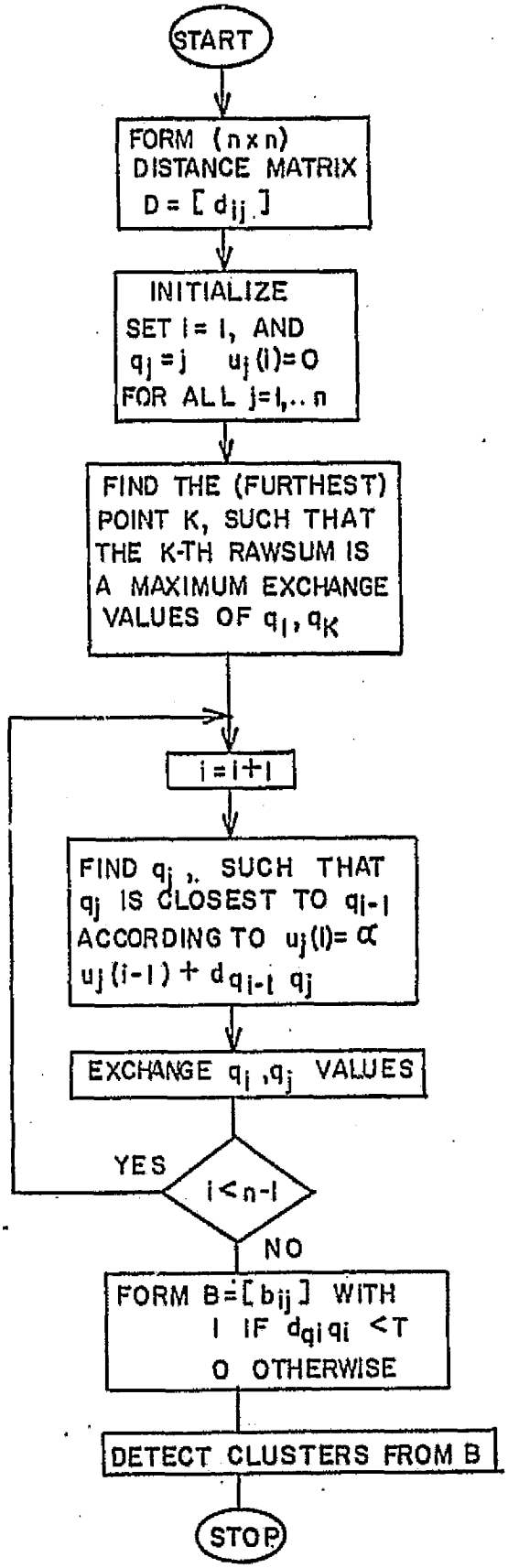


Figure B.2 A Flowchart of the Clustering Procedure.

In some cases if the binary matrix does not satisfy the condition given by Eq. B.8, i.e. there are 0's within the belt of 1's. Some modification has to be made. The simplest would be to fill in those undesired 0's with 1's such that Eq. B.8 can be satisfied. Actually the first seven steps to transform the distance matrix is designed to reduce such possibility of modification.

B.2 Theoretical Explanation

The procedure for extracting clusters from the rearranged matrix B which satisfies Eq. B.8 is much simpler than the procedure (Step 1 through Step 6) to rearrange the sequence of points. The steps involved in the former procedure in fact are logic operations. It is seen that the final $\ell+1$ pairs selected which satisfy the condition of Eq. B.8 give the nonoverlapped and mutually disassociated point subsets. In order to obtain such a "condensed" matrix as described by Eq. B.8 for cluster extracting, after the first seven steps of the point rearrangement procedure, the resultant distance matrix D' of Eq. B.4 must have the property that the small values of d_{ij} will be located closer to the matrix diagonal than those large values. Explanations will be given in the following paragraphs as how this property can be achieved by these steps of operation, and the rationale of these steps is.

Eq. B.2 is in the form of a first order autoregressive process [B5]. It is a weighted sum of the distance from point q_j to all points $(q_1, q_2, \ldots, q_{i-1})$ previously arranged.

This statement becomes clear by expanding Eq. B.2 through a backward substitution of Eq. B.2 itself. i.e.

$$\begin{aligned} u_j(i) &= \alpha u_j(i-1)+d_{q_{i-1}q_j} \\ &= \alpha[\alpha u_j(i-2)+d_{q_{i-2}q_j}]+d_{q_{i-1}q_j} \\ &= \alpha^{i-2}d_{q_1q_j}+\alpha^{i-3}d_{q_2q_j}+\ldots+d_{q_{i-1}q_j} \end{aligned} \qquad (B.9)$$

where the index i refers to the fact that the i-th position of the sequence is to be determined and index j denotes a candidate point for the i-th position in sequence Q. According to Eq. B.3, we observe that the i-th point in sequence Q is chosen to be j such that the value of $u_j(i)$ is a minimum. This means that j is the point closest to the previously arranged points $(q_1,\ldots,q_{i-1})$, where closeness is measured by the weighted distance given by Eq. B.9. After all points have been rearranged, the final sequence Q has the property that for each point q_i the value

$$u_{q_i} = u_{q_i}(i) = \alpha^{i-2}d_{q_1q_i}+\ldots+d_{q_{i-1}q_i} \qquad (B.10)$$

is a minimum with respect to all q_j, with j=i, i+1,...,n.

Before explaining how to determine the value of α (which is discussed in next section), the reason why minimizing u(i) will lead to the "condensed" binary matrix (a band of one's along the diagonal) will be explained in the following.

Referring back to Eq. B.10, the quantity αu_{q_i}, can be written as following

$$\alpha u_{q_i} = \alpha^{i-1} d_{q_1 q_i} + \ldots + d_{q_{i-1} q_i}$$

$$= \sum_{k=1}^{i-1} \alpha^{i-k} d_{q_k q_i}$$

Let $d'_{ij} = d_{q_i q_j}$ define $u'_i = u_{q_i}$

$$u'_i = \sum_{k=1}^{i-1} \alpha^{i-k} d'_{ki} \qquad \text{(B.11)}$$

With $D'=[d'_{ij}]$, also $d'_{ii}=0$, it is clear the u'_i in Eq. B.7 is a weighted sum of elements of the i-th column of D' from the topmost d'_{1i} to the diagonal element d'_{ii}. From Eq. B.11 the weighting constants for elements of D' are illustrated as following

$$W = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{m-1} \\ & 1 & \alpha & \cdots & \alpha^{m-2} \\ & & 1 & \cdots & \alpha^{m-3} \\ & & & \ddots & \vdots \\ & & & & 1 \end{bmatrix}$$

With $0 \leq \alpha \leq 1$, from the above expression we observe that the weight decreases as the element of D' is further away from the diagonal. Thus minimizing Eq. B.11 will lead to the desired matrix D' in which the larger elements are placed further away from the diagonal than those smaller elements,

because the weights are smaller away from the diagonal.

B.3 Correlation Property of the Series u(i) (=u_{q_i}(i))

The correlation coefficient of u(i) and u(i-j) is defined as following:

$$\gamma_j = \frac{\mathrm{Cov}[u(i),u(i-j)]}{V[u(i)]^{1/2}V[u(i-j)]^{1/2}}$$

$$= \frac{E[(u(i)-\overline{u(i)})(u(i-j)-\overline{u(i-j)}]}{\{E[(u(i)-\overline{u(i)})^2]E[(u(i-j)-\overline{u(i-j)})^2]\}^{1/2}}$$

$$i > 0, \quad i-j > 0 \qquad \text{(B.12)}$$

Assuming the number of points is large, and the distances are uniformly distributed, we may approximately model the series u(i) as an asymptotically weak stationary process

i.e. $E[u(i)]=U$ for large values of i

and $E[u(i)\,u(j)] = S(|i-j|)$ (B.13)

Substituting these expressions into Eq. B.8. We have

$$\gamma_{|i-j|} = \frac{S(|i-j|)-U^2}{S(0)-U^2} \qquad \text{(B.14)}$$

Let $k=|i-j|$, the above equation can be written in the following form

$$\gamma_k = \frac{S(k)-U^2}{S(o)-U^2} \qquad \text{(B.15)}$$

The three unknown quantities U, S(K), S(0) will be obtained by the following derivation:

Let $E[d_{ij}]=d$ (B.16a)

$V[d_{ij}]=\sigma^2$ (B.16b)

Then

$$U = E[u(i)]$$

$$= E[\sum_{\ell=1}^{i-1} \alpha^{\ell-1} d'_{i-\ell,i}]$$

$$= \sum_{\ell=1}^{i-1} \alpha^{\ell-1} E[d'_{i-\ell,i}]$$

$$= \sum_{\ell=1}^{i-1} \alpha^{\ell-1} d$$

$$= \frac{1-\alpha^i}{1-\alpha} d$$

$$U \cong d/(1-\alpha) \qquad \text{(B.17)}$$

for sufficient large i and $0<\alpha<1$

To obtain the value of S(0), we may square both sides of Eq. B.2 and take the expectation; then

$$S(0) = \alpha^2 S(0) + 2\alpha U d + \alpha^2 + d^2$$

The above expression is arrived at by substituting Eq. B.16 into the term of $E[d'^2_{i-1,i}]$ and using the assumption that u(i-1) is independent of the distance $d_{q_{i-1}q_i} = d'_{i-1,i}$. Rearranging the terms, and substituting the expression in Eq. B.17 for U, we have

$$(1-\alpha^2)S(0) = \frac{2\alpha d^2}{1-\alpha} + \alpha^2 + d^2$$

$$\text{So} \qquad S(0) = \frac{d^2}{(1-\alpha)^2} + \frac{\sigma^2}{1-\alpha^2} \qquad \text{(B.18)}$$

Finally for the value of S(K), we have

$$S(K) = E[u(i)u(i-k)]$$

$$= E[(\alpha u(i-1) + d_{i-1,i}) \cdot u(i-K)]$$

$$= \alpha S(K-1) + \frac{d^2}{1-\alpha}$$

By an iterative back substitution, we get

$$S(K) = \alpha^K S(0) + \frac{d^2}{1-\alpha} \sum_{\ell=1}^{K} \alpha^{\ell-1}$$

$$= \alpha^K S(0) + \frac{d^2}{1-\alpha} \frac{1-\alpha^K}{1-\alpha}$$

$$= \frac{\alpha^K d^2}{(1-\alpha)^2} + \frac{\alpha^K \sigma^2}{1-\alpha^2} + \frac{d^2(1-\alpha^K)}{(1-\alpha)^2}$$

$$= \frac{\alpha^K \sigma^2}{1-\alpha^2} + \frac{d^2}{(1-\alpha)^2} \qquad \text{(B.19)}$$

Substitute B.17, B.18 and B.19 into B.15

$$K = \frac{\frac{\alpha^K \sigma^2}{1-\alpha^2} + \frac{d^2}{(1-\alpha)^2} - \frac{d^2}{(1-\alpha)^2}}{\frac{d^2}{(1-\alpha)^2} + \frac{\sigma^2}{1-\alpha^2} - \frac{d^2}{(1-\alpha)^2}}$$

i.e. $\gamma_K = \alpha^K$ (B.20)

This expression is arrived at by the assumption as previously stated that the number of points is large and distances are uniformly distributed, such that the series u(i) in Eq. B.10 can be approximately described by an asymptotic weekly stationary process. But if this assumption is not valid, that is, clusters exist in that set of points, we shall expect that the true value of γ_k will not be a monotonically decreasing function of K as Eq. B.20 shows. This is because for a proper value of α, points belonging to same cluster will gather as neighbors in the sequence Q. When the point q_i and its previous neighbors (with index less than 1) are in the same cluster, u(i) will be a relative small value;

conversely when q_i and its previous neighbors are not in the same cluster, u(i) will be relatively large. That is if several clusters exist in the sequence and points belonging to the same cluster are neighbors, we might expect a periodic change in the values of u(i), and this periodic change of u(i) will result in the nonstationarity of both S(K) and the correlation coefficient γ_K.

From this discussion, it is clear that we might expect the value γ_K to drop below a certain significance level when two points with indexes which differ by K do not belong to the same cluster. If we model this decreasing of γ_K by Eq. B.20, then the value of α is determined in the following manner.

Assume T_n is the threshold value, such that two points with distance less than T_n will be considered as belonging to the same cluster. The probability that an arbitrary pair of distinct points are in the same cluster is given by

$$p = P(d_{ij}<T \mid i\neq j) = \frac{1}{n(n-1)}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}\right] \qquad (B.21)$$

$$\text{where} \quad b_{ij} = \begin{cases} 1 & \text{if } d_{ij} < T_n \\ 0 & \text{otherwise} \end{cases} \qquad (B.2)$$

and n is the total number of points

The averaged number of points a point is associated to (with distance less than threshold) is

$$m = n \times p \qquad (B.22)$$

Because of the symmetric property, for a point in the

sequence the expected number of associated points on either side of that point will be half the value of n in Eq. B.22. Set N equal to that number i.e.

$$N = \frac{m}{2} = \frac{n \times p}{2} \qquad (B.23)$$

With the previous discussion, we know the correlation coefficient $\gamma_K=\alpha^K$ should decrease to some insignificant level as K approaches the number N. This is because as previously discussed N is considered to be the expected limit that two points belong to the same cluster. As a consequence, the value of α can be determined by empirically setting the value for insignificance as 0.1. This gives

$$\alpha^N = 0.1$$

$$\text{i.e.} \quad \alpha = (0.1)^{1/N} \qquad (B.24)$$

with N being determined from Eq. B.

As the threshold constant T_n appeared in B.22, two approaches can be used to determine its value. One is subjective and another is objective, depending upon the purpose of the clustering.

For the objective approach, the threshold can be determined from the historgram of the distance distribution, because if clusters exists, the distance distribution will be multimodal.

For the subjective approach, the threshold is determined such that it is equal to the maximum distance that two points are mutually associated. The maximum distance is usually defined as a desired property of the result clusters.

Reference

[B1] R. E. Bonner, "On Some Clustering Techniques," IBM Journal, January, 1964.

[B2] J. G. Auguston and J. Minker, "An Analysis of Some Graph Theoretical Cluster Techniques," J/ACM, Vol. 17, No. 4, pp. 571-588, October, 1970.

[B3] C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. Comp., Vol. C-20, pp. 68-86, January, 1971.

[B4] Z. Chen and K. S. Fu, "On Graph Theory and Cluster Analysis," Allerton Conference on Circuit and System Theory, Monticello, Ill., October, 1972.

[B5] G. E. Box and G. M. Jenkins, Time Series Analysis, Forecast and Control, p. 9, Holden-Day, San Francisco, 1970.

APPENDIX C

METHODS OF APPROXIMATING THE CLASSIFICATION PROBABILITIES

Transformed Divergence D_T (Eq. 5.1a) and transformed Bhattacharyya Distance B_T (Eq. 5.1b) are used as seperability criteria to cluster the classes into groups. The theoretical aspects of these types of transform have been discussed in Ref. 53 and 54. Empirical methods are used to approximate the classification probabilities from those distances, for the reasons mentioned in Section 2.1.2 that there is no exact method to predict these probabilities. Experimental results which relate the probability of correct classification to the seperability measures D_T and B_T are also reported [53] [54]. Some of these results are shown here in Fig. C.1(a) and C.1(b). They are superimposed classification results of 2790 and 40,000 data sets respectively. For each data set 2000 samples are classified, and the estimated probability of correct classification is then plotted against the seperability measure. Also shown in Fig. C.1(a) are the least-squares polynomial approximation (of degree 3), and the theoretically derived bound [46] on performance as function of separability.

Clearly, there is no one-to-one relationship between probability of correct classification and the measure of seperability in both figures. But for the range of

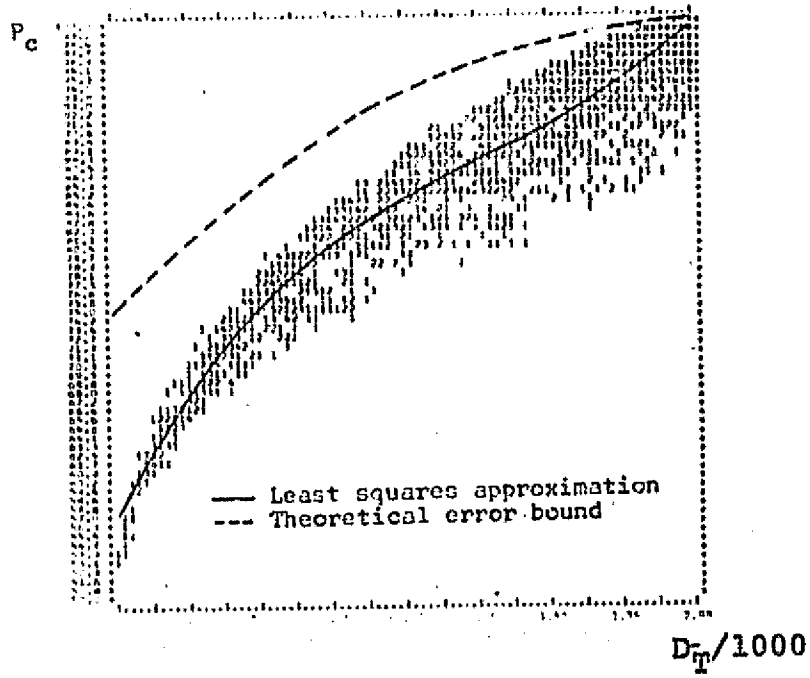


Figure C.1a Error Rate versus Transformed Divergence D_T.

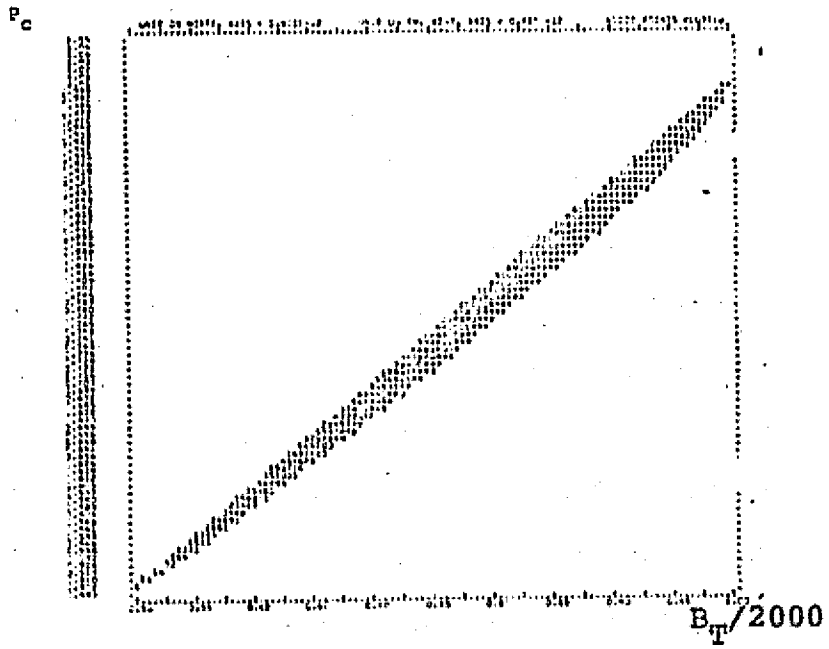


Figure C.1b Error Rate versus Transformed Bhattacharyya Distance B_T.

classification accuracy likely to be encountered in real problems--say, 80 percent to 100 percent--the mean of performance has an approximate linear relationship to the seperability measures. Hence, approximations based on this observation are made to predict the misclassification rate, i.e.

$$\varepsilon_{12} = K_1(1 - D_T/2000) \qquad \text{(C.1a)}$$

or

$$\varepsilon_{12} = K_2(1 - B_T/2000) \qquad \text{(C.1b)}$$

where constants K_1, K_2 are adjusted to be 0.32 and 0.5 respectively, for the seperability measures in the range from 1000 to 2000.

These approximations are valid for two-class classification. For more than two classes, the misclassification rate is approximated by:

$$\varepsilon = \gamma \sum_{i=1}^{n} \sum_{j=1}^{n} p_i \, \varepsilon_{ij} \qquad \text{(C.2)}$$

where ε_{ij} is given by Eq. C.1, and $\varepsilon_{ii} = 0$.

p_i is the probability that a sample is from class ω_i

γ is a constant

The factor γ ($0 \leq \gamma \leq 1$) is included since the summation of pairwise errors (of two class classification) is always greater than or equal to the true error for multiclass classification [33]. It has been observed that in order for ε to be close to its true value, γ should decrease as n (the number of classes) increases. For this reason, γ is

approximated by:

$$\gamma = (\frac{2}{n})^{\beta} \qquad \beta > 0 \tag{C.3}$$

This form is chosen such that γ decreases as n increases, and γ is one in case with two classes. When the number of classes is about 10, β is experimentally set as 0.7. For other values of n the same value of β is used for a rough approximation.

In the case there are n classes in a nonterminal node which has only $m(\geq n)$ immediate descendant nodes, an (n×m) associativity matrix $A = [a_{ij}]$ can be formed according to

$$a_{ij} = \begin{cases} 1 & \text{if class } \omega_i \text{ belongs to the j-th immediate descendant node} \\ 0 & \text{otherwise} \end{cases} \tag{C.4}$$

The probability Q_{ij} of a point from class ω_i being classified into the γ-th immediate descendant node is approximated in the following manner:

$$Q_{ij} = \begin{cases} e_{ij} & \text{if } a_{ij} = 0 \qquad \text{(C.5a)} \\ p_i - e_i & \text{if there is only one j such that } a_{ij}=1 \qquad \text{(C.5b)} \\ \dfrac{\bar{d}_i}{\bar{d}_{ij}+\bar{d}_{i\ell}}(p_i - e_i) & \text{if } a_{ij}=1 \text{ and } a_{i\ell}=1 \qquad \text{(C.5c)} \end{cases}$$

with $$e_{ij} = \sum_{k=1}^{n} \gamma p_i \varepsilon_{ik} (1-a_{kj}) \tag{C.6}$$

$$e_i = \sum_{j=1}^{m} e_{ij} \tag{C.7}$$

and $\bar{d}_{i\ell}$ is the mean of the separabilities from class ω_i to the "core" classes (Appendix B) of $C\ell$

By using the clustering procedure explained in Appendix B, a class can belong to one or at most two clusters. Thus Eq. C.5 approximates all the possible values of Q_{ij}. Notice that only the probability of correct classification is approximated in Eq. C.5b. The probability P_i of a sample from class ω_i in the j-th immediate descendant node d_j (cluster C_j) then is given the value of Q_{ij} for the a priori probability of a further stage. The probability P_j appeared in Eq. 4.6, that a classification path will pass through a particular node d_j is

$$P_j = \sum_{i=1}^{n} a_{ij} \, Q_{ij} \qquad (C.8)$$

And error rate $\varepsilon(d_k)$ (d_k denotes the nonterminal node under consideration, i.e. the immediate ascendant node which generates m descendant nodes as previously mentioned) is now given by:

$$\varepsilon(d_k) = \sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij} \qquad (C.9)$$

APPENDIX D

DESCRIPTION OF DATA SETS FOR EXPERIMENTS

D.1 Training and Test Fields for Experiment 5.1

TRAINING FIELDS

CLASS RC							
66000652	8BH	357	399	8	61	97	8
66000652	12DP	521	573	8	173	217	8
66000652	12BP	561	581	8	29	105	8
66000652	13D RC	613	635	8	121	193	8
CLASS CORN1							
66000652	11BCN1	489	525	8	65	107	8
66000652	8A*CON1	361	399	8	5	43	8
66000652	CORN2	261	287	8	37	69	8
66000652	C	309	345	8	1	37	8
66000652	C	701	753	8	205	217	8
CLASS OATS1							
66000652	OATS1	417	457	8	37	89	8
66000652	O	365	377	8	133	193	8
66000652	O	581	593	8	125	201	8
CLASS SOYB1							
66000652	SB	61	89	8	41	97	8
66000652	SYB3	125	149	8	41	83	8
66000652	SYB3	225	275	8	109	185	8
66000652	SYB1	645	699	8	53	85	8
66000652	SB	709	785	8	41	61	8
66000652	SYB3	761	785	8	121	197	8
CLASS WHEAT							
66000652	WHT	285	319	8	109	193	8
66000652	WHT2	585	693	8	205	213	8
66000652	W	345	357	8	129	205	8
66000652	W	497	513	8	165	217	8

AREA CLASSIFIED

66000652	1	850	4	1	220	4

D.1, cont.

TEST FIELDS

TEST 1							
66000652	8BH	357	399	1	61	97	1
66000652	12DP	521	573	1	173	217	1
66000652	12BP	561	581	1	29	105	1
66000652	13D RC	613	635	1	121	193	1
66000652	5A	221	261	1	1	35	1
66000652	9D	433	447	1	125	199	1
66000652	13BP	593	635	1	53	91	1
66000652	R	705	725	1	125	197	1
TEST 2							
66000652	11BCN1	489	525	1	65	107	1
66000652	8A*CON1	361	399	1	5	43	1
66000652	CORN2	261	287	1	37	69	1
66000652	C	309	345	1	1	37	1
66000652	C	701	753	1	205	217	1
66000652	CORN2	401	419	1	119	187	1
66000652	CORN2	161	211	1	29	79	1
66000652	C	469	481	1	13	101	1
66000652	CORN2	589	641	1	5	41	1
TEST 3							
66000652	OATS1	417	457	1	37	89	1
66000652	O	365	377	1	133	193	1
66000652	O	581	593	1	125	201	1
66000652	OATS1	329	339	1	129	193	1
66000652	OATS2	537	553	1	25	107	1
66000652	O	597	609	1	125	201	1
TEST 4							
66000652	SB	61	89	1	41	97	1
66000652	SYB3	125	149	1	41	83	1
66000652	SYB3	225	275	1	109	185	1
66000652	SYB1	645	699	1	53	85	1
66000652	SB	709	785	1	41	61	1
66000652	SYB3	761	785	1	121	197	1
66000652	SB	65	89	1	117	157	1
66000652	SYB3	293	341	1	43	97	1
66000652	SYB3	489	515	1	117	161	1
66000652	SYB1	645	667	1	125	193	1
66000652	SYB2	709	781	1	69	105	1
TEST 5							
66000652	WHT	285	319	1	109	193	1
66000652	WHT2	585	693	1	205	213	1
66000652	W	345	357	1	129	205	1
66000652	W	497	513	1	165	217	1
66000652	WHT	349	397	1	109	123	1
66000652	W	457	493	1	165	217	1
66000652	WHT2	649	701	1	1	45	1

D.2 Details of Experiment 5.2

D.2.1 Field Descriptions

TRAINING FIELDS

CLASS C							
66000652	11BCN1	489	525	8	65	107	8
66000652	8A*CON1	361	399	8	5	43	8
66000652	CORN2	261	287	8	37	69	8
66000652	C	309	345	8	1	37	8
66000652	C	701	753	8	205	217	8
CLASS S							
66000652	SB	61	89	8	41	83	8
66000652	SYB3	125	149	8	41	83	8
66000652	SYB3	235	265	8	129	165	8
66000652	SYB1	645	670	8	53	85	8
66000652	SB	709	785	8	41	61	8

AREA CLASSIFIED

66000652		1	850	4	1	220	4

TEST FIELDS

TEST 1							
66000652	CORN2	161	211	1	29	79	1
66000652	CORN	221	255	1	39	55	1
66000652	CORN2	261	287	1	37	69	1
66000652	C	309	345	1	1	37	1
66000652	8A*CON1	361	399	1	5	43	1
66000652	CORN2	401	419	1	119	187	1
66000652	C	469	481	1	13	101	1
66000652	11BCN1	489	525	1	65	107	1
66000652	CORN2	589	641	1	5	41	1
66000652	C	701	753	1	205	217	1
TEST 2							
66000652	SB	61	89	1	41	83	1
66000652	SB	65	89	1	117	157	1
66000652	SYB3	125	149	1	41	83	1
66000652	SYB3	235	265	1	129	165	1
66000652	SYB3	293	341	1	43	97	1
66000652	SYB3	489	515	1	117	161	1
66000652	SYB1	645	667	1	125	193	1
66000652	SYB1	645	670	1	53	85	1
66000652	SYB2	709	781	1	79	95	1
66000652	SB	709	785	1	41	61	1

CLASS C

CHANNEL	1	2	3	4	5	6	7	8	9	10	11	12
SPECTRAL BAND	0.40 - 0.44	0.44 - 0.46	0.46 - 0.48	0.48 - 0.50	0.50 - 0.52	0.52 - 0.55	0.55 - 0.58	0.58 - 0.62	0.62 - 0.66	0.66 - 0.72	0.72 - 0.80	0.80 - 1.00
MEAN	84.89	79.71	60.77	61.68	85.57	88.14	63.46	82.31	67.96	78.26	98.95	78.10
STD. DEV.	5.53	5.31	3.28	3.37	7.02	5.96	3.94	8.10	7.58	8.29	9.37	6.07

CORRELATION MATRIX

SPECTRAL BAND	0.40 - 0.44	0.44 - 0.46	0.46 - 0.48	0.48 - 0.50	0.50 - 0.52	0.52 - 0.55	0.55 - 0.58	0.58 - 0.62	0.62 - 0.66	0.66 - 0.72	0.72 - 0.80	0.80 - 1.00
0.40-0.44	1.00											
0.44-0.46	0.90	1.00										
0.46-0.48	0.85	0.87	1.00									
0.48-0.50	0.82	0.87	0.88	1.00.								
0.50-0.52	0.80	0.88	0.92	0.90	1.00							
0.52-0.55	0.68	0.80	0.83	0.86	0.93	1.00						
0.55-0.58	0.62	0.75	0.82	0.87	0.91	0.96	1.00					
0.58-0.62	0.55	0.7(	0.80	0.84	0.89	0.88	0.92	1.00				
0.62-0.66	0.48	0.64	0.75	0.82	0.83	0.84	0.90	0.97	1.00			
0.66-0.72	0.41	0.57	0.70	0.73	0.81	0.84	0.90.	0.96	0.95	1.00		
0.72-0.80	0.02	0.02	0.00	-0.08	0.07	0.22	0.15	-0.08	-0.17	-0.01	1.00	
0.80-1.00	-0.01	0.04	-0.01	-0.08	0.07	0.21	0.14	-0.09	-0.15	.02	0.82	1.00

D.2.2 Statistics of Training Sets

CLASS S

CHANNEL	1	2	3	4	5	6	7	8	9	10	11	12
SPECTRAL BAND	0.40 - 0.44	0.44 - 0.46	0.46 - 0.48	0.48 - 0.50	0.50 - 0.52	0.52 - 0.55	0.55 - 0.58	0.58 - 0.62	0.62 - 0.66	0.66 - 0.72	0.72 - 0.80	0.80 - 1.00
MEAN	87.56	82.57	62.74	64.21	89.86	91.31	66.50	88.14	73.36	85.53	96.29	75.58
STD. DEV.	5.05	4.92	3.40	3.24	6.58	5.35	3.31	5.93	4.73	5.50	6.97	4.12

CORRELATION MATRIX

SPECTRAL BAND	0.40 - 0.44	0.44 - 0.46	0.46 - 0.48	0.48 - 0.50	0.50 - 0.52	0.52 - 0.55	0.55 - 0.58	0.58 - 0.62	0.62 - 0.66	0.66 - 0.72	0.72 - 0.80	0.80 - 1.00
0.40-0.44	1.00											
0.44-0.46	0.90	1.00										
0.46-0.48	0.93	0.90	1.00									
0.48-0.50	0.90	0.89	0.92	1.00								
0.50-0.52	0.92	0.91	0.93	0.93	1.00							
0.52-0.55	0.88	0.89	0.89	0.93	0.94	1.00						
0.55-0.58	0.83	0.86	0.88	0.91	0.92	0.93	1.00					
0.58-0.62	0.88	0.89	0.92	0.93	0.94	0.92	0.93	1.00				
0.62-0.66	0.83	0.85	0.86	0.92	0.90	0.89	0.92	0.95	1.00			
0.66-0.72	0.79	0.82	0.83	0.90	0.90	0.90	0.91	0.92	0.93	1.00		
0.72-0.80	0.37	0.40	0.41	0.42	0.47	0.55	0.49	0.39	0.35	0.46	1.00	
0.80-1.00	0.12	0.16	0.19	0.20	0.26	0.30	0.30	0.17	0.13	0.31	0.77	1.00

D.2.2, cont.

D.2.3 Classification Results and Estimated Error Bounds

FEATURE SUBSETS	MEASURED ERROR RATE (%)	UPPER BOUND ON ERROR RATE (%)
10	21.8	42.0
4, 10	18.8	37.2
4, 10, 12	17.9	33.2
4, 9, 10, 12	17.8	28.9
1, 4, 9, 10, 12	18.5	27.7
1, 4, 8, 9, 10, 12	17.7	26.4
1, 4, 8, 9, 10, 11, 12	18.4	25.3
1, 4, 7, 8, 9, 10, 11, 12	19.5	23.9
1, 4, 6, 7, 8, 9, 10, 11, 12	20.0	22.9
1, 4, 5, 6, 7, 8, 9, 10, 11, 12	20.3	22.0
1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	21.1	21.0
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	20.6	20.4

D.3 Training and Test Fields for Experiment 5.7

TRAINING FIELDS

CLASS SOYBEANS								
66000600 25-6	65	81	4	69	89	4	3	SOYBEANS1
66000600 31-13	237	253	4	141	167	4	3	SOYBEANS2
66000600 36-7	307	327	4	59	81	4	3	SOYBEANS3
66000600 7-23	773	777	4	135	179	4	3	SOYBEANS4
CLASS CORN								
66000600 36-4	167	177	4	33	77	4	1	CORN1
66000600 36-9	267	283	4	45	61	4	1	CORN2
66000600 36-8	319	341	4	21	31	4	1	CORN3
66000600 12-9	603	625	4	13	33	4	1	CORN4
CLASS OATS								
66000600 6-2	365	373	4	145	185	4	3	OATS1
66000600 1-11	421	455	4	63	83	4	3	OATS2
66000600 7-1	591	599	4	135	181	4	3	OATS3
CLASS WHEAT I								
66000600 31-12	295	303	4	134	175	4	4	WHEAT1
66000600 6-14	471	495	4	177	201	4	4	WHEAT2
66000600 7-2	607	665	4	203	211	4	4	WHEAT3
CLASS RED CLVR								
66000600 6-10	439	447	4	139	183	4	6	RED CL1
66000600 1-1	539	565	4	175	195	4	6	RED CL2
66000600	599	619	4	69	95	4	6	RED CL3
CLASS ALFALFA								
66000600 7-24	731	737	4	129	177	4	6	ALFALFA1
66000600 7-24	749	755	4	131	171	4	6	ALFALFA2
66000600 7-22	809	817	4	155	183	4	6	ALFALFA3
CLASS RYE								
66000600 6-8	527	569	4	127	155	4	7	RYE1
CLASS OR SOIL								
66000600 36-1	97	115	4	49	85	4	5	&R OO -
CLASS WHEAT II								
66000600 12-10	655	695	4	17	41	4	9	W&E&T4

AREA CLASSIFIED

66000600	1	850	4	1	220	4	

D.3, cont.

TEST FIELDS

```
GROUP SOYBEANS(1/1/),CORN(2/2/),OATS(3/3/),WHEAT(4/4,9/),RED CLVR(5/5/),
GROUP ALFALFA(6/6/),RYE(7/7/),BR SOIL(8/8/)

TEST 1
66000600  25-6      57   89   1    47   103  1    SOYBEANS
66000600  30-4      63   79   1    115  169  1    SOYBN COVERS W
66000600  31-1      93   101  1    113  183  1    SOYBN COVERS W
66000600  36-2      123  133  1    43   101  1    SOYBEANS
66000600  36-2      133  149  1    43   83   1    SOYBEANS
66000600  31-13     217  273  1    109  201  1    SOYBEANS
66000600  12-3      705  797  1    69   111  1    SOYBN E PRT PR
66000600  36-7      291  341  1    43   97   1    SOYBN VOLUNTR
66000600  6-9       489  519  1    115  161  1    SOYBEANS
66000600  7-27      643  663  1    125  197  1    SOYBEANS
66000600  12-7      647  699  1    51   87   1    SOYBEANS
66000600  12-2      647  675  1    93   111  1    SOYBEANS
66000600  12-3      705  797  1    33   63   1    SOYBNW. PRT P
66000600  7-23      759  785  1    121  197  1    SOYBN PLT CIRC
TEST 2
66000600  36-4      157  187  1    17   101  1    CORN
66000600  36-4      189  215  1    17   79   1    CORN
66000600  36-10     221  255  1    39   55   1    CORN
66000600  36-9      261  287  1    39   65   1    CORN
66000600  36-8      307  349  1    19   35   1    CORN
66000600  6-11      401  421  1    111  199  1    CORN
66000600  12-9      589  643  1    3    43   1    CORN DIFF VARI
TEST 3
66000600  31-11     327  335  1    109  197  1    OATS
66000600  6-2       365  377  1    131  183  1    OATS DITCH W E
66000600  1-11      413  467  1    45   93   1    OATS
66000600  7-1       583  605  1    121  193  1    OATS
TEST 4
66000600  31-12     285  317  1    109  199  1    WHEAT
66000600  6-1       347  353  1    107  205  1    WHEAT
66000600  6-1       385  393  1    109  203  1    WHEAT
66000600  6-14      459  509  1    167  211  1    WHT 2 VARIETIE
66000600  7-2       581  589  1    203  211  1    WHEAT
66000600  12-10     649  699  1    3    43   1    WHEAT 2 VAR LO
TEST 5
66000600  31-23     129  133  1    113  199  1    RD CL DIVRT SO
66000600  1-1       357  399  1    61   95   1    RED CL HAY
66000600  6-10      433  453  1    113  197  1    RED CL HAY
66000600  6-7       521  561  1    173  215  1    RED CL PASTURE
66000600  1-6       559  581  1    49   109  1    RED CL PASTURE
66000600  12-8      589  633  1    49   109  1    RED CL PASTURE
66000600  7-29      613  619  1    121  183  1    RD CL DIVERTED
66000600  7-28      629  637  1    123  191  1    RED CL HAY
66000600            675  695  1    127  195  1    RED CL
TEST 6
66000600  7-24      729  737  1    121  195  1    ALFALFA HAY
66000600  7-24      745  757  1    121  195  1    ALFALFA HAY
66000600  7-22      793  815  1    121  195  1    ALFA. HAY GRAS
TEST 7
66000600  6-8       525  577  1    119  163  1    RYE
TEST 8
66000600  36-3      137  149  1    87   101  1    BARE SOIL
66000600  36-1      95   117  1    45   89   1    BARE SOIL
```