AgRISTARS

SR-P1-04194
NAS9-15466

A Joint Program for Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing

December 1981

Supporting Research

Technical Report

Multistage Classification of Multispectral Earth Observational Data: The Design Approach

by M. J. Muasher and D. A. Landgrebe

Purdue University
Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907

SR-P1-04194
NAS9-15466
LARS 101481

MULTISTAGE CLASSIFICATION OF MULTISPECTRAL EARTH OBSERVATIONAL DATA: THE DESIGN APPROACH

M.J. Muasher and D.A. Landgrebe

Purdue University
Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907-0501

December 1981

Star Information Form

1. Report No.	2. Government Accession No	3. Recipient's Catalog No
SR-P1-04194		

4. Title and Subtitle	5. Report Date
Multistage Classification of Multispectral Earth Observational Data: The Design Approach	December 1981
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No
M.J. Muasher and D.A. Landgrebe	101481

9. Performing Organization Name and Address	10. Work Unit No.
Purdue University Laboratory for Applications of Remote Sensing 1220 Potter Drive West Lafayette, IN 47906-1399	11. Contract or Grant No. NAS9-15466
	13. Type of Report and Period Covered Technical
12. Sponsoring Agency Name and Address NASA Johnson Space Center Remote Sensing Research Division Houston, TX 77058	14. Sponsoring Agency Code

15. Supplementary Notes

F.G. Hall, Technical Monitor

M.E. Bauer, Principal Investigator

16. Abstract

One of the main problems in a multistage decision tree procedure is predicting the optimal features to be used at every node. An algorithm is proposed which predicts the optimal features at every node in a binary tree procedure. The algorithm estimates the probability of error by approximating the area under the likelihood ratio function for two classes and taking into account the number of training samples used in estimating each of these two classes.

Some results on feature selection techniques, particularly in the presence of a very limited set of training samples, are presented. Results comparing probabilities of error predicted by the proposed algorithm as a function of dimensionality as compared to experimental observations are shown for aircraft and Landsat data. Results are obtained for both real and simulated data.

Finally, two binary tree examples which use the algorithm are presented to illustrate the usefulness of the procedure.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Binary tree, feature selection, Hughes Phenomenon, K-L expansion, likelihood function, number of training samples	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

TABLE OF CONTENTS

	Page
LIST OF TABLES	iii
LIST OF FIGURES	iv
ABSTRACT	vii
CHAPTER 1 - INTRODUCTION	1
1.1 Multistage Classification	1
1.2 Review of Literature	6
1.2.1 Training Procedure	6
1.2.2 Performance Estimators	7
1.2.3 Multistage Classifiers	12
1.3 Summary of Contents	19
CHAPTER 2 - PARAMETER CONSIDERATIONS FOR A MULTISTAGE BINARY TREE CLASSIFIER	20
2.1 The Hughes Phenomenon	20
2.2 Simultaneous Diagonalization: Theory	27
2.3 Feature Selection	29
2.4 Simulation Algorithm	32
2.4.1 Need For A Simulation Algorithm	32
2.4.2 Statistical Background	35
CHAPTER 3 - PERFORMANCE ESTIMATOR: APPROXIMATION TO THE PROBABILITY OF ERROR	37
3.1 The Likelihood Function	37
3.2 Performance Estimator	42
3.2.1 The Normal Assumption	43
3.2.2 The Modified Gamma Assumption: Fukunaga and Krile Version	44
3.2.3 Proposed, Modified Algorithm	56

Page

CHAPTER 4 - EXPERIMENTAL RESULTS 64

4.1 Introduction 64
4.2 Experiments on Feature Selection 65
4.3 Experiments on the Hughes Phenomenon 74
4.4 Experiments Comparing Algorithm and Experimental Results 84
4.5 Experiments on a Binary Tree Classification Procedure 104

CHAPTER 5 - SUMMARY AND CONCLUSIONS 114

5.1 Summary of Results 114
5.2 Suggestions for Further Research 115

LIST OF REFERENCES 117

APPENDICES

Appendix A Generation of Normally Distributed Samples 124
Appendix B On The Probability Density Functions of $\hat{\sigma}_1^2$ And $\hat{\sigma}_2^2$ 127
Appendix C Derivation of the Variance of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ 134
Appendix D Classification Results Tables . . . 141
Appendix E Computer Program Listings 150
Appendix F Description of Data Sets For Experiments 161

VITA 172

LIST OF TABLES

Table		Page
D.1	Classification Results of Aircraft, Simulated Data, Using 20 samples per class.	142
D.2	Classification Results of Aircraft, Simulated Data, Using 13 samples per class.	143
D.3	Classification Results of Aircraft, Real Data, Using 20 samples per class.	144
D.4	Classification Results of Aircraft, Real Data, Using 13 samples per class.	145
D.5	Classification Results of Landsat, Multitemporal, Simulated Data, Using 20 samples per class. . .	146
D.6	Classification Results of Landsat, Multitemporal, Simulated Data, Using 13 samples per class. . .	147
D.7	Classification Results of Landsat, Multitemporal, Real Data, Using 20 samples per class.	148
D.8	Classification Results of Landsat, Multitemporal, Real Data, Using 13 samples per class.	149

LIST OF FIGURES

Figure		Page
1.1	An Example of a "Single-Stage" Algorithm In Classifying Multispectral Data.	2
1.2	An Example of a "Multi-Stage" Algorithm In Classifying Multispectral Data.	5
2.1	The Hughes Phenomenon	21
2.2	Explanation of the Hughes Phenomenon	22
2.3	Delineation of Optimal Subspace by Simultaneous Diagonalization.	30
3.1	Probability Density Functions of $h(X/w_i)$ and The Probability of Error.	41
3.2	A Flowchart of Fukunaga and Krile's Algorithm. .	57
4.1	Classification Results of Data in Experiment 4.1 Using Three Feature Selection Techniques	68
4.2	Classification Results of Data in Experiment 4.2 Using Three Feature Selection Techniques. . . .	70
4.3	Classification Results of Data in Experiment 4.3 Using Three Feature Selection Techniques. . . .	72
4.4	Experimental Classification Results of Aircraft, Simulated Data Using Different Numbers of Training Samples.	76
4.5	Experimental Classification Results of Aircraft, Real Data Using Different Numbers of Training Samples.	79
4.6	Experimental Classification Results of Landsat, Multitemporal, Simulated Data Using Different Numbers of Training Samples.	81

Figure Page

4.7 Experimental Classification Results of Landsat, Multitemporal, Real Data Using Different Numbers of Training Samples. 82

4.8 Classification Results of Fukunaga and Krile's Example Reproduced. 86

4.9 A Flowchart of the Modified Algorithm 88

4.10 Classification Results of Aircraft, Simulated Data, Using 20 Samples per Class. 89

4.11 Classification Results of Aircraft, Simulated, Data, Using]3 Samples per Class. 92

4.12 Classification Results of Aircraft, Real Data, Using 20 Samples per Class. 93

4.13 Classification Results of Aircraft, Real Data, Using 13 Samples per Class. 95

4.14 Classification Results of Landsat, Multitemporal, Simulated Data, Using 20 Samples per Class. . . 97

4.15 Classification Results of Landsat, Multitemporal, Simulated Data, Using 13 Samples per Class. . . 99

4.16 Classification Results of Landsat, Multitemporal, Real Data, Using 20 Samples per Class. 100

4.17 Classification Results of Landsat, Multitemporal, Real Data, Using 13 Samples per Class. 102

4.18 Binary Tree Design Structure of Landsat, Multitemporal, Real Data, Using 13 Training Samples per Class, With Numbers Inside Nodes Indicating Number of Training Samples Used. . . 106

4.19 Single Stage and Binary Tree Classification Results of Landsat, Multitemporal, Real Data, Using 13 Samples per Class. 108

Figure Page

4.20 Binary Tree Design Structure of Aircraft, Real Data, Using 13 Samples per Class. 110

4.21 Single-Stage and Binary Tree Classification Results of Aircraft, Real Data, Using 13 Training Samples per Class. 111

ABSTRACT

One of the main problems in a multistage decision tree procedure is predicting the optimal features to be used at every node. An algorithm is proposed which predicts the optimal features at every node in a binary tree procedure. The algorithm estimates the probability of error by approximating the area under the likelihood ratio function for two classes, and taking into account the number of training samples used in estimating each of these two classes. Some results on feature selection techniques, particularly in the presence of a very limited set of training samples are presented. Results comparing probabilities of error predicted by the proposed algorithm as a function of dimensionality as compared to experimental observations are shown for aircraft and Landsat data. Results are obtained for both real and simulated data. Finally, two binary tree examples which use the algorithm are presented to illustrate the usefulness of the procedure.

CHAPTER 1
INTRODUCTION

1.1 Multistage Classification

A number of different types of classifiers are now in routine use in remote sensing. Most of these classification algorithms, using pattern recognition techniques, can be regarded as "single-stage" classifiers, where an "unknown" pattern is tested against all classes using one feature subset, and then the pattern is assigned to one of the present classes in a single-stage decision procedure. An example of such a procedure is shown in Figure 1.1.

In recent years, as classification of multispectral data has found a larger number of users and a wider range of applications, the need has been felt for alternate, more powerful techniques than the conventional classifiers, through the use of which more information could be extracted more accurately and/or efficiently from the scene. Some of the reasons that have warranted this need include:

1. The need to extract more detailed information from data. The opportunity to do so results from the

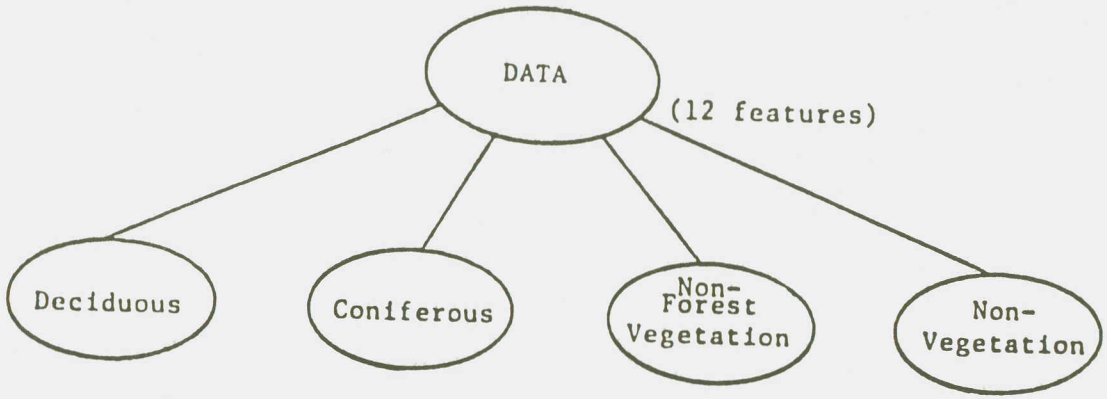


Figure 1.1 An Example of a "Single-Stage" Algorithm In Classifying Multispectral Data.

emergence of more complex data sets. The growing use of multitype data bases containing Landsat data with a variety of other quantitative geodata together with the anticipated launching of more sophisticated sensors such as the Thematic Mapper result in the opportunity to extract considerably more information from the data.

2. The broadening of the range of applications. As pattern recognition methods have developed, they have found a larger number of users with a wider range of applications. The feedback from these different and versatile uses has indicated problems and needs not initially present.

3. The ever present need for improved classification accuracy. There are some applications for which conventional classifiers have proved to be marginal at best. Some of these are listed in Swain et al. (1) and include multi-image analysis and the use of mixed feature types.

4. The need for improved processing efficiency. The conventional, single-stage, classifiers use only one particular feature subset and are somewhat inefficient, as they must compare an unknown pattern against all possible classes before assigning that pattern to a particular class.

Because of these and other factors, there has been some research in recent years directed towards developing multistage classifiers, whereby the decision procedures go through several stages before finally assigning a pattern to a class. An example of such a procedure is shown in Figure 1.2.

The purpose of this research is to develop a layered decision algorithm that can increase the accuracy and efficiency over the conventional single-stage classification approach. Developing such an algorithm requires, among other things, a careful look at some parameters that are crucial to any successful attempt at tackling such a complex problem. In particular, three areas have to be investigated:

1. The development of an adequate training procedure to define an initial set of spectral classes with their respective statistics;

2. The investigation of various error estimators and the development of an adequate performance estimator that can reasonably predict the accuracy or any trends in performance;

3. The development of an algorithm to build a binary tree making use of the above-mentioned methods.

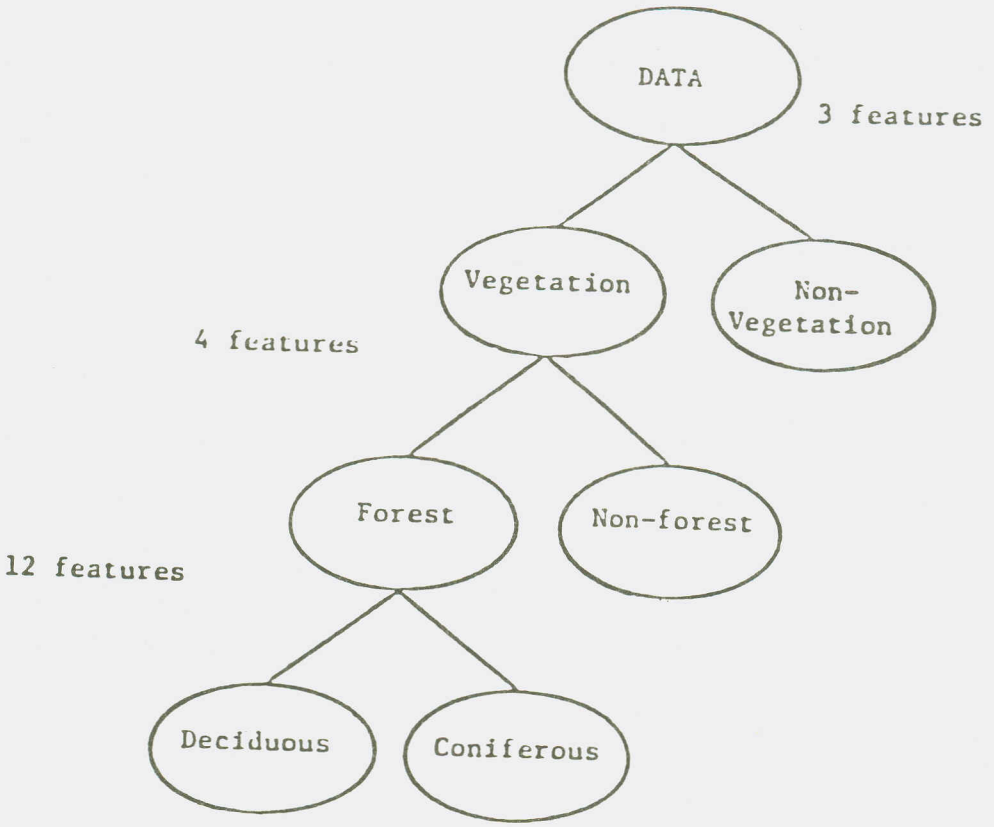


Figure 1.2 An Example of a "Multi-Stage" Algorithm In Classifying Multispectral Data.

Of these three areas, the most important problem is believed to be the development of an accurate error estimator, especially in the presence of what has come to be known as the Hughes phenomenon (elaborated upon later in the review of literature). Predicting the conditions under which the Hughes phenomenon occurs provides the key to the solution of the problem. Therefore, a considerable portion of the research has been directed towards trying to understand and predict the impact of this phenomenon.

1.2 Review of Literature

1.2.1 Training Procedure

Several training methods have been suggested in the literature. We will not attempt to list all of them, but rather will give a background of some of the methods reviewed and used in this work.

The training process is the procedure whereby labeled samples are selected and used to compute class statistics which in turn are used to classify unlabeled (i.e., "unknown") samples.

Several parameter estimation methods (training methods) have appeared in the literature. Sample-partitioning methods, the leaving-one-out method, clustering are but a few. See, for example, Fukunaga (2) and Duda and Hart (3).

For remote sensing purposes, clustering has been widely used in developing training statistics. Two basic approaches have been: a supervised clustering approach, in which the analyst selects areas of known cover types , each set of areas belonging to one cover type is clustered separately, and then the statistics for these areas are then obtained with the aid of a computer; and the non-supervised clustering approach, in which the entire training area is subdivided into clusters by the clustering algorithm and each cluster is then identified by the analyst and given a specific label. The statistics of each cluster corresponding to a cover type or a subclass of a cover type are then calculated. Fleming et al. (4,5) investigated several clustering approaches and their effect on classification accuracy. Among the approaches they used were non-supervised clustering, supervised clustering, modified clustering, mono- (aggregate) cluster blocks, and multi- (class-conditional) cluster blocks.

1.2.2 Performance Estimators

A key factor in the design of a layered decision algorithm is the ability to predict how the algorithm will perform in terms of accuracy at every node. While optimizing the performance at every node does not necessarily produce a globally optimal tree, it is still a very important and useful step in the design.

Several performance (or error) estimators have appeared in the literature. Again, we will not attempt here to exhaustively list all the contributions made, but rather will give an idea of how the research in this area has progressed.

Performance estimators can be divided into two main categories:

Performance functions which have some sort of direct relationship with the probability of error. Examples are Parzen estimators (see (2)), the k-nearest neighbor error estimator (see (6)). More recently, Mobasseri et al. (7) published an error estimator that computes the minimum probability of error through use of a combined analytical and numerical integration over a sequence of simplifying transformations of the feature space. The results have been shown to be similar to those obtained by conventional techniques. However, the algorithm becomes computationally too inefficient to use as the number of classes and/or features increases. Moore, Whitsitt and Landgrebe (8) (see also Whitsitt and Landgrebe (9)) developed a stratified posterior estimator which, like Mobasseri's, depends only on a given set of statistics. This was later used by Wiersma (10) and both estimators (Mobasseri's and Whitsitt's) were compared in (11) and found to give similar results, with Whitsitt's algorithm being faster in some cases. The former procedure

uses a "deterministic" grid to sample the feature space, while the latter uses an internally generated random data base and assigns the feature vector to the appropriate class via the maximum a posteriori principle. Both procedures assume normal class conditional statistics.

Separability measures, most of which have only a subtle, indirect, and often unknown, relationship to the probability of error. Various separability measures have been in common use in remote sensing applications. Among these are: Divergence (12), Transformed Divergence (13), Jeffreys-Matusita distance (14,15), Bhattacharyya distance (16) and the Mahalanobis distance (17). (See list in (24).)

Several works have been reported comparing different separability measures and their effects on performance. (See (9,13,18,19,62).)

There are two problems with most of the above separability measures applied to remote sensing applications: (1) ambiguity and (2) linearity in pairwise error. The term ambiguity implies here that there does not exist a one-to-one relationship between the value of the measure and the probability of error. Linearity means that equal incremental changes in the measure imply equal changes in the probability of error, over the whole range. Whitsitt (9) developed a distance measure $D_{erf} = erf(\sqrt{2B})$ where B is the Bhattacharyya distance and $erf(\cdot)$ is the gaussian error

function. He found that the resulting measure is less ambiguous and more linear than the measure B.

Another key factor in the process of error estimation is the choice of feature subsets. The problems here are twofold:

1. As the number of features becomes large, it becomes desirable to choose a subset of these features that can adequately predict the accuracy. This selection process also can become expensive if one must search through all possible combinations of the feature set. It is desirable, therefore, to have a priori knowledge of the importance of each feature in relation to the probability of error. The Karhunen-Loeve expansion (attributed to Karhunen (20), and Loeve (21)) in pattern recognition literature has historically been used as a feature selection technique. It has the advantage of producing uncorrelated features (in theory, but the features are actually approximately uncorrelated in a practical K-L transformation). In addition, it imposes an ordering on the features in terms of importance in a representation error sense. As a result, first feature is "likely" to be more important than the second in calculating the probability of error, and so on. More recently, Oja and

Karhunen (22,23) published two papers on the construction of K-L expansions for pattern recognition purposes that do not require the computation of any covariance matrices.

2. The probability of error is not necessarily monotonically decreasing as the number of features increases. This is due to a peculiar phenomenon that has come to be known as the Hughes phenomenon. Hughes (25) found that with a fixed and finite training pattern sample, recognition accuracy can first increase as the number of measurements on a pattern increases, but decay with measurement complexity higher than some optimum value. He also reported that for unlimited training data, this does not occur and the recognition accuracy reaches an optimum only at infinite measurement dimensionality. According to Hughes, if insufficient sample data are available to estimate the pattern probabilities accurately, then a Bayes recognizer is not necessarily optimal. Many papers have since been published on this phenomenon, confirming it or trying to explain why it occurs (see (26-42)). Thus, it appears that a successful design should predict when and if such phenomena occur.

1.2.3 Multistage Classifiers

In recent years, some work has appeared in the literature aimed at developing multistage classification algorithms. There is much yet to be learned about such algorithms, and no work has been reported claiming optimality (or even close to optimality) of results.

In general, earlier work can be grouped into two main categories:

Sequential classification methods. These can be found in several papers and books (see, for example, (43-45)). Basically, the method consists of observations made on feature measurements, one at a time. After an observation is made, the classifier either reaches a final decision and the process is terminated, or it makes another observation until a final decision is reached.

Hierarchical classification methods. These are subdivided into two categories:

1. Hierarchical clustering methods. Examples of such work are found in Fukunaga (2), Dubes and Jain (46), who present a semi-tutorial review of the state of the art in cluster validity, and Lukasova (47). In general, hierarchical clustering is designed to generate a classification tree. The "root" node of the tree represents a collection of samples (either a training data set or the entire sample

set) and each terminal node represents either an individual sample or a group of samples belonging to some class within the set of classes in the data set. The method attempts to divide the set of samples in each node into disjoint subsets which form new nodes. Defined as such, the method is often nonparametric and depends heavily on the ability of the algorithm to find meaningful divisions of samples that correspond at terminal nodes with meaningful classes.

2. Decision trees and criterion functions. Most of the work done in multistage algorithms belongs to this category. Often, a decision tree is built using an optimization or criterion function that dictates the structure of the tree. It is this kind of approach that will be of greatest concern in this research.

Hierarchical methods differ from sequential methods in certain important respects. While in sequential schemes any class can be accepted at any stage of the measurement process, in hierarchical schemes certain classes are excluded from consideration at each stage. Also, sequential methods impose a linear ordering on the features. In hierarchical methods, features used along one decision path can be different from those used along another path.

In 1971, Nadler (48) tried to calculate error rates in a hierarchical decision structure under assumptions of statistical independence among the members of the hierarchy.

Even under such assumptions, the results assume "small" probabilities of errors at any level.

Several heuristic methods of constructing tree designs have been proposed in the literature. Some studies were done using optimization methods to automate the classifier design procedure, but the assumptions made were often too restrictive. Meisel and Michalopoulos (49) in 1973 presented a two-stage partitioning algorithm for the design of an optimal binary tree. In the first stage, a suboptimal sufficient partition is obtained. The second stage optimizes the result of the first stage through a dynamic programming approach. The method allows only for linear discriminant functions to partition the space, certainly a suboptimal and too restrictive condition.

In 1974, Wu et al. (50) reported on a decision tree approach with direct application to multispectral data analysis. Several design procedures were proposed (one of which is manual), with special emphasis on a heuristic, machine-implemented approach. The optimality criterion used is a weighted sum of computation cost and accuracy. Results were presented which showed superiority in efficiency (but infrequently in accuracy) over the conventional classifier. The criterion function used, as it cannot predict beforehand the structure of the tree below that node, assumes all the nodes below the node under consideration are terminal nodes,

and hence is necessarily suboptimal. Later papers have appeared that have pointed to applications using this particular classifier (51,52).

In 1976, You and Fu (53) presented a linear binary tree classifier that uses linear discriminant functions at decision stages with an application to multispectral remotely sensed data. The procedure includes a grouping algorithm, a separability measure, and an error minimization procedure using the Fletcher-Powell algorithm (54). Again, the procedure is certainly suboptimal because of the assumption of linearity. Results reported, though, show that this classifier is much faster and more accurate than the maximum likelihood classifier with the same number of features. This is due to the fact that the procedure uses different feature subsets (with a restriction on their number) at each node, compared with only one feature subset used in the one-stage maximum likelihood classifier.

Kulkarni and Kanal (55) used dynamic programming and branch-and-bound methodologies in the design of hierarchical classifiers. The criterion of optimality they used is a weighted sum of the probability of error and the average measurement cost incurred in classifying a random sample. The design assumes that the features used at the nodes are statistically independent and that the decision at each node is a function of only that particular feature observation,

the design using only one best feature at each tree node. Further, the design of the optimal tree assumes a very low error rate for the tree, a very restrictive assumption since in many cases a high error rate is specifically the reason why a layered classifier was selected, i.e., to improve the accuracy. Although the authors presented some methods to reduce the complexity of their design algorithms, the examples they used involve only a small number of classes and features.

In 1977, Parkih (56) compared several classification techniques of clouds, including hierarchical design. However, his paper offers no new insights or major results that would help improve the state of the art.

Also in 1977, Sethi and Chatterjee (57) developed an algorithm for the design of an efficient decision tree with application to pattern recognition problems involving discrete variables. A criterion function was defined to estimate the minimum expected cost of a tree in terms of the weights of its terminal nodes and costs of the measurements, which then was used to establish the search procedure for the efficient decision tree. The concept of prime events was used to obtain the number of nodes and the corresponding weights in the design sample. No optimality claim was made, but the procedure was found to lead to the optimal tree in most of the cases. The procedure uses only one feature at

every node, and its applicability to remotely sensed multispectral data is very doubtful.

In 1978, Breiman (58) presented a procedure for building a binary classification tree. He used a criterion function that is only a function of the parent node and the two descendent nodes. He used one best feature at every node. He also reported on another regression algorithm developed at Survey Research Center, University of Michigan (59), in which the criterion function tries to reduce the variances of the two descendent nodes as much as possible from the variance of the parent node.

Rounds (60) in 1979 developed a binary decision tree algorithm, but again one feature is selected at every node. The approach is a nonparametric one, based on the Kolmogorov-Smirnov criterion.

Dattatreya and Sarma (61) in 1981 presented a multistage binary tree "minimum-cost" classifier, when general cost functions are associated with the tasks of feature measurements. The optimization of the binary tree is carried out using dynamic programming. However, one feature is only selected at every node.

In summary, most of the work done with multistage classifiers often imposed too restrictive assumptions or conditions, such as using one feature only at each node, or hav-

ing a linear discriminant function. Moreover, very few results have been reported on situations where the Hughes phenomenon occurs, namely, working with a limited set of training samples.

The major contributions of this research are then:

1. The development of some theroretical results that clearly show the dependence of the accuracy of the estimated statistics of the classes under consideration on the number of training samples used to estimate the statistics of those classes, as well as on the number of features used.

2. The development of an error estimator which is particularly useful when the number of training samples is limited, and which is suited for a binary tree classification procedure. This estimator, which allows the selection of a "near optimal" feature subset at every node, has no restrictions on the number of features that can be used at any node.

3. The incorporation of the above error estimator in a binary tree procedure, showing the usefulness of such a procedure in predicting the optimal features that lead to the best accuracy that can be attained given a fixed set of training samples.

1.3 Summary of Contents

In chapter 2, some parameter considerations for a multistage binary tree classifier are addressed in detail. The Hughes phenomenon is elaborated upon, and a technique known as "sumultaneous diagonalization" is introduced. Feature selection techniques are also treated. A data simulation algorithm that is repeatedly used in the research is also treated.

In chapter 3, an approximation algorithm to the probability of error is proposed that takes into account the Hughes phenomenon.

Chapter 4 presents experimental results on real and simulated data.

Finally, chapter 5 summarizes conclusions about the study. Some analytical details, together with computer listings and training data are placed in appendices.

CHAPTER 2
PARAMETER CONSIDERATIONS
FOR
A MULTISTAGE BINARY TREE CLASSIFIER

2.1 The Hughes Phenomenon

One of the major needs for a decision tree classifier originates from a dimensionality problem often referred to as the Hughes Phenomenon (25). A considerable portion of this research is directed towards understanding the Hughes phenomenon. Figure 2.1 illustrates the phenomenon conceptually. In the presence of a limited training sample size, the mean recognition accuracy as a function of the measurement complexity (number of features for our purposes) exhibits a peaking effect. Contrary to intuition, the mean accuracy does not always increase with additional measurements. Further, peaking of the curve shifts up and to the right as the number of samples increases, disappearing in the case of an infinite number of training samples (complete knowledge of the underlying distributions).

Figure 2.2 suggests a concept for one possible explanation of this phenomenon. Figure 2.2a shows a hypothetical

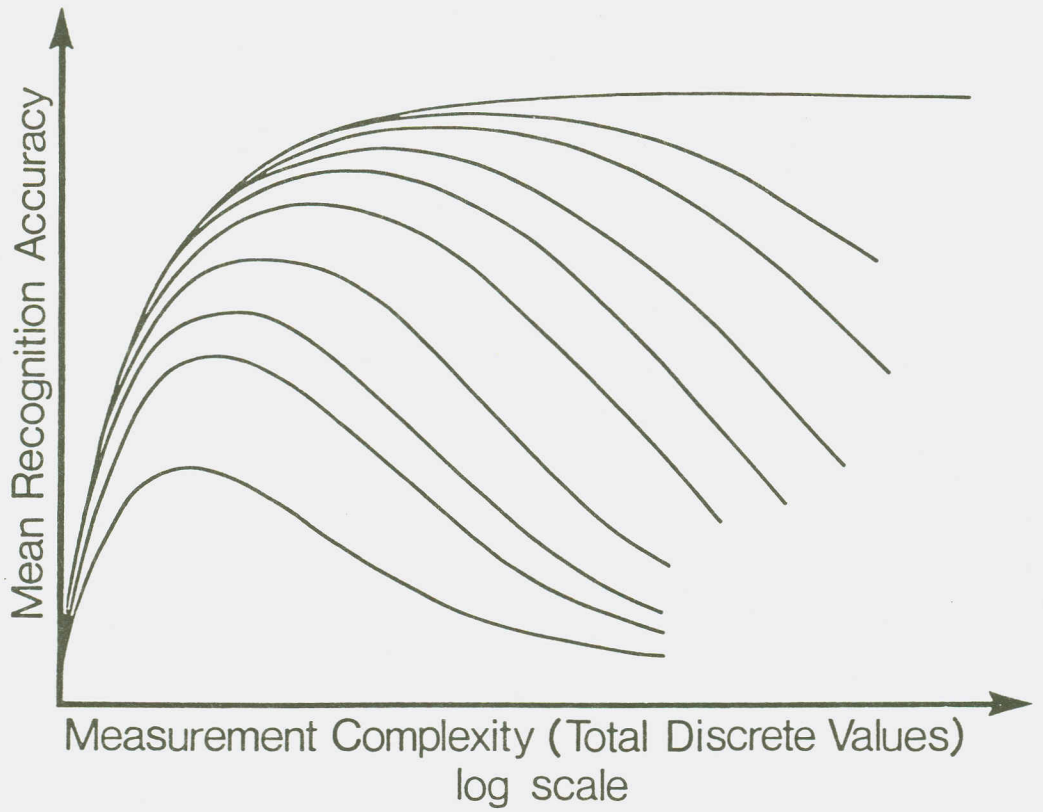
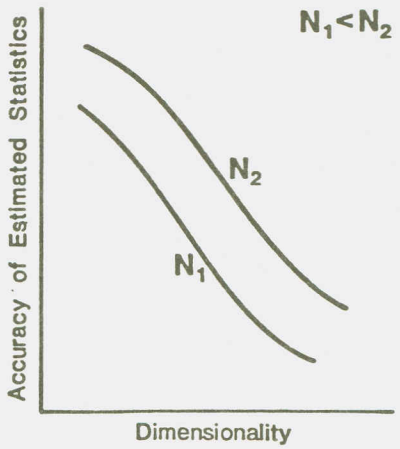
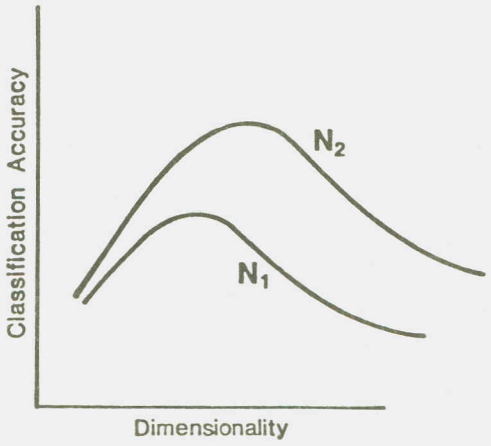


Figure 2.1 The Hughes Phenomenon.

2.2a



2.2b



2.2c

Figure 2.2 Explanation of the Hughes Phenomenon

graph of class separability plotted vs. dimensionality. As dimensionality increases, so does class separability (a non-decreasing function of dimensionality) until it saturates, and any further increase in dimensionality does not have a significant effect on class separability. But this is not the only effect on the mean accuracy. With the presence of a fixed, limited training sample size, any increase in dimensionality necessarily results on the average in a degradation in the accuracy of statistics estimation of the class distributions. Thus, conceptually, one should expect a curve similar to that of Figure 2.2b.. Further, as the number of samples increases, the curve should shift to the right, i.e., for any given dimensionality, the larger sample size should provide a better estimate of the true distributions. Assuming these two effects are the dominant effects on accuracy, adding the two effects results in Figure 2.2c, a curve similar to Figure 2.1. Based upon this concept of the phenomenon, the solution to the problem lies in being able to predict quantitatively how the number of samples present affects the accuracy of the estimated statistics . Especially in remote sensing applications of pattern recognition methods, training samples are limited as ground truth is often not present or difficult to get. Thus, the importance of the Hughes phenomenon becomes evident, as well as the validity of this conceptual explanation of it.

The Hughes phenomenon was studied by many researchers. (See (26-42)). Hughes (25), who was one of the earliest to introduce it and treat it in some detail, tried to explain it from a nonparametric point of view. The explanation given by Wacker and Landgrebe (62) is of another nonparametric case, where the Euclidean distance measure is used for discrimination among classes.

Several researchers (28-34) tried to study the effect of limited training sample size and independence of measurements on the recognition accuracy.

In 1979, Trunk (38) provided a simple example in which he showed theoretically that the probability of error approaches zero as the dimensionality increases and all the parameters are known in a two-class problem, but it approaches one-half as the dimensionality increases and the parameters are estimated.

In remote sensing applications, where maximum likelihood classifiers are frequently used, and where the assumption of class-conditional multivariate normally distributed data is invoked, not much work concerning the dimensionality problem has been reported yet. Wacker and El-Sheikh (40-42) presented some papers dealing with dimensionality problems for two-class Gaussian problems. Their results again show a Hughes phenomenon occuring with finite training data.

It then follows that any error estimator in a multistage classification algorithm that can claim some optimality in results from an accuracy point of view, should be able to predict when/if a peaking occurs in the curve mentioned earlier. It is this key problem that this research is attempting to solve, i.e. the development of an error estimator that can accurately predict the Hughes phenomenon.

Working with multispectral data, one almost always has to work with multiple feature measurements and multiple classes. In this research, we propose a binary tree multistage classifier. This means that any node in the tree is either a terminal node or is further subdivided into two nodes (with statistics corresponding to two classes).

The advantages of a binary tree procedure are the following:

1. Working with two classes allows a theoretical understanding of the problem. Many pattern recognition results that apply to two-class problems fail to do so in multi-class ones. This is particularly true in the "simultaneous diagonalization" technique that will be introduced shortly.

2. Most feature selection algorithms used in pattern recognition applications generally, and in remote

sensing applications specifically, are optimal only when applied to two-class problems. For multi-class problems, a separability criterion is averaged over pairs of classes and thus is optimal only in an average sense. Working with a binary tree, then, should provide us with both convenience and accuracy.

Working with multiple features, several properties are desired in these features which will make further analysis easier:

Uncoupled (Independent) Features. Uncoupling of features from one another simplifies analysis a great deal as it permits evaluating the effect of each feature separately from other features.

Ordered Features. If the features can be ordered, or at least approximately so, in terms of their effect on the probability of error, then the process of feature selection would be made easier.

Optimal Separability. The features should be optimal with respect to the probability of error for two distributions at hand. Putting it in different words, the feature subset should be tailored to the separability of the two distributions.

To this end, a technique known as a "simultaneous diagonalization" (63,64) is discussed in the next section.

2.2 Simultaneous Diagonalization: Theory

Let $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ be the estimated covariance matrices for classes 1 and 2, respectively. We seek a transformation matrix A such that

$$A\hat{\Sigma}_1 A = I \qquad A\hat{\Sigma}_2 A = \Lambda \tag{2.1}$$

where I is the identity matrix and Λ is a diagonal matrix.

This transformation would uncouple the features, while not affecting the probability of error because the latter is invariant under linear transformations. We proceed to find such a transformation as follows. (For more details, see (2), pp. 31-35.)

Let Θ and Φ be the eigenvalue and eigenvector matrices of $\hat{\Sigma}_1$, respectively; then

$$\Theta^{-\frac{1}{2}}\Phi^T \hat{\Sigma}_1 \Phi \Theta^{-\frac{1}{2}} = I \qquad (\Phi^T\hat{\Sigma}_1\Phi = \Theta) \tag{2.2}$$

$$\Theta^{-\frac{1}{2}}\Phi^T \hat{\Sigma}_2 \Phi \Theta^{-\frac{1}{2}} = K \qquad K \text{ is a general matrix} \tag{2.3}$$

Next, we desire to diagonalize K. To find eigenvalues of K, it is necessary to solve the equation

$$\left| K - \lambda I \right| = 0 \tag{2.4}$$

Replacing K and I in (2.4) by (2.2) and (2.3), we get

$$\left| \Theta^{-\frac{1}{2}} \Phi^T \hat{\Sigma}_2 \Phi \Theta^{-\frac{1}{2}} - \lambda \Theta^{-\frac{1}{2}} \Phi^T \hat{\Sigma}_1 \Phi \Theta^{-\frac{1}{2}} \right| = 0 \tag{2.5}$$

Or

$$\left| \Theta^{-\frac{1}{2}} \Phi^T \right| \left| \hat{\Sigma}_2 - \lambda \hat{\Sigma}_1 \right| \left| \Phi \Theta^{-\frac{1}{2}} \right| = 0 \tag{2.6}$$

Since $\Theta^{-\frac{1}{2}} \Phi^T$ is nonsingular, it follows that

$$\left| \hat{\Sigma}_2 - \lambda \hat{\Sigma}_1 \right| = 0 \tag{2.7}$$

or,

$$\left| \hat{\Sigma}_1^{-1} \hat{\Sigma}_2 - \lambda I \right| = 0 \tag{2.8}$$

So, only the eigenvalue and eigenvector matrices of $\hat{\Sigma}_1^{-1} \hat{\Sigma}_2$ need be calculated.

The eigenvalue matrix is then Λ , and the transpose of the eigenvector matrix, A^T, serves as the transformation matrix.

The idea behind simultaneous diagonalization is to transform the original features into a new space where the features are independent and then choose a subset of these features in the new space which is optimal with respect to the probability of error. This is illustrated in Figure 2.3.

2.3 Feature Selection

Before proceeding to discuss the approximation algorithms to estimate the probability of error, we digress briefly to discuss how the features are ordered.

The literature offers many studies made on comparing different separability measures and their effectiveness in choosing the best feature subset (see (9,13,18,62,65)). It appears that the Bhattacharyya distance is one of the most suitable separability measures for distinguishing between classes. Thus, it will be used as a basis for feature selection. The fact that the features are independent allows us to determine the effect of each feature on the probability of error separately.

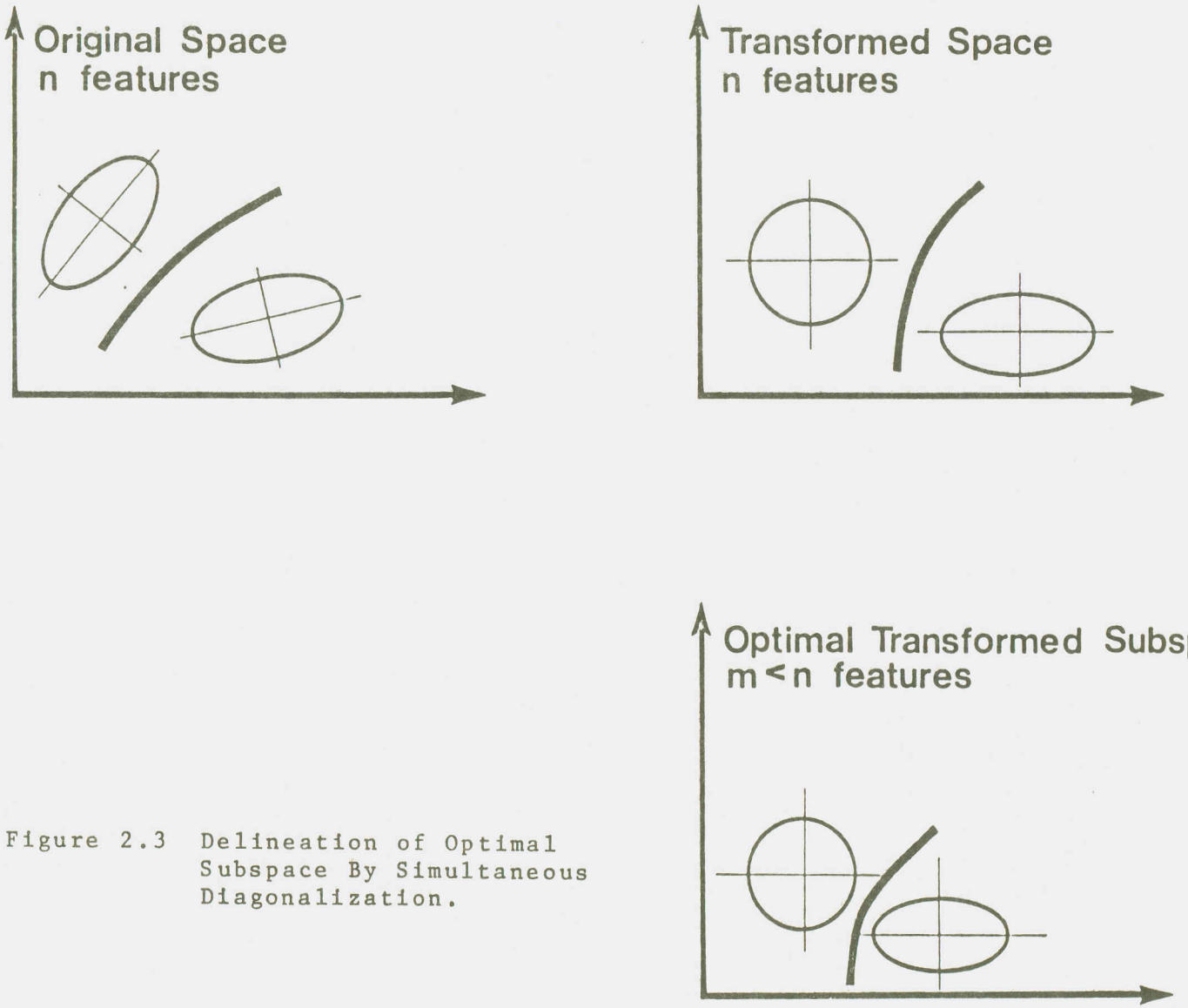


Figure 2.3 Delineation of Optimal Subspace By Simultaneous Diagonalization.

The Bhattacharyya distance for two normal distributions can be expressed as follows:

$$B = \frac{1}{8} (\hat{M}_1-\hat{M}_2)^T \left(\frac{\hat{\Sigma}_1 + \hat{\Sigma}_2}{2}\right)^{-1}(\hat{M}_1-\hat{M}_2) + \frac{1}{2} \ln \frac{\left| \frac{1}{2}(\hat{\Sigma}_1+\hat{\Sigma}_2)\right|}{\left|\hat{\Sigma}_1\right|^{\frac{1}{2}} \left|\hat{\Sigma}_2\right|^{\frac{1}{2}}} \tag{2.9}$$

After the simultaneous diagonalization transformation, however, B can be expressed as:

$$B = \sum_{i=1}^{p} \left[\frac{1}{4} \frac{(d_{1i}-d_{2i})^2}{\lambda_i + 1} + \frac{1}{2} \ln\left(\frac{1}{2}\left(\frac{1}{\lambda_i^{\frac{1}{2}}} + \lambda_i^{\frac{1}{2}} \right)\right)\right] \tag{2.10}$$

where d_{ij} is the jth element of the transformed class-conditional mean: $D_i = \hat{A}^T \hat{M}_i$; and λ_i is the ith diagonal element of Λ.

Thus, it is clear that for every feature i, B can be calculated separately. The feature with the largest B is the best feature, the one with the second largest is the second best, and so on. Also, the two best are the best two, and so on.

2.4 Simulation Algorithm

2.4.1 Need For A Simulation Algorithm

For remote sensing data analysis, several assumptions are commonly made. These assumptions are usually that the data are class-conditionally distributed multivariate normal and that the data used to train the classifier are representative of the area of interest. This second assumption actually has several parts. The assumption is made that in the process of training, all classes present in the scene are found, and all spectral subclasses of each class are also represented in the training data. Furthermore, the parameters of the distribution of each subclass are also assumed to be known from the training data. Each pixel is assumed to come from one of the training classes, and also is assumed to be entirely of one cover type.

In actual practice, these assumptions are not met. The number of spectral classes in the area is not known and clustering or some other method is used to determine the number of subclasses, in addition to estimating the statistics of those subclasses. Some of these methods also lead to non-normal subclasses. In particular, the clustering algorithm available through LARSYS truncates the tails of the subclass distributions and so leads to non-normal distributions.

There are also questions relating to a single picture element. A single pixel in Landsat data covers an area approximately 80 meters by 50 meters. More than one cover type may be present in this area and result in a "mixture pixel" observation. It is not clear how the distribution of the spectral response of mixture pixels can be related to the distribution of the spectral response of "pure pixels".

There has been much speculation in the remote sensing community as to the effect of the non-satisfaction of the basic assumptions. Whenever new algorithms are brought forth, the old question is raised again, indicating that there is insufficient understanding of the interaction of the real attributes of the data and the theory of the algorithms. At times it is not clear whether a particular result is due to aspects of the algorithm or to the extent the data set deviates from the assumptions.

In testing new algorithms, deviations from the assumptions may obscure the action of the new process. One way to clarify the situation is to apply the algorithm first to a data set satisfying the assumptions.

Such a data set could be obtained artificially, through simulation. The analyst could then know: how many classes exist in the data; the true distributions of the classes, including normality if desired; the observations could really be independent; and no pixel would be a "mixture

pixel". New algorithms could be studied on such a data set with the knowledge that any "strange" effects are indeed algorithm rather than data problems.

In many cases where simulated data have been used in the past, the data were too artificial, in the sense that all aspects of the image were controlled, removing the natural variation in object size, position, and relationship which occur in real data. This limited the use of the simulated data sets in testing new algorithms.

The natural spatial information occuring in multispectral data could be retained in a simulated image by spatially basing the simulation on a classification. It would be even better to base the simulated data on a digitized "ground truth" map if the spectral characteristics of the cover types were known. By basing the simulation on a classification, the number of classes, their exact distributions, and the class of each pixel in the area are known. If the classification was sufficiently accurate, then the spatial information held in the classification map will be close to the actual cover type map and actual spatial content of the original data. For each pixel in the area, a random vector distributed according to the pixel's class statistics could be generated. This becomes the simulated data vector.

This simulated method was reported in LARS Technical Report 070980 (66), and the program will be used for testing the error estimator developed.

2.4.2 Statistical Background

From the classification chosen as a basis for the simulation, the following are known: the number of classes K, the set of classes (ω_i, i=1,...K), the class distributions ($f(\omega_i)$, i=1,...K), their means and covariances (μ_i and Σ_i, i=1,...K), the number of channels p, and the class of every pixel in the scene.

From classical statistics:

(1) Let X:px1, A:pxp, and b:px1.

If $X \sim N(0, I_p)$, then $Y = AX + b \sim N(b, AI_pA^T = AA^T)$

(where I is the identity matrix having dimensionality p).

(2) Let Σ be a symmetric, positive definite matrix. Then there exists A, such that

$AA^T = \Sigma$ (A is denoted $\Sigma^{\frac{1}{2}}$)

To simulate a pixel which was a member of class i in the base classification, $N(0, I_p)$ (the random vector for each pixel is independent of other vectors) is generated. (See Appendix A.) Next $Y = \Sigma_i^{\frac{1}{2}}X + \mu_i$ is calculated; it is then a

random vector from the population $N(\mu_i, \Sigma_i)$. This process is repeated for each pixel of the base classification and the random vectors thus generated are stored appropriately, i.e., so as to correspond to their simulated spatial location.

The program requires as an input a classification map stored on a results tape. The results tape has the class statistics for p-dimensions also stored on it. The program then, uses the results map and the stored statistics to generate a p-dimensional data set, which is stored on a user specified output tape in LARSYS format.

Appendix A provides a mathematical derivation related to the generation of normally distributed samples. Appendix E provides a Fortran program listing for the simulation program.

With all the preliminaries discussed, we are now ready to begin our discussion of the error estimator algorithm.

CHAPTER 3

PERFORMANCE ESTIMATOR:

APPROXIMATION TO THE PROBABILITY OF ERROR

3.1 The Likelihood Function

As mentioned earlier, our goal is to develop a performance estimator that can predict where the peak in the Hughes curve occurs. Some of the most serious difficulties facing researchers in trying to estimate the probability of error in multidimensional analysis are:

1. The need to carry out a multiple integration on the multivariate probability density function. Most often, this integration is almost impossible to carry out analytically, and numerical integration that is often costly has to be perfomed.

2. The measurement features are often correlated, making it difficult to assess the importance of each feature separately on the probability of error.

3. In most of the cases, one has to deal with multiclass problems (greater than 2) which further complicates multivariate probability density functions.

It would be much easier, therefore, if one could work with a function that is one-dimensional but carries all the information present. Fortunately, since we are looking at two classes at a time in a binary tree procedure, such a function does exist, and is called the likelihood function (minus the log of the likelihood ratio). See, for example, (66).

The likelihood function, denoted h(X), is given by:

$$h(X) = -\ln p(X/w_1) \;/\; p(X/w_2) \qquad (3.1)$$

where

$p(X/w_i)$ is the probability density function of X given w_i.

In remote sensing applications, the assumption of multivariate class-conditional normal distributions is almost always invoked, and will be consistently used in this work.

Using this assumption, $p(X/w_i)$ becomes:

$$p(X/w_i) = \frac{1}{(2\pi)^{p/2} \left| \Sigma_i \right|^{\frac{1}{2}}} \exp\left(-\tfrac{1}{2}(X^T - M_i^T)\Sigma_i^{-1}(X - M_i)\right) \qquad (3.2)$$

where M_i is the mean vector of class i.

Σ_i is the covariance matrix of class i.

p is the number of dimensions.

In practice, M_i and Σ_i are estimated from training statistics and are replaced by $\hat{M}_i$ and $\hat{\Sigma}_i$.

The Bayes decision rule for minimum error may be written as follows:

$$P(w_1/X) \gtrless P(w_2/X) \rightarrow X \in \begin{cases} w_1 \\ w_2 \end{cases} \tag{3.3}$$

The a posteriori probabilities $P(w_i/X)$ may be calculated from the a priori probabilities $P(w_i)$ and the conditional density functions $p(X/w_i)$ using Bayes theorem, i.e.

$$P(w_i/X) = p(X/w_i)\ P(w_i) / p(X) \tag{3.4}$$

Since $p(X)$ is common to both sides of the inequality of (3.3), the decision rule can be expressed as:

$$p(X/w_1)\ P(w_1) \gtrless p(X/w_2)\ P(w_2) \rightarrow X \in \begin{cases} w_1 \\ w_2 \end{cases} \tag{3.5}$$

$$\ell(X) = \frac{p(X/w_1)}{p(X/w_2)} \gtrless \frac{P(w_2)}{P(w_1)} \rightarrow X \in \begin{cases} w_1 \\ w_2 \end{cases} \tag{3.6}$$

$h(X)$ can then be written as:

$$h(X) = -\ln(\ell(X)) = \tfrac{1}{2}(X-M_1)^T \Sigma_1^{-1} (X-M_1) - \tfrac{1}{2}(X-M_2)^T \Sigma_2^{-1} (X-M_2)$$

$$+\tfrac{1}{2}\ln \frac{|\Sigma_1|}{|\Sigma_2|} \gtrless \ln \frac{P(w_1)}{P(w_2)} \rightarrow X \in \begin{cases} w_2 \\ w_1 \end{cases} \qquad (3.7)$$

In practice, since M_i and Σ_i are replaced by $\hat{M}_i$ and $\hat{\Sigma}_i$, $h(X)$ becomes (after moving $\ln P(w_1)/P(w_2)$ to the L.H.S.):

$$\hat{h}(X) = \tfrac{1}{2}(X-\hat{M}_1)^T \hat{\Sigma}_1^{-1} (X-\hat{M}_1) - \tfrac{1}{2}(X-\hat{M}_2)^T \hat{\Sigma}_2^{-1} (X-\hat{M}_2)$$

$$+\tfrac{1}{2} \ln \frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_2|} - \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)} \gtrless 0 \rightarrow X \in \begin{cases} w_2 \\ w_1 \end{cases} \qquad (3.8)$$

The Bayes test for minimum error reduces then to looking at the value of $\hat{h}(X)$, assigning measurements with positive values to class 2, and measurements with negative values to class 1.

Note that $\hat{h}(X)$ is a one-dimensional random variable. The problem then is to know, or estimate, the probability density function of $\hat{h}(X)$. Once that is known, the probability of error can be obtained by carrying out a scalar integration. Figure 3.1 shows the probability density functions for $h(X)$ given either class 1 or 2.

The probability of error can be calculated as:

$$\varepsilon = p(\text{error}) = p(\text{error}/w_1)P(w_1) + p(\text{error}/w_2)P(w_2) \qquad (3.9)$$

Discriminant function:

$$\hat{h}(X) = -\ln \frac{\hat{p}(X|\omega_1)}{\hat{p}(X|\omega_2)}$$

$$= (X-\hat{M}_1)^T \hat{\Sigma}_1^{-1} (X-\hat{M}_1) - (X-\hat{M}_2)^T \hat{\Sigma}_2^{-1} (X-\hat{M}_2) + \ln \frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_2|} - \ln \frac{\hat{P}(\omega_1)}{\hat{P}(\omega_2)} \begin{matrix} \omega_1 \\ < \\ > \\ \omega_2 \end{matrix} 0$$

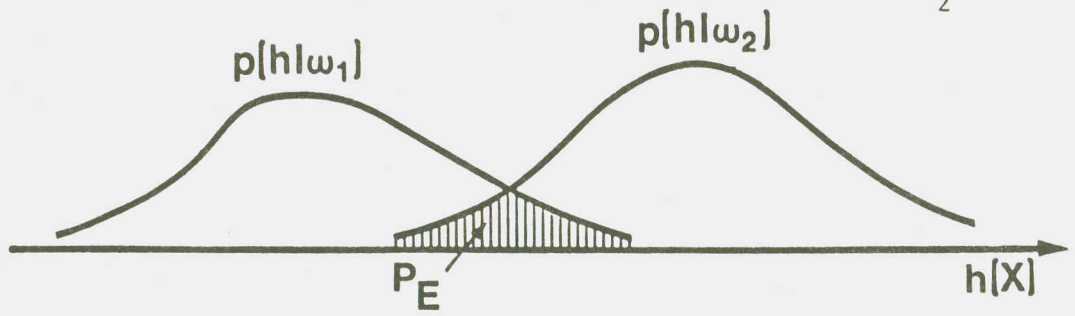


Figure 3.1 Probability Density Functions of $h(X/w_i)$ and The Probability of Error.

Let the domain or decision space of X be divided into regions Γ_1 and Γ_2. Then, if a sample belongs to w_1, an error occurs whenever $X \varepsilon \Gamma_2$. Similarly, if a sample belongs to w_2, an error occurs whenever $X \varepsilon \Gamma_1$. Thus,

$$\varepsilon = P(X \varepsilon \Gamma_2 / w_1)\ P(w_1) + P(X \varepsilon \Gamma_1 / w_2)\ P(w_2) \qquad (3.10)$$

In terms of the probability density functions of $\hat{h}(X/w_i)$, this becomes:

$$\varepsilon = P(w_1) \int_0^{\infty} p(h/w_1)\ dh + P(w_2) \int_0^{\infty} p(h/w_2)\ dh$$

$$= e_1 + e_2 \qquad (3.11)$$

The probability of error is then the area under the two curves in Figure 3.1 multiplied by the prior probabilities. The objective is to develop an algorithm which will approximate the class-conditional probability of $\hat{h}(X)$, and hence, the probability of error.

3.2 Performance Estimator

Fukunaga and Krile (64) developed an algorithm that approximates $\hat{h}(X)$. This algorithm assumes there are two-class multivariate normal distributions, and was tested using one eight-dimensional simulated data set.

The algorithm, however, assumes the training samples are enough to reasonably estimate the true statistics of the distributions, and hence does not take into account the Hughes phenomenon. Put in other words, in situations where the training samples are few and do not reflect the true statistics of the distributions, the algorithm will treat the statistics obtained from the training samples as a "perfect" estimation of some "wrong" distributions, when in fact they are an "imperfect" estimation of the true statistics.

It is this algorithm, proposed by Fukunaga and Krile, that we will use and modify to take into account the Hughes phenomenon. Therefore, it seems appropriate to explain the algorithm in detail, and then discuss the modifications made to it.

3.2.1 The Normal Assumption

Looking at equation (3.8), since $\hat{h}(X)$ is a quadratic function in general of a normal random variable X, it cannot itself in general be normally distributed. However, in the case where $\Sigma_1 = \Sigma_2$, $\hat{h}(X)$ becomes a linear function of X and hence is normally distributed.

In most cases, however, $\Sigma_1 \neq \Sigma_2$. Fukunaga and Krile still tried to assume that $\hat{h}(X)$ is normally distributed.

An algorithm was developed and tested in this research under the assumption that $\hat{h}(X)$ is normally distributed (although $\Sigma_1 \neq \Sigma_2$) but results showed it to be a very poor approximation of the probability of error and hence it was not further analyzed.

3.2.2 The Modified Gamma Distribution Assumption: Fukunaga and Krile Version

Consider $\hat{h}(X)$ as given by equation (3.8). Applying the simultaneous diagonalization technique described earlier, $\hat{\Sigma}_1$ is transformed to the identity matrix I, and $\hat{\Sigma}_2$ is transformed to a diagonal matrix Λ . The transformation matrix is denoted A^T, or the transpose of the eigenvector matrix A.

Without losing generality, we assign the origin of the coordinate system such that:

$$\hat{m}_1 = 0 \quad \text{and} \quad \hat{m}_2 = \hat{M}_1 - \hat{M}_2 \tag{3.12}$$

With $X \varepsilon w_1$, $\hat{h}(X)$ can be written as another function of Y, where $Y=A^T X$, as follows:

$$\hat{h}(Y/w_1) = Y^T Y - (Y-\hat{D})^T \hat{\Lambda}^{-1} (Y-\hat{D}) + \ln \left| \frac{\hat{\Sigma}_1}{\hat{\Sigma}_2} \right| - 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)} \tag{3.13}$$

where $\hat{D} = A^T \hat{m}_2$.

Since the features are now uncoupled, this can be written as:

$$\hat{h}(Y/w_1) = \sum_{i=1}^{p} \left(y_i^2 - \frac{1}{\hat{\lambda}_i}(y_i-\hat{d}_i)^2 - \ln \hat{\lambda}_i\right) - 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)}$$

$$= \sum_{i=1}^{p} \left(\left(1- \frac{1}{\hat{\lambda}_i}\right)\left(y_i + \frac{\hat{d}_i}{\hat{\lambda}_i-1}\right)^2 - \left(\frac{\hat{d}_i^2}{\hat{\lambda}_i-1} + \ln \hat{\lambda}_i\right)\right) \qquad (3.14)$$

$$- 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)}$$

where p is the number of dimensions.

$\hat{d}_i$ is the ith element of vector $\hat{D}$.

Now, we have $\hat{h}(Y/w_1)$ in terms of p independent Gaussian random variables y_i, each of which has zero mean and unit variance with respect to class w_1.

Defining a new transformed variable Z and a transformed difference- of-means vector $\hat{\nu}$ as follows:

$$Z = (\hat{\Lambda}^{-\frac{1}{2}} A^T)(X-\hat{m}_2) \qquad (3.15)$$

$$\hat{\nu} = (\hat{\Lambda}^{-\frac{1}{2}} A^T)\hat{m}_2 = \hat{\Lambda}^{-\frac{1}{2}} \hat{D} \qquad (3.16)$$

$\hat{h}(X/w_2)$ can be expressed as a function of the new variable Z and $\hat{\nu}$ by substituting (3.15) and (3.16) into (3.8) as follows:

$$\hat{h}(Z/w_2) = (Z+\hat{\nu})^T \hat{\Lambda} (Z+\hat{\nu}) - Z^T Z + \ln \frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_2|} - 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)} \tag{3.17}$$

Again, since the features are uncoupled, we can write $\hat{h}(Z/w_2)$ as follows:

$$\hat{h}(Z/w_2) = \sum_{i=1}^{p} (\hat{\lambda}_i (z_i+\hat{\nu}_i)^2 - z_i^2 - \ln \hat{\lambda}_i) - 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)}$$

$$= \sum_{i=1}^{p} ((\hat{\lambda}_i - 1) \left(z_i + \frac{\hat{\lambda}_i^{\frac{1}{2}} \hat{d}_i}{\hat{\lambda}_i - 1}\right)^2 - \frac{\hat{d}_i}{\hat{\lambda}_i - 1} + \ln \hat{\lambda}_i)) - 2 \ln \frac{\hat{P}(w_1)}{\hat{P}(w_2)} \tag{3.18}$$

Again, we have an expression in terms of p independent Gaussian variables z_i, each of which has zero mean and unit variance.

Next, we define the following quantities for convenience:

$$a_{1i} = 1 - 1/\hat{\lambda}_i \tag{3.19}$$

$$b_{1i} = \hat{d}_i/(\hat{\lambda}_i - 1) \tag{3.20}$$

$$a_{2i} = \hat{\lambda}_i - 1 \tag{3.21}$$

$$b_{2i} = \hat{\lambda}_i^{\frac{1}{2}} \hat{d}_i / (\hat{\lambda}_i - 1) \tag{3.22}$$

$$C = \sum_{i=1}^{p} (\ln \hat{\lambda}_i + \hat{d}_i/(\hat{\lambda}_i - 1) + 2 \ln \hat{P}(w_1)/\hat{P}(w_2) \tag{3.23}$$

Substituting equations (3.19)-(3.23) back into equations (3.14) and (3.18), we get:

$$\hat{h}(Y/w_1) = \sum_{i=1}^{p} (a_{1i} (y_i + b_{1i})^2) - C \tag{3.24}$$

$$\hat{h}(Z/w_2) = \sum_{i=1}^{p} (a_{2i} (z_i + b_{2i})^2) - C \tag{3.25}$$

Referring from now on to Y and Z as ξ, and to y_i and z_i as ξ_i, we find that $\hat{h}(\hat{\xi}/w_1)$ and $h(\hat{\xi}/w_2)$ have the same functional form, except for the values of a_{1i}, b_{1i}, a_{2i}, and b_{2i}.

Theorem 3.1

If $X = (x_1, \ldots, x_p)$ where the x_i are a sample from a Normal$(0,\sigma^2)$ population, then the random variable $V = \sum_{i=1}^{p} x_i^2 / \sigma^2$ has a χ_p^2, or chi-square, distribution.

Proof:

See (67), p. 16.

Theorem 3.2

If $s_1,\ldots,s_p$ are independent random variables, then the density of their sum $s_1+s_2+\ldots+s_p$ equals the convolution of their respective densities.

Proof

See (68), p. 189.

Examining equations (3.24) and (3.25), shows that the density functions of $\hat{h}(\xi/w_1)$ and $\hat{h}(\xi/w_2)$ can be obtained by convolving the densities of p non-central (because of the b_{1i} and the b_{2i} terms) χ^2 variables having multiplicative constants a_{1i} and a_{2i}, and adding a shift parameter C.

The density of $\hat{h}(\xi)$ is divided into three parts:

$$V_{kr} = \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki} (\xi_{ki} + b_{ki})^2 \quad \text{for } a_{ki} \geq 0 \tag{3.26}$$

$$V_{ks} = \sum_{a_{kj} < 0}^{p_{ks}} a_{kj} (\xi_{kj} + b_{kj})^2 \quad \text{for } a_{kj} < 0 \tag{3.27}$$

$$C = \sum_{i=1}^{p} (\ln \hat{\lambda}_i + \hat{d}_i/(\hat{\lambda}_i - 1) + 2 \ln \hat{P}(w_1)/\hat{P}(w_2) \tag{3.28}$$

$$(p = p_{kr} + p_{ks}) \qquad (k = 1,2)$$

The density function of V_{kr}, $p_{kr}(h)$, is the convolution of p_{kr} densities of squared Gaussian variables having multiplicative constants. All p_{kr} densities lie above the

positive h axis with $a_{ki} \geq 0$. Similarly, the density function of V_{ks}, $p_{ks}(h)$, is the convolution of p_{ks} densities of squared Gaussian variables with multiplicative constants. All p_{ks} densities lie on the negative h axis with $a_{kj} < 0$.

A gamma density function is given by:

$$g_{p,\lambda} = \lambda^p \, x^{p-1} \, e^{-\lambda x} / \Gamma(p) \qquad (3.29)$$

Let k be a positive integer. With $p=1/2k$, and $\lambda = 1/2$, the gamma density $g(p,\lambda)$ is referred to as the chi-squared density with k degrees of freedom. (See (67),p.13).

Theorem 3.3

If $X_1,\ldots,X_n$ are independent random variables with gamma distributions $(p_1,\lambda),\ldots,(p_n,\lambda)$, then $Y=X_1+\ldots+X_n$ has a gamma distribution $(p_1+\ldots+p_n,\lambda)$.

Proof

See (67). p. 15.

Since what we have is the summation of chi-squared random variables (special form of a gamma distribution), both $p_{kr}(h)$ and $p_{ks}(-h)$ ($p_{ks}(h)$ reflected to the positive side) can be reasonably approximated by a general gamma form, especially for large n_{kr} and n_{ks}, as follows:

$$g(h) = \begin{cases} \dfrac{h^{\alpha}\, e^{-h/\beta}}{\beta^{\alpha+1}\, \Gamma(\alpha+1)} & h \geq 0 \\ 0 & h < 0 \end{cases} \qquad (3.30)$$

The parameters α and β can be determined so that the mean η and the variance σ^2 of the "true " distribution match those of the approximation.

Next, we calculate the expected values η_{kr} and η_{ks} of V_{kr}, and V_{ks}, and the variances σ^2_{kr} and σ^2_{ks}.

$$V_{kr} = \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki} (\xi_{ki} + b_{ki})^2 \qquad a_{ki} \geq 0$$

$$= \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki} (\xi^2_{ki} + 2\, b_{ki}\, \xi_{ki} + b^2_{ki})$$

$$E(V_{kr}) = \hat{\eta}_{kr} = \sum_{a_{ki} \geq 0}^{p_{kr}} (1 + 0 + b^2_{ki})$$

or,

$$\hat{\eta}_{kr} = \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki} (1 + b^2_{ki}) \qquad \text{for } p_{kr}(h) \qquad (3.31)$$

(ξ_{ki} has zero mean and unit variance)

Similarly,

$$\hat{\eta}_{ks} = \sum_{a_{kj} < 0}^{p_{ks}} a_{kj} (1 + b^2_{kj}) \qquad \text{for } p_{ks}(h) \qquad (3.32)$$

$$E(V_{kr}^2) = E\Big(\sum_{a_{ki},\, a_{kj} \geq 0}^{p_{kr}} a_{ki}\, a_{kj}\, (\xi_{ki}+b_{ki})^2\, (\xi_{kj}+b_{kj})^2$$

$$= E\Big(\sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki}^2 (\xi_{ki}^2 + 4\, b_{ki}\, \xi_{ki}^3 + 6\, b_{ki}^2\, \xi_{ki}^2$$

$$+ 4\, b_{ki}^3\, \xi_{ki} + b_{ki}^4)\Big) + 0$$

(The zero term comes because ξ_{ki} is independent from ξ_{kj} and hence they are mutually orthogonal as $E(\xi_{ki}) = 0$)

$$= \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki}^2\, (3 + 6\, b_{ki}^2 + b_{ki}^4) \qquad (3.33)$$

where $E(\xi_{ki}^n) = \begin{cases} 1.3. \ \ldots \ .(n-1) & \text{for n even} \\ 0 & \text{for n odd} \end{cases}$

$$E^2\,(V_{kr}) = \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki}^2 (1 + b_{ki}^2) + 0$$

$$= \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki}^2 (1 + 2\, b_{ki}^2 + b_{ki}^4) \qquad (3.34)$$

$$\text{Var}\,(V_{kr}) = \hat{\sigma}_{kr} = E(V_{kr}^2) - E^2\,(V_{kr})$$

$$= 2 \sum_{a_{ki} \geq 0}^{p_{kr}} a_{ki}^2 (1 + 2\, b_{ki}^2) \quad \text{for } p_{kr}(h) \qquad (3.35)$$

Similarly,

$$\hat{\sigma}_{ks} = 2 \sum_{a_{kj} < 0}^{p_{ks}} a_{kj}^2 (1 + 2\, b_{kj}^2) \qquad \text{for } p_{ks}(h) \tag{3.36}$$

For a random variable h, which has a gamma distribution with parameters α and β, (See equation (3.30)), then

$$E(h) = (\alpha + 1)\beta \qquad \text{Var}(h) = (\alpha + 1)\beta^2 \tag{3.37}$$

(See (67), p. 44)

Therefore, α_{kr}, α_{ks}, β_{kr}, β_{ks}, can be calculated as:

$$\alpha_{kr} = (\hat{\eta}_{kr}^2 / \hat{\sigma}_{kr}^2) - 1 \tag{3.38}$$

$$\alpha_{ks} = (\hat{\eta}_{ks}^2 / \hat{\sigma}_{ks}^2) - 1 \tag{3.39}$$

$$\beta_{kr} = \hat{\sigma}_{kr}^2 / \hat{\eta}_{kr} \tag{3.40}$$

$$\beta_{ks} = \hat{\sigma}_{ks}^2 / \hat{\eta}_{ks} \tag{3.41}$$

The density function $p(h/w_i)$, i=1,2, which is our final goal, is then the convolution of two gamma densities with a constant shift: one is distributed on the positive side of the h-axis, and the other on the negative side.

However, the convolution of these two gamma densitities is hard to obtain in an explicit mathematical expression, because in general, α is not an integer. Since we do not favor a numerical integration technique for calculating the error rate, a "modified " gamma distribution is proposed as follows:

$$g'(h) = \begin{cases} \dfrac{(h-c)^{\gamma} e^{-(h-c)/\delta}}{\delta^{\gamma+1} \Gamma(\gamma+1)} & \text{for } h \geq c \\ 0 & \text{for } h < c \end{cases} \qquad (3.42)$$

$\gamma = 0$ or 1

In other words, Gamma density curves are roughly categorized into two types: one is $\exp(-h/\beta)$, and the other is $h \exp(-h/\beta)$, depending on whether α obtained by (3.38) or (3.39) is larger than or smaller than a threshold value of 0.35. (The threshold value of 0.35 is a compromise value, chosen in an attempt to match the maximum value and location of the maximum value of the gamma density to the modified gamma approximation. It is further explained in (64)).

The procedure proposed by Fukunaga and Krile, then, is as follows:

1) Calculate $\hat{\eta}_{kr}$, $\hat{\eta}_{ks}$, $\hat{\sigma}^2_{kr}$, $\hat{\sigma}^2_{ks}$ from equations (3.31),(3.32),(3.35), and (3.36)

2) Calculate α_{kr} and α_{ks} form equations (3.38) and (3.39).

3) $\gamma_{kr} = 0$ if $\alpha_{kr} < 0.35$, and $\gamma_{kr} = 1$ if $\alpha_{kr} \geq 0.35$. Similarly for γ_{ks}.

4) Calculate δ_{kr}, δ_{ks}, and c_{kr}, c_{ks} by the following equations: (modified forms of equations (3.38)-(3.41))

$$\gamma_{kr} = \frac{(\hat{\eta}_{kr} - c_{kr})^2}{\hat{\sigma}^2_{kr}} - 1 \qquad (3.43)$$

$$\gamma_{ks} = \frac{(\hat{\eta}_{ks} - c_{ks})^2}{\hat{\sigma}^2_{ks}} - 1 \qquad (3.44)$$

$$\delta_{kr} = \hat{\sigma}^2_{kr} / (\hat{\eta}_{kr} - c_{kr}) \qquad (3.45)$$

$$\delta_{ks} = \hat{\sigma}^2_{ks} / (\hat{\eta}_{ks} - c_{ks}) \qquad (3.46)$$

Equations (3.43)-(3.46) are the same as (3.38)-(3.41), except for the shift of the mean c_{kr} or c_{ks}.

The convolution of $p_{kr}(h)$ and $p_{ks}(h)$, $p^*_k(h)$, k=1,2, can be obtained as an explicit expression. The result is : (See (64) for details)

$$p_k^*(t) = \begin{cases} \dfrac{\delta_{ks}^{\gamma kr}}{(\delta_{kr}+\delta_{ks})^{\gamma kr+1}} \left[-\dfrac{t}{\delta_{ks}} + \dfrac{(\gamma kr+\gamma ks)\delta_{kr}}{\delta_{kr}+\delta_{ks}} \right]^{\gamma ks} e^{t/\delta_{ks}} & \text{for } t \le 0 \\ \dfrac{\delta_{kr}^{\gamma ks}}{(\delta_{kr}+\delta_{ks})^{\gamma ks+1}} \left[\dfrac{t}{\delta_{kr}} + \dfrac{(\gamma kr+\gamma ks)\delta_{ks}}{\delta_{kr}+\delta_{ks}} \right]^{\gamma kr} e^{-t/\delta_{kr}} & \text{for } t \ge 0 \end{cases} \quad (3.47)$$

Defining the distance d as :

$$d_k = C - (c_{kr} - c_{ks}) \quad (3.48)$$

We can find e_1 by integrating $p_1^*(t)$ form d_1 to ∞, and e_2 by integrating $p_2^*(t)$ from $-\infty$ to d_2. The term d_k brings the shift parameter C back ino the picture, and also accounts for the displacement of the (h/w_k) approximations by c_{kr} and c_{ks}. In general,

$$D^*(d_k) = \int_{-\infty}^{d_k} p_k^*(t)dt =$$

$$\begin{cases} \left(\dfrac{\delta_{ks}}{\delta_{kr}+\delta_{ks}}\right)^{\gamma kr+1} \left[\dfrac{-d_k}{\delta_{ks}} + 1 + \dfrac{(\gamma kr+\gamma ks)\delta_{kr}}{\delta_{kr}+\delta_{ks}} \right]^{\gamma ks} e^{d_k/\delta ks}, & d_k \le 0 \\ 1 - \left(\dfrac{\delta_{kr}}{\delta_{kr}+\delta_{ks}}\right)^{\gamma ks+1} \left[\dfrac{d_k}{\delta_{kr}} + 1 + \dfrac{(\gamma kr+\gamma ks)\delta_{ks}}{\delta_{kr}+\delta_{ks}} \right]^{\gamma kr} e^{-d_k/\delta kr}, & d_k \ge 0 \end{cases}$$

(3.49)

where $D^*(d_k)$ is the approximation for $\text{Prob}(\hat{h}/w_k \leq 0)$. Thus, the approximated values of recognition errors are:

$$e_1 = \hat{P}(w_1)\ (1 - D^*(d_1)) \tag{3.50}$$

$$e_2 = \hat{P}(w_2)\ (D^*(d_2)) \tag{3.51}$$

3.2.3 Proposed, Modified Algorithm

Figure 3.2 shows a flowchart of Fukunaga's and Krile's algorithm. The algorithm assumes that the training statistics are an accurate representation of the true statistics of the two distributions. This being the case, the probability of correct classification that the algorithm projects is monotonically non-decreasing as a function of dimensionality. It is this drawback in the algorithm that we are trying to correct such that the algorithm would take into account the number of samples used for training.

Looking back at the calculation of the parameters of the modified gamma distribution, we see that all of them depend on two parameters, $\hat{\eta}_k$ and $\hat{\sigma}_k$, or the mean and variance of h. If these parameters are inaccurate, then all of the other parameters will be affected.

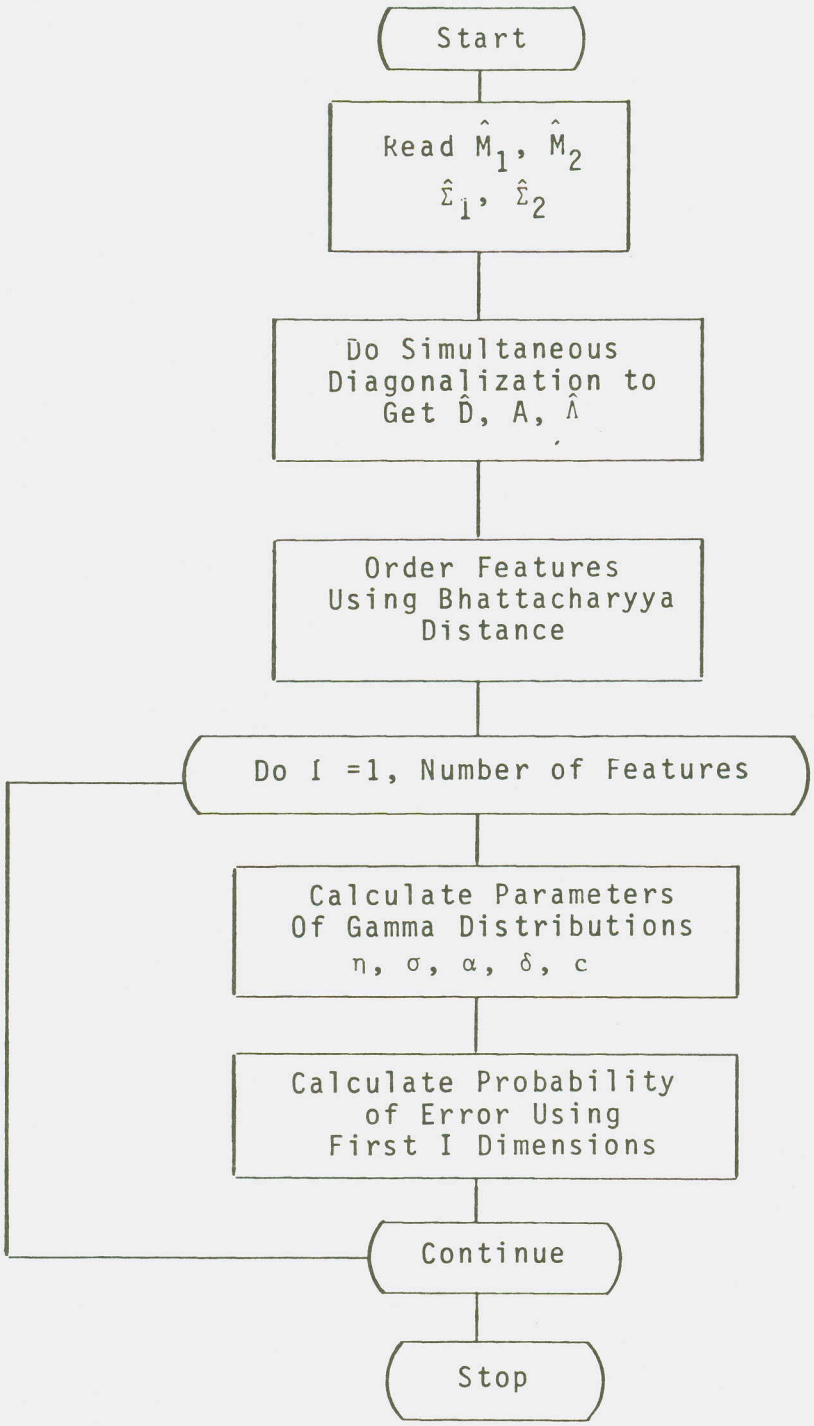


Figure 3.2 A Flowchart of Fukunaga and Krile's Algorithm.

We propose to look at the way these parameters, particularly $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, are distributed as a function of the number of training samples. We then want to incorporate that information in our estimation of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, such that the algorithm has a more realistic picture of what the training samples represent.

Estimating the probability density function of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ is by no means an easy task. For the amount of information that we have, such an estimation is very involved and impractical. A discussion of the difficulties one faces in attempting such an estimation is found in Appendix B.

We propose instead to look at the variances of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, and then incorporate that information in our estimation of these parameters.

Let us look at $\hat{\sigma}_1^2$, (Var $(\hat{h}/w_1)$) and $\hat{\sigma}_2^2$, (Var $(\hat{h}/w_2)$). From equation (3.35), (or (3.36)):

$$\hat{\sigma}_1^2 = 2 \sum_{i=1}^{p} a_{1i}^2 \, (1 + 2 \, b_{1i}^2) \tag{3.35}$$

Substituting for a_{1i} and b_{1i} by their values from (3.19) and (3.20) in (3.35), we get:

$$\hat{\sigma}_1^2 = 2 \sum_{i=1}^{p} (1 - 1/\hat{\lambda}_i)^2 \, (1 + 2 \, \hat{d}_i^2/(\hat{\lambda}_i - 1)^2) \tag{3.52}$$

After multiplying, this reduces to:

$$\hat{\sigma}_1^2 = 2 \sum_{i=1}^{p} (1 - 2/\hat{\lambda}_i + (2\hat{d}_i^2 + 1)/\hat{\lambda}_i^2) \qquad (3.53)$$

In matrix form, this can be written as:

$$\hat{\sigma}_1^2 = 2 (\mathrm{tr}(I - \hat{\Lambda}^{-1})^2 + 2 \hat{D}^T (\hat{\Lambda}^{-1})^2 \hat{D}) \qquad (3.54)$$

Or in terms of the original distributions:

$$\hat{\sigma}_1^2 = 2 (\mathrm{tr}(I - \hat{\Sigma}_2^{-1} \hat{\Sigma}_1)^2 + 2 \hat{m}_2^T \hat{\Sigma}_2^{-1} \hat{\Sigma}_1 \hat{\Sigma}_2^{-1} \hat{m}_2) \qquad (3.55)$$

(See (64)).

Similarly,

$$\hat{\sigma}_2^2 = 2 \sum_{i=1}^{p} a_{2i}^2 (1 + 2 b_{2i}^2)$$

$$= 2 \sum_{i=1}^{p} (\hat{\lambda}_i - 1)^2 (1 + 2 \hat{\lambda}_i \hat{d}_i^2 / (\hat{\lambda}_i - 1)^2)$$

$$= 2 \sum_{i=1}^{p} (\hat{\lambda}_i^2 + 2 (\hat{d}_i - 1)\hat{\lambda}_i + 1) \qquad (3.56)$$

In matrix form, $\hat{\sigma}_2^2$ can be written as:

$$\hat{\sigma}_2^2 = 2 (\mathrm{tr}(\hat{\Lambda} - I)^2 + 2 \hat{D}^T \hat{\Lambda} \hat{D}) \qquad (3.57)$$

Or, in terms of the original distributions:

$$\hat{\sigma}_2^2 = 2\left(\mathrm{tr}\left(\hat{\Sigma}_1^{-1}\hat{\Sigma}_2 - I\right)^2 + 2\,\hat{m}_2^T\hat{\Sigma}_1^{-1}\hat{\Sigma}_2\hat{\Sigma}_1^{-1}\hat{m}_2\right) \qquad (3.58)$$

(See (64)).

In order to calculate the variances of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, we make the following assumptions:

1. The original and transformed means, $\hat{M}_1, \hat{M}_2$, and $\hat{D}$ are assumed to be constant. Experience has shown that one can approximate first- order statistics with a relatively few number of training samples.
2. $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are independent. This is to say that we will ignore any relationships that might exist between the two classes.

Having assumed the above, the results are: (See Appendix C for the complete derivation)

$$\mathrm{Var}(\hat{\sigma}_1^2) = 4\sum_{i=1}^{p}\left(\frac{2}{\lambda_i^2}\left(\frac{4}{n_1}+\frac{4}{n_2}+\frac{8}{n_1n_2}\right) - \frac{4}{\lambda_i^3}\left(\frac{4}{n_1}+\frac{4}{n_2}+\frac{8}{n_1^2}\right.\right.$$

$$+\frac{8}{n_2^2}+\frac{32}{n_1n_2}+\frac{48}{n_1n_2^2}+\frac{48}{n_1^2n_2}+\frac{64}{n_1^2n_2^2}+\frac{4d_i^2}{n_1}+\frac{8d_i^2}{n_2}+\frac{24d_i^2}{n_1n_2}+\frac{16d_i^2}{n_2^2}$$

$$\left.\left.+\frac{32d_i^2}{n_1n_2^2}\right) + \frac{1}{\lambda_i^4}\left(\frac{8}{n_1}+\frac{8}{n_2}+\frac{128}{n_1n_2}+\frac{40}{n_1^2}+\frac{40}{n_2^2}+\frac{48}{n_1^3}+\frac{48}{n_2^3}+\frac{512}{n_1^2n_2}\right.\right.$$

$$+ \frac{1920}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_2^3 n_1} + \frac{2112}{n_1^2 n_2^3} + \frac{2112}{n_1^3 n_2^2} + \frac{2304}{n_1^3 n_2^3} + 4d_i^2 \left(\frac{4}{n_1} + \frac{8}{n_2} \right.$$

$$\left. + \frac{8}{n_1^2} + \frac{40}{n_2^2} + \frac{64}{n_1 n_2} + \frac{256}{n_1 n_2^2} + \frac{96}{n_1^2 n_2} + \frac{48}{n_2^3} + \frac{288}{n_1 n_2^3} + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_1^2 n_2^3} \right)$$

$$\left. + 4d_i^4 \left(\frac{2}{n_1} + \frac{8}{n_2} + \frac{40}{n_2^2} + \frac{24}{n_1 n_2} + \frac{48}{n_2^3} + \frac{88}{n_1 n_2^2} + \frac{96}{n_1 n_2^3} \right) \right) \right) \qquad (3.59)$$

$$\mathrm{Var}(\hat{\sigma}_2^2) = 4 \sum_{i=1}^{p} \left[\lambda_i^4 \left(\frac{8}{n_1} + \frac{8}{n_2} + \frac{128}{n_1 n_2} + \frac{40}{n_1^2} + \frac{40}{n_2^2} + \frac{48}{n_1^3} + \frac{48}{n_2^3} + \frac{512}{n_1^2 n_2} \right.\right.$$

$$+ \frac{512}{n_1 n_2^2} + \frac{1920}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_1 n_2^3} + \frac{2112}{n_1^3 n_2^2} + \frac{2112}{n_1^2 n_2^3} + \frac{2304}{n_1^3 n_2^3} \Big) + 4\lambda_i^3 \left(d_i^2 \left(\frac{8}{n_1} \right.\right.$$

$$+ \frac{4}{n_2} + \frac{8}{n_2^2} + \frac{40}{n_1^2} + \frac{64}{n_1 n_2} + \frac{256}{n_1^2 n_2} + \frac{96}{n_2^2 n_1} + \frac{48}{n_1^3} + \frac{288}{n_2^3 n_1} + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_2^2 n_1^3} \Big)$$

$$\left. - \left(\frac{4}{n_1} + \frac{4}{n_2} + \frac{8}{n_1^2} + \frac{8}{n_2^2} + \frac{32}{n_1 n_2} + \frac{48}{n_1 n_2^2} + \frac{48}{n_1^2 n_2} + \frac{64}{n_1^2 n_2^2} \right) \right) + 2\lambda_i^2 \left(\left(\frac{4}{n_1} \right.\right.$$

$$+ \frac{4}{n_2} + \frac{8}{n_1 n_2} \Big) + 2d_i^4 \left(\frac{8}{n_1} + \frac{2}{n_2} + \frac{40}{n_1^2} + \frac{24}{n_1 n_2} + \frac{48}{n_1^3} + \frac{88}{n_1^2 n_2} + \frac{96}{n_1^3 n_2} \right)$$

$$\left. \left. - 4d_i^2 \left(\frac{2}{n_2} + \frac{4}{n_1} + \frac{12}{n_1 n_2} + \frac{8}{n_1^2} + \frac{16}{n_1^2 n_2} \right) \right) \right] \qquad (3.60)$$

Note that $\mathrm{Var}(\hat{\sigma}_1^2)$ and $\mathrm{Var}(\hat{\sigma}_2^2)$ are inversely proportional to the number of training samples used to estimate the statistics of classes 1 and 2, and directly proportional to the number of dimensions. In other words, as the number of training samples increases, the variances of our

estimates of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ decrease, as expected. Also, as the number of dimensions is increased, the variances of the estimates increase.

Since we do not have the probability density functions of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, we want to think of a reasonable way to incorporate the effect of the number of training samples into our estimation of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$. We claim that a better estimation of the true variances σ_1^2 and σ_2^2 consists of our estimation of these variances, $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, plus some multiplicative factor of the standard deviations of these estimates, namely the square roots of $\mathrm{Var}(\hat{\sigma}_1^2)$ and $\mathrm{Var}(\hat{\sigma}_2^2)$, that were calculated above.

This multiplicative factor was chosen empirically. Experimental results in Chapter 4 show that the variance of the probability of error generally increases with increasing dimensionality, especially in the presence of a very limited training data set. Results also show that the probability of error is inversely proportional to the number of training samples. Moreover, it is very sensitive to the number of training samples in the cases where that number is not much greater than the number of dimensions.

Based on the above observations, the following empirical formula for the multiplicative factor was used:

$$\text{M.F.} = 2\, p^2/(n_1 \cdot n_2) \qquad (3.61)$$

where p is the number of dimensions

n_1 and n_2 are as before.

The new procedure to calculate the probability of error, becomes as follows:

1) Calculate $\hat{\eta}_{kr}$, $\hat{\eta}_{ks}$, $\hat{\sigma}_{kr}$, $\hat{\sigma}_{ks}$, from equations (3.31), (3.32), (3.35), and (3.36)
2) Update $\hat{\sigma}^2_{kr}$ and $\hat{\sigma}^2_{ks}$ as follows:

$$\hat{\sigma}^2_{kr} \text{ (new)} = \hat{\sigma}^2_{kr} \text{ (old)} + (2p^2/n_1.n_2).\ (\text{Var}(\hat{\sigma}^2_{kr}))^{\frac{1}{2}}$$

$$\hat{\sigma}^2_{ks} \text{ (new)} = \hat{\sigma}^2_{ks} \text{ (old)} + (2p^2/n_1.n_2).\ (\text{Var}(\hat{\sigma}^2_{ks}))^{\frac{1}{2}}$$

3) $\gamma_{kr} = 1$ if $\alpha_{kr} \geq 0.35$, and $\gamma_{kr} = 0$ if $\alpha_{kr} < 0.35$. Similarly for γ_{ks}.

4) Calculate δ_{kr}, δ_{ks}, and c_{kr}, c_{ks}, from equations ((3.43) - (3.46)).
5) Calculate p^*_k (t) and $D^*(d_k)$ from equations (3.47) and (3.49).
6) Calculate the probability of error from equations (3.59) and (3.60).

We are ready now to procced to Chapter 4, where several experimental results are shown.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

Some results on feature selection techniques will be presented first. Next, several experimental results illustrating the Hughes phenomenon are shown. Results comparing probabilities of error predicted by the proposed algorithm as a function of dimensionality as compared to experimental observations are then presented for aircraft and Landsat data. Results are obtained for both real and simulated data. Finally, two binary tree classification procedures that make use of the algorithm are presented to illustrate the usefulness of the procedure.

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.

Detailed training and test field descriptions for all the experiments conducted are found in Appendix F.

4.2 Experiments on Feature Selection Techniques

In this section, some experiments on different feature selection techniques are presented. The purpose of conducting these experiments is to choose an effective feature selection technique, particularly when dealing with a small number of training samples.

Experiment 4.1

Two classes of wheat and corn are selected from multispectral scanner (hereafter referred to as MSS or aircraft) data of the 1971 Flightline 210 from the Corn Blight Watch Experiment, and classified. The data was collected on August 13, 1971. Part of the selected data is used for training and a much larger portion is used for testing. The number of features used for classification varies from one to twelve, and the number of training samples for each class is chosen such that it is much higher than the number of features (265 samples for wheat, 569 samples for corn). A principle components (Karhunen-Loeve) transformation is applied to the data, and then three feature selection techniques are compared:

1) In the first feature selection method, the features are ordered according to the largest eigenvalues resulting from the K-L expansion. This method, referred to hereafter as the K-L ordering method,

assumes that the best feature is that which corresponds to the largest eigenvalue of the mixture covariance matrix of the whole data set, the second best corresponds to the second largest eigenvalue, ...etc. This ordering then imposes the condition that a feature subset with lower dimensionality is always a subset of another with higher dimensionality. The method then depends on the eigenvalues of the mixture covariance matrix, and ignores any among-class variabilities.

2) The second feature selection technique method is referred to as the Transformed Divergence method (13). The transformed divergence, D_T, is defined as follows:

$$D_T = 2000\ (1 - e^{-D/8}) \qquad (4.1)$$

where D is the divergence of two normal distributions, and is defined as follows (12):

$$D = \frac{1}{2}\ tr\ (\hat{\Sigma}_1-\hat{\Sigma}_2)(\hat{\Sigma}_2^{-1}-\hat{\Sigma}_1^{-1}) + \frac{1}{2}\ (\hat{M}_1-\hat{M}_2)^T(\hat{\Sigma}_1^{-1}+\hat{\Sigma}_2^{-1})(\hat{M}_1-\hat{M}_2) \qquad (4.2)$$

For a given dimensionality, the method chooses the feature subset with that dimensionality which gives the largest value of D_T. Unlike the K-L method, a feature subset of lower dimensionality is not neces-

sarily a subset of another with higher dimensionality. This method is applied to the data after it has been K-L transformed.

3) The third feature selection technique method used is the Bhattacharyya distance (16), defined by equation (2.9). In this method, a simultaneous diagonalization technique is applied to the covariance matrices of the two classes (after a K-L transformation of the data), and the best feature is then selected as that which corresponds to the largest value of B as defined by equation (2.10). The second largest is that which corresponds to the second largest B, and so on. As in the K-L method, a feature subset of lower dimensionality is always a subset of one with higher dimensionality. The transpose of the eigenvector matrix obtained is then multiplied by the observation vectors to transform the data, the mean vectors and the covariance matrices are transformed, and the data classified.

Results are shown in Figure 4.1, which plots the recognition accuracy (P_{cc}%) as a function of dimensionality. It is seen that of the three methods, the transformed divergence one gives the poorest performance. The K-L method is better, but the best method is that obtained from the Bhattacharyya ordering, which saturates at a very low dimension-

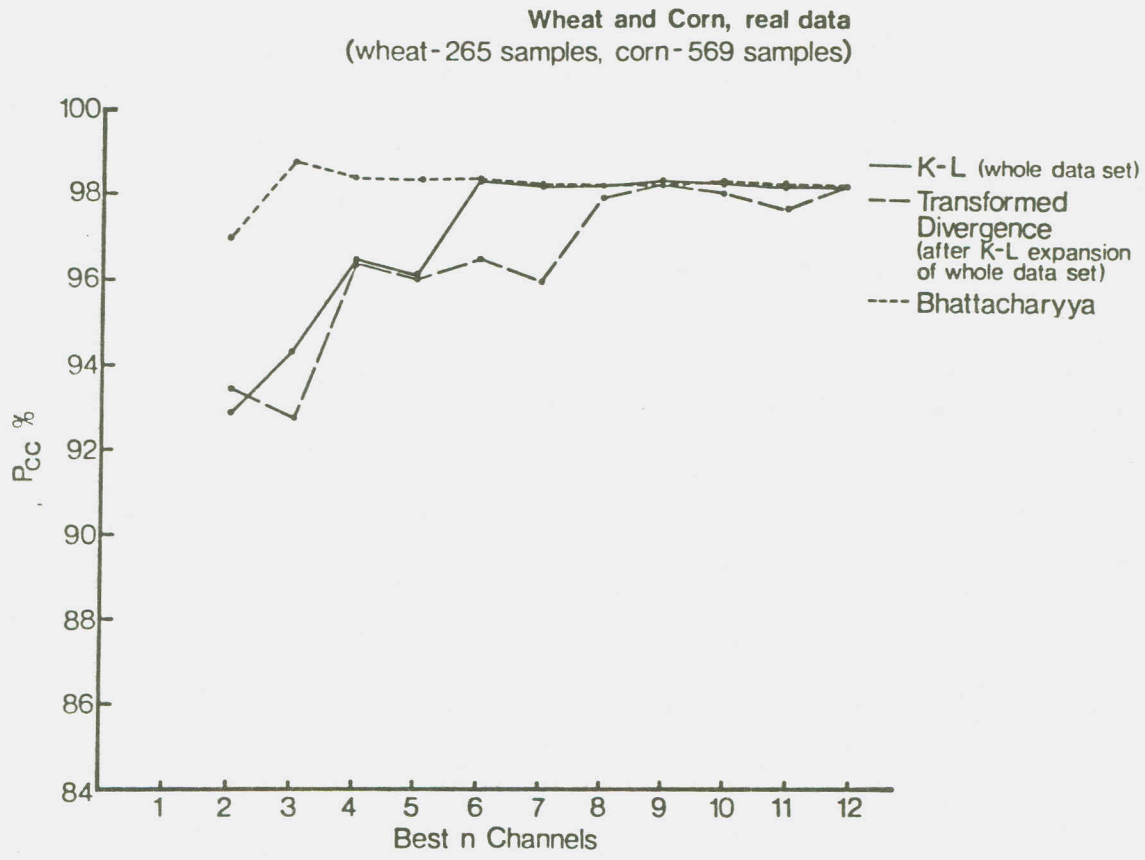


Figure 4.1 Classification Results of Data in Experiment 4.1 Using Three Feature Selection Techniques.

ality. Note that as dimensionality increases, the three curves start approaching each other, until they all coincide when all features are used (The probability of error is invariant under any linear transformation).

Experiment 4.2

In this experiment, 20 samples each of wheat and corn are chosen randomly from the training samples of experiment 4.1. The test samples are the same in both experiments. Again, the same three feature selection techniques elaborated upon above are used. Classification results are shown in Figure 4.2. Unlike the results in experiment 4.1, the Bhattacharyya ordering here gives the poorest results. Further, it does not exhibit a peaking effect, an effect that is expected when working with such a small number of training samples. The transformed divergence ordering does much better and does exhibit a peaking effect. However, it has a lot of fluctuations. The K-L ordering, on the other hand, while giving slightly poorer results than transformed divergence at low dimensionality, is better than the other two techniques at high dimensionality and has less fluctuations.

Figure 4.2 Classification Results of Data in Experiment 4.2 Using Three Feature Selection Techniques.

Experiment 4.3

Another two classes, corn and forest, are selected from the same data set described in experiment 4.1. Again, 20 samples per class are chosen randomly from a larger set of training samples, and the three feature selection techniques are compared. Results appear in Figure 4.3

Again, we notice that the Bhattacharyya ordering does poorer than the other two techniques, and does not exhibit a peaking effect. Transformed divergence gives better results, but again has a lot of fluctuations. The K-L ordering is superior to both, and has less fluctuations.

It should be noted again that the K-L ordering we used is based over the full data set. It is dependent on the mixture covariance matrix of the full data set, and thus ignores any between class variabilities resulting from differences between class covariance matrices. Because it is always dependent on the full data set, the number of training samples used to estimate the mixture covariance mtrix is almost always large, and hence a good estimate is obtained.

The Bhattacharyya ordering used, on the other hand, although it takes into account between class variabilities, depends heavily on the number of training samples used to estimate the individual covariance matrices of the classes at hand. Thus, as the number of training samples decreases,

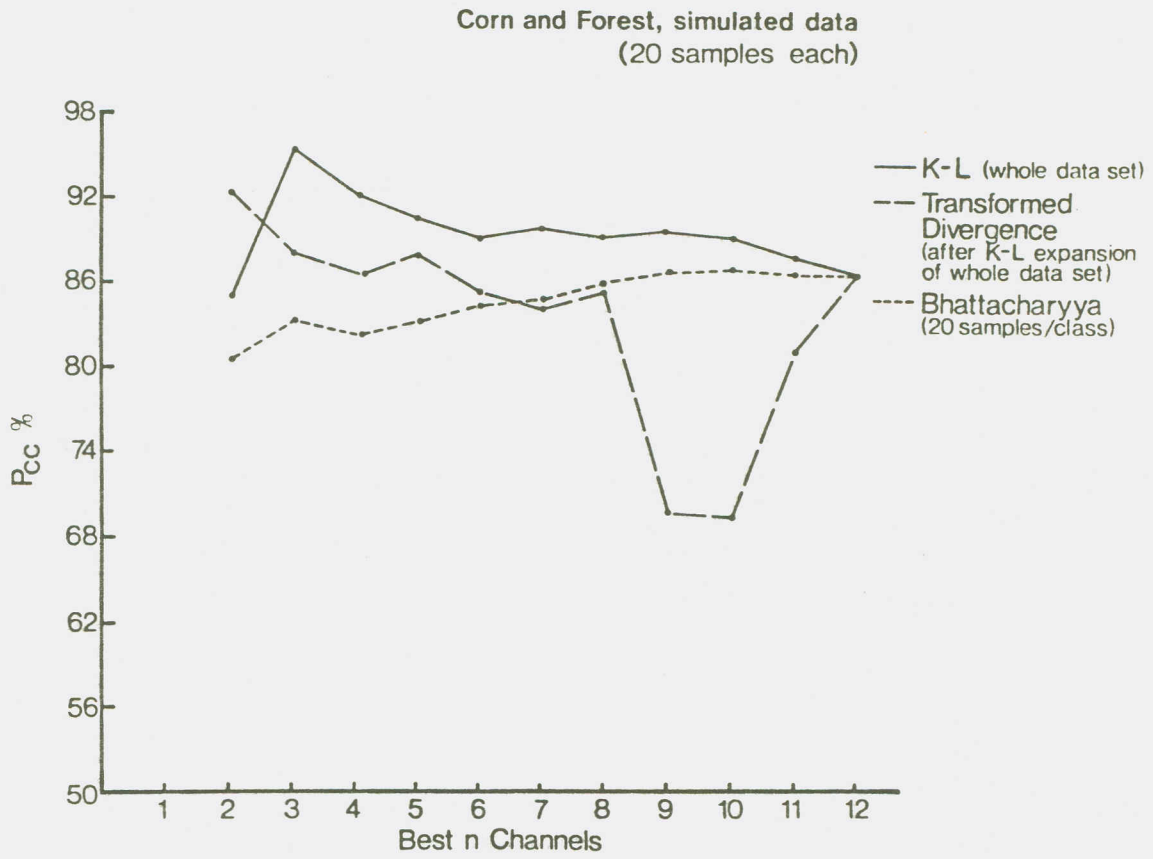


Figure 4.3 Classification Results of Data in Experiment 4.3 Using Three Feature Selection Techniques.

poorer estimates of the covariance matrices are obtained, leading to poorer transformations.

It appears that the transformation obtained from the simultaneous diagonalization technique is very sensitive to the number of training samples used to estimate the statistics of the classes at hand. While it produces superior results when there are enough samples, it fails to do so when the training samples are limited.

Indeed, Wu (50) published results in which he showed that the divergence criterion breaks down when the number of training samples is small, and no longer is an effective predictor of accuracy.

The K-L ordering, while ignoring the among-class variabilities in the scene, is only dependent on the number of data points in the data set used to approximate the mixture covariance matrix, but is otherwise independent of the number of training samples used. Thus, while sacrificing the information we get about the variability between classes in the set, experimental results show that this sacrifice is more than warranted when dealing with a small number of training samples. While not claiming that the K-L ordering gives the optimal results, we think it is a very effective procedure in the presence of few training samples, that is not surpassed by any other procedure that we know of, given the circumstances above.

Based on the above, and on the fact that the K-L ordering is a very efficient technique in that it reduces the number of permutations of features that have to be searched through to only the number of features present, it will be used as a feature selection technique throughout the remainder of the experiments.

4.3 Experiments on the Hughes Phenomenon

In this section, some experimental results that illustrate the Hughes phenomenon will be presented. The objectives of conducting these experiments are to demonstrate the existence of this phenomenon in remote sensing applications, and to verify the hypothetical explanation of it. Experiments will be performed on aircraft and Landsat data, both simulated and real. In all the following experiments, no results are obtained for the dimensionality of one. Tabulated classification results are found in Appendix D.

Experiment 4.4

The data set described in experiment 4.1 is simulated using the algorithm described in section 2.4. Two classes, corn and forest, are selected and 500 training samples are chosen for each class. A larger, mutually exclusive set is

used for testing. The K-L method is used in ordering the features, and the data selected is classified using the best 2,3,4,... ,12 features. Subsequently, 5 training sets are randomly chosen from the larger training set, each set having 20 samples per class of corn and forest. The five sets are classified, using the same test fields above, and the average classification accuracy, (sometimes referred to as the probability of correct classification, or P_{cc}), is calculated for each subset of features. Another 5 training sets are then randomly chosen, this time with 13 samples per class of corn and forest (The minimum number of samples possible for 12 features without getting singular covariance matrices). Again, the 5 sets are classified and the average classification accuracy is calculated for each feature subset. The results are then plotted in Figure 4.4.

Looking at Figure 4.4, it is seen that when the number of training samples is adequate, as in the 500 samples per class case, the probability of correct classification is a monotonically non-decreasing function of dimensionality. Since in a K-L ordering, the information is concentrated in the first few channels, we notice that after the best 5 features, the recognition accuracy tends to saturate.

When the number of training samples per class drops to 20, however, we see that not only does the accuracy drop from the 500 samples case, but also it exhibits a slight Hughes phenomenon. Although the curve has a maximum at

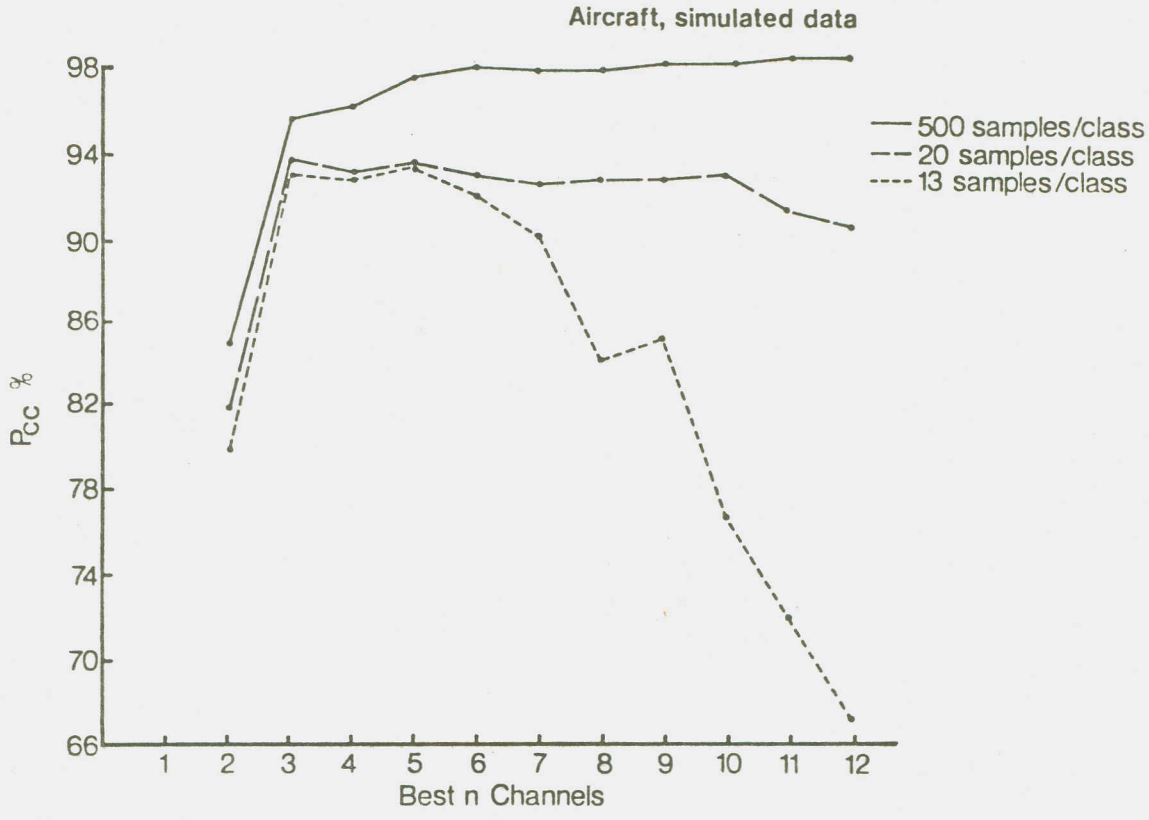


Figure 4.4 Experimental Classification Results of Aircraft, Simulated Data Using Different Numbers of Training Samples.

dimensionality 3, it is approximately constant until the best 10 features, after which it starts decreasing, even though slightly.

The 13 samples per class case offers a dramatic change from the two other curves. There is a clear peaking effect here, with the curve reaching a maximum at dimensionality 5, after which it drops drastically.

The results conform with the hypothetical curves of Figures 2.1 and 2.2. The 20 samples and 13 samples curves can be made smoother if more than 5 sets are averaged, and hence we should look at them with the idea in mind that these are only approximations of what the true curves look like. However, the trend these curves point to is clear. In the presence of a limited set of training samples, an increase in dimensionality can result in a decrease in the classification accuracy, with this effect disappearing as the number of training samples increases.

Experiment 4.5

The same aircraft data set as that used in experiment 4.1 is used, but without any simulation. 400 samples each of corn and forest are selected for training, and a larger, separate set is used for testing. Again, 5 different training sets of 20 and 13 samples per class are randomly chosen

from the original training set and classified. The average classification results for each feature subset are calculated and plotted. Results appear in Figure 4.5.

The curves in Figure 4.5 are not as smooth as they are in Figure 4.4. This is attributed to the fact that we are working with real data, which does not as well satisfy the assumptions we make as the simulated data does. Still, the curve with the 13 samples does generally poorer than the other two curves and drops dramatically in accuracy, whereas the 400 samples curve appears to saturate almost from the start. The 20 samples curve appears to have a slight peaking effect, although the curve is very noisy.

Experiment 4.6

The data set used in this experiment is obtained from Landsat, flown over Henry County, Indiana. To obtain a data set with more than the 4 features available from Landsat on any particular date, four data sets flown over the site at different times are used. The dates the data was collected on are: June 9, July 16, August 20, and September 26, all in 1978. The data is concatenated, and a K-L transformation was performed on it. Simulated data, more precisely meeting such assumptions as normality is generated, and the first 12 channels are used for classification. We will refer to this data as multitemporal data to indicate that it is collected over different times.

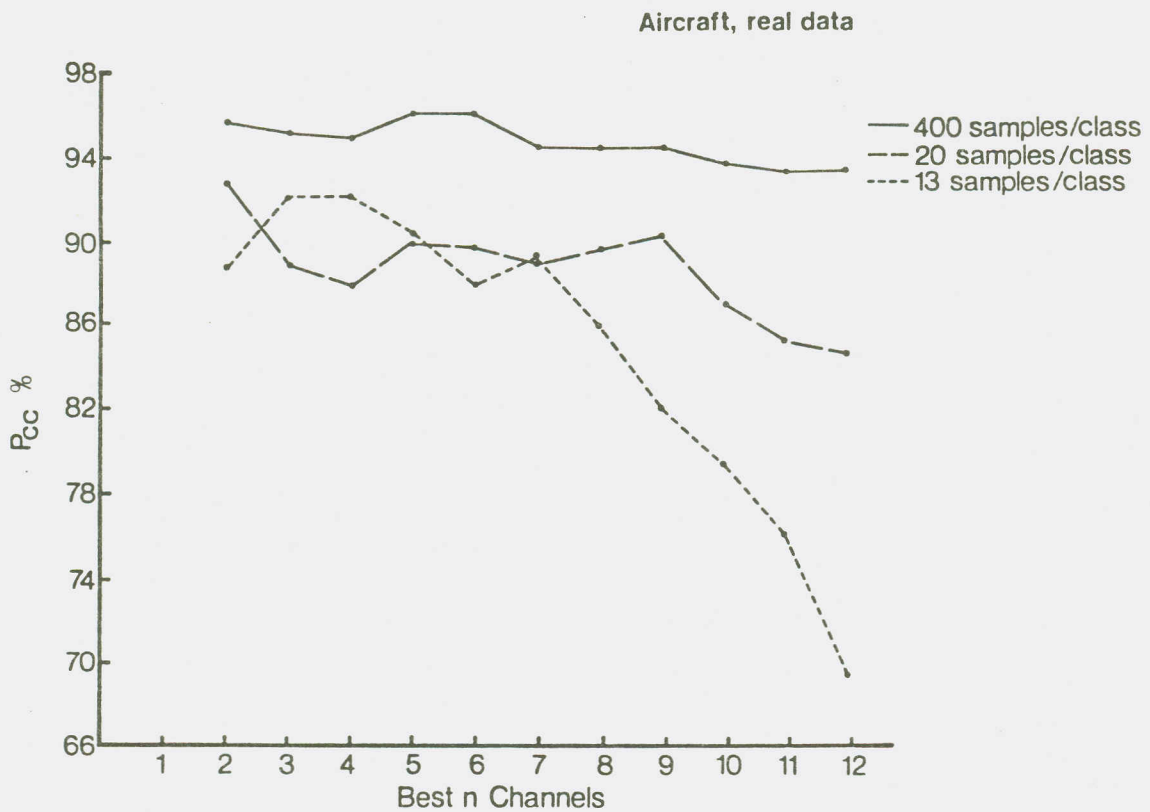


Figure 4.5 Experimental Classification Results of Aircraft, Real Data Using Different Numbers of Training Samples.

Two classes, corn and soybeans, are selected with 250 samples per class for training, and a larger independent set for testing. Again, 5 different training sets of 20 and 13 samples per class are chosen from the original training set and classified. Results are averaged and plotted in Figure 4.6.

The same results obtained in the previous two experiments are again evident. Note that even with 20 or 13 samples per class, the accuracy obtained is very close to that obtained by using all the available training samples. This is due to the fact that the two classes chosen are highly separable and thus are easily distinguishible even when using a small number of training samples to estimate their statistics.

Experiment 4.7

The same data set as experiment 4.6 is used, but without any simulation. Two classes, corn and soybeans, are selected with 250 samples per class used for training, and a larger, separate, set for testing. Again, 5 different training sets of 20 and 13 samples per class are randomly chosen from the original training set and classified. Results are averaged and plotted in Figure 4.7.

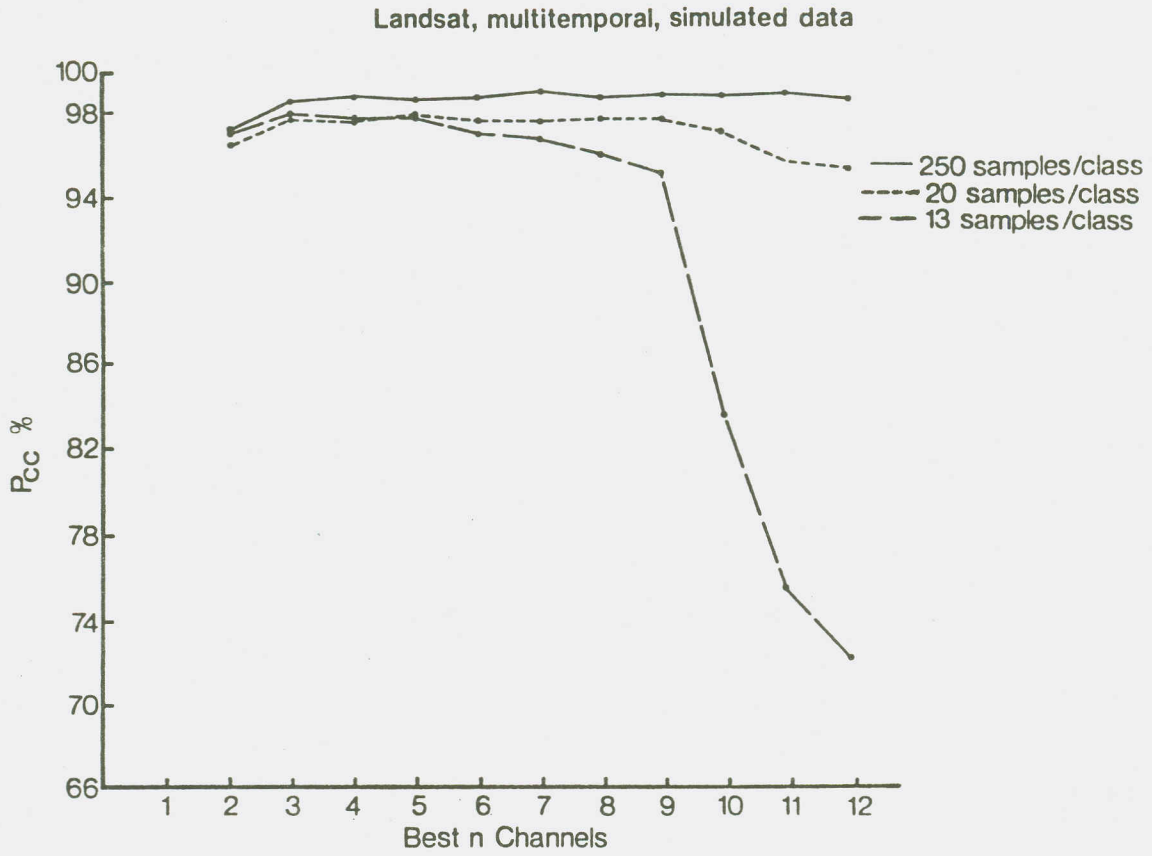


Figure 4.6 Experimental Classification Results of Landsat, Multitemporal, Simulated Data Using Different Numbers of Training Samples.

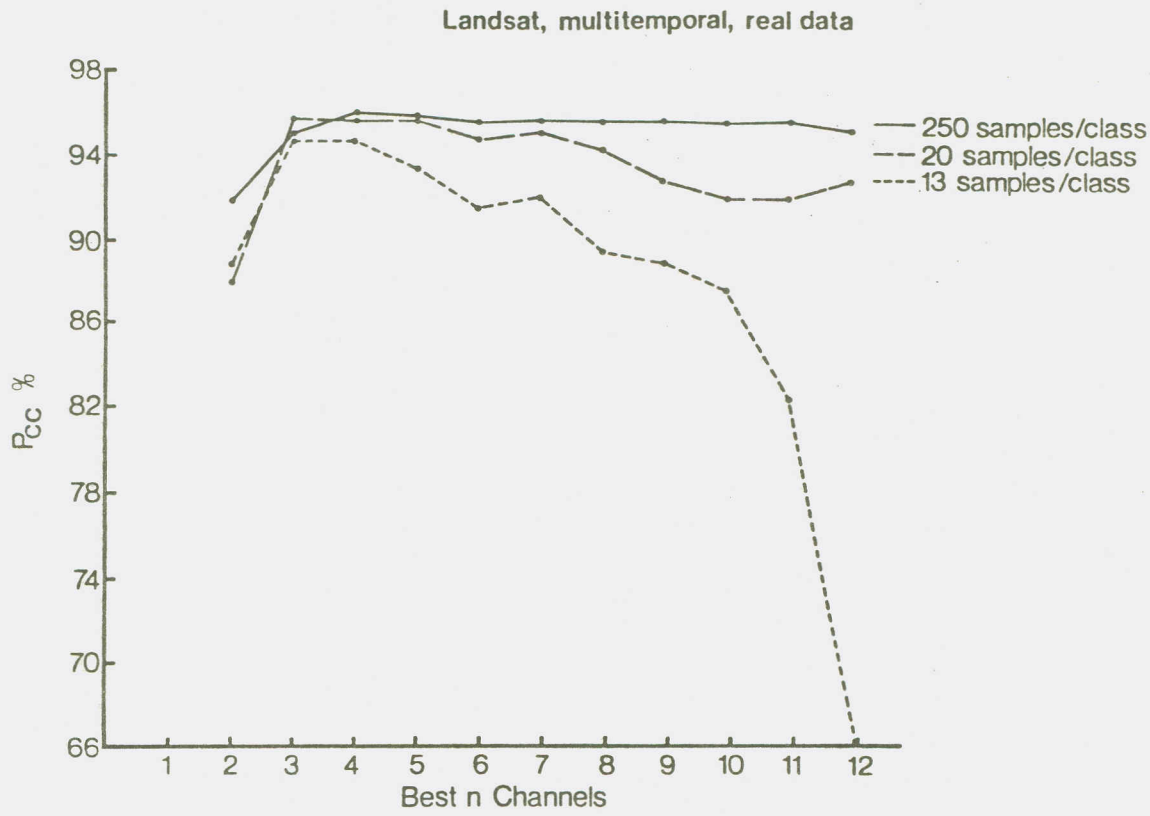


Figure 4.7 Experimental Classification Results of Landsat, Multitemporal, Real Data Using Different Numbers of Training Samples.

The same observations noticed in the three previous experiments apply here. There is a drastic drop in accuracy when 13 samples are used, a slight one when 20 samples are used, and no drop when 250 samples are used.

Summarizing the results of the last four experiments, we see that there is a definite Hughes phenomenon in the presence of a limited number of training samples compared to the number of features used. Further, as the number of samples increases, the accuracy for any given dimensionality increases, and the peak in the curve shifts to the right, i.e., the peaking effect takes place at a higher dimensionality, as is seen in Figures 4.4-4.7.

Studying Figures 4.4-4.7 reveals that the region between 13 samples and 20 samples is a very sensitive one when working with a maximum dimensionality of 12. While there is a sharp decline in accuracy at 13 samples per class, there is only a slight one at 20 samples per class. Another point to note is that the 20 and 13 samples are chosen from spectrally homogeneous classes, and so a very large number of samples is not needed to estimate the statistics of these classes. In a practical situation, the 20 and 13 samples curves might not be as close to the curves with large numbers of training samples as they are in these experiments.

The results of the last four experiments were a factor in choosing the empirical formula, or equation (3.61), discussed in Section 3.2.3. A formula was sought that takes the sensitivity in the number of training samples into account, as well as other factors that were discussed earlier.

4.4 Experiments Comparing Algorithm and Experimental Results

In this section, several experiments will be conducted to assess the performance of the proposed algorithm. Again, aircraft and Landsat data are used, both simulated and real, and the number of training samples used will be varied. But first, we will reproduce the results obtained by Fukunaga and Krile (64) to verify the validity of the algorithm.

Experiment 4.8

The data set used by Fukunaga and Krile is described in detail in Marill and Green (12). The data is simulated, has two classes and eight features. Each class has 200 training samples, and both the exact, or true, and the algorithm recognition rates are calculated. The true recognition rates are not calculated again in here, but are reproduced from Fukunaga and Krile, who used numerical integration to arrive at them.

Two methods used by Fukunaga and Krile are employed here: The normal assumption, discussed briefly in Section 3.2.1, and the modified gamma assumption, discussed in Section 3.2.2 and used throughout this research. The Bhattacharyya distance was used by Fukunaga and Krile, and although we have shown it to have limitations, it is used as a criterion for ordering the features. Results appear in Figure 4.8.

The results show that the modified gamma assumption method is a reasonable approximation of the true probability of correct classification. The normal assumption, though, does not give a good approximation of P_{cc}, and hence it is not further used.

While in this experiment, the modified gamma assumption is compared to the true probability of error, in actual practice the true probability of error cannot be calculated because the underlying distributions are not known. Therefore, in the following experiments, the proposed algorithm is compared to an average of five classifications obtained from five different training sets having the same number of training samples. This average classification serves as an estimate of the "true" error curve. This fact should be remembered as the experimental curves that are obtained are not as "smooth" as what the true curves would be expected to

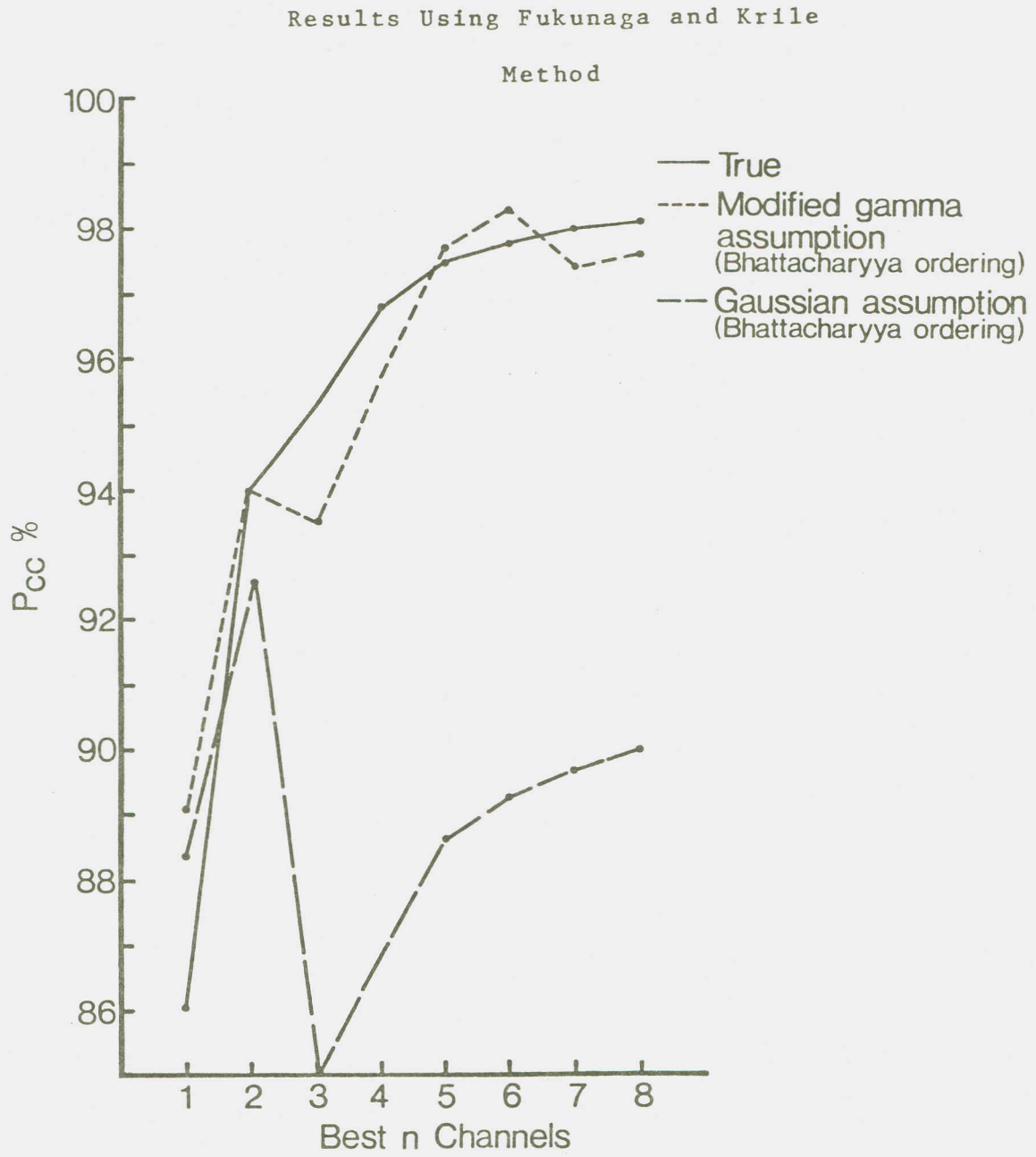


Figure 4.8 Classification Results of Fukunaga and Krile's Example Reproduced.

be. The algorithm curves, on the other hand, being dependent, among other things, on the number of training samples in an average way, are expected to be "smoother" than the experimental ones.

Before we embark on studying the next experiments, it is appropriate at this point to look at a flowchart describing the modified algorithm that is proposed. This is shown in Figure 4.9. This figure is to be compared to Figure 3.2, or Fukunaga and Krile's algorithm, to see the changes that are made.

Experiment 4.9 (Aircraft, Simulated Data, 20 Samples per Class)

The simulated, aircraft data set used in Experiment 4.4 is used here. Two classes, corn and forest, are used. The experimental, 20 samples per class curve, in Figure 4.4 is plotted again in Figure 4.10, together with the approximation to the probability of correct classification predicted by the proposed algorithm. Also plotted in Figure 4.10 are the standard deviations for each feature subset of the five different classifications performed.

We see that the algorithm is a good approximation to the experimental curve. The approximation is not as good at lower dimensionalities as it is at higher ones, because the

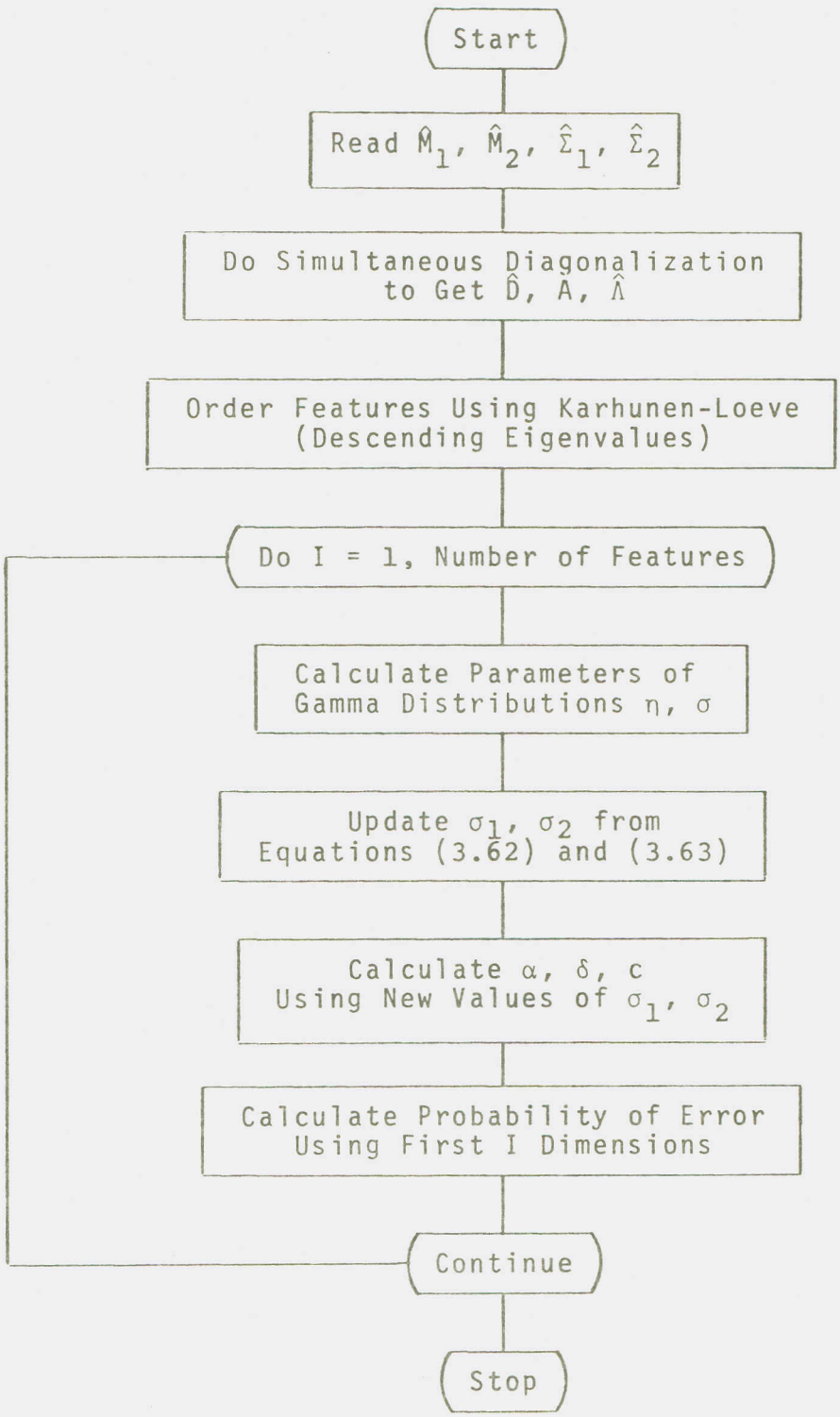


Figure 4.9 A Flowchart of the Modified Algorithm.

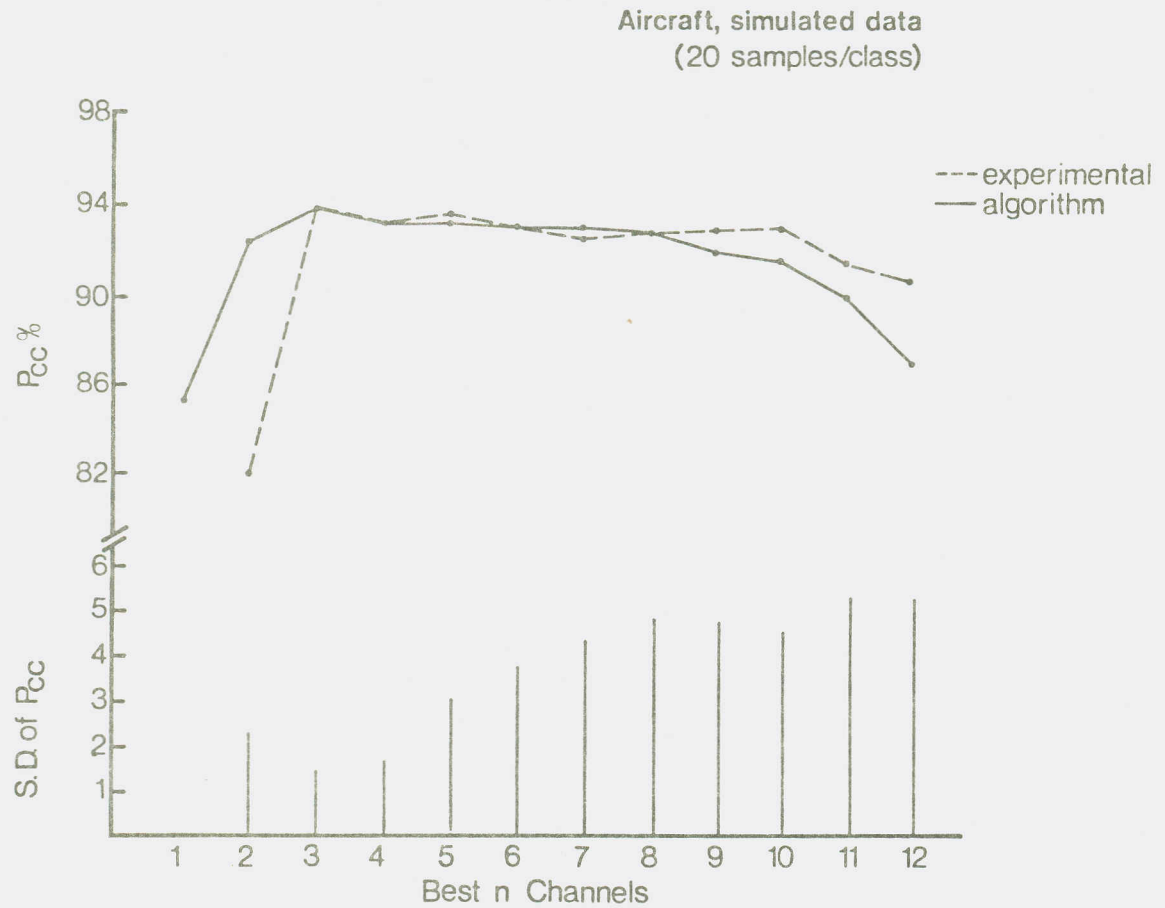


Figure 4.10 Classification Results of Aircraft, Simulated Data, Using 20 Samples per Class.

assumptions the algorithm makes are better at higher dimensionalities. However, the two curves do peak at the same dimensionality, 3, but more importantly, they have a similar shape. Both remain relatively constant for a while and then start decreasing at about the dimensionality of 8.

Examining the standard deviations of P_{cc}, it is observed that in general they have an increasing trend as the dimensionality increases. Put in other words, the curves indicate that the variance of the probability of error seems to increase with increasing dimensionality. This agrees with the hypothetical explanation given of the Hughes phenomenon, namely that the accuracy of the estimated statistics decreases with increasing dimensionality (i.e. becoming more random and hence increasing the variance of error) and that when this effect outweighs the increase in separability between classes due to increasing dimensionality, a peaking effect is observed. As the number of samples is decreased, larger increases in the variance of error are expected.

Experiment 4.10 (Aircraft, Simulated Data, 13 Samples per Class)

The same example used in Experiment 4.9 is used again, but with 13 samples per class used for training. The exper-

imental curve of Experiment 4.4 is reproduced, together with the curve predicted by the algorithm. The standard deviation of P_{cc} is again plotted. Results appear in Figure 4.11.

Again, the curve predicted by the algorithm is a better approximation of the experimental curve at high dimensionality. The experimental curve, however, is not very sensitive to dimensionality at lower values, and thus a small ambiguity in where the peak occurs can be afforded. Still, both curves predict a peak at 3. The standard deviation of the error again has an increasing trend as dimensionality increases.

Experiment 4.11 (Aircraft, Real Data, 20 Samples per Class)

The example used in Experiment 4.5 is repeated. Again, two classes are used, corn and forest, from the aircraft, real data set. Twenty samples per class are used for training, and five different sets of training samples are classified and averaged. The average is then compared to the algorithm performance. Results appear in Figure 4.12.

The experimental curve has a lot of error variance as can be seen from the curve and does not seem to be following any general pattern, although it starts consistently decreasing after dimensionality 9. It is interesting to

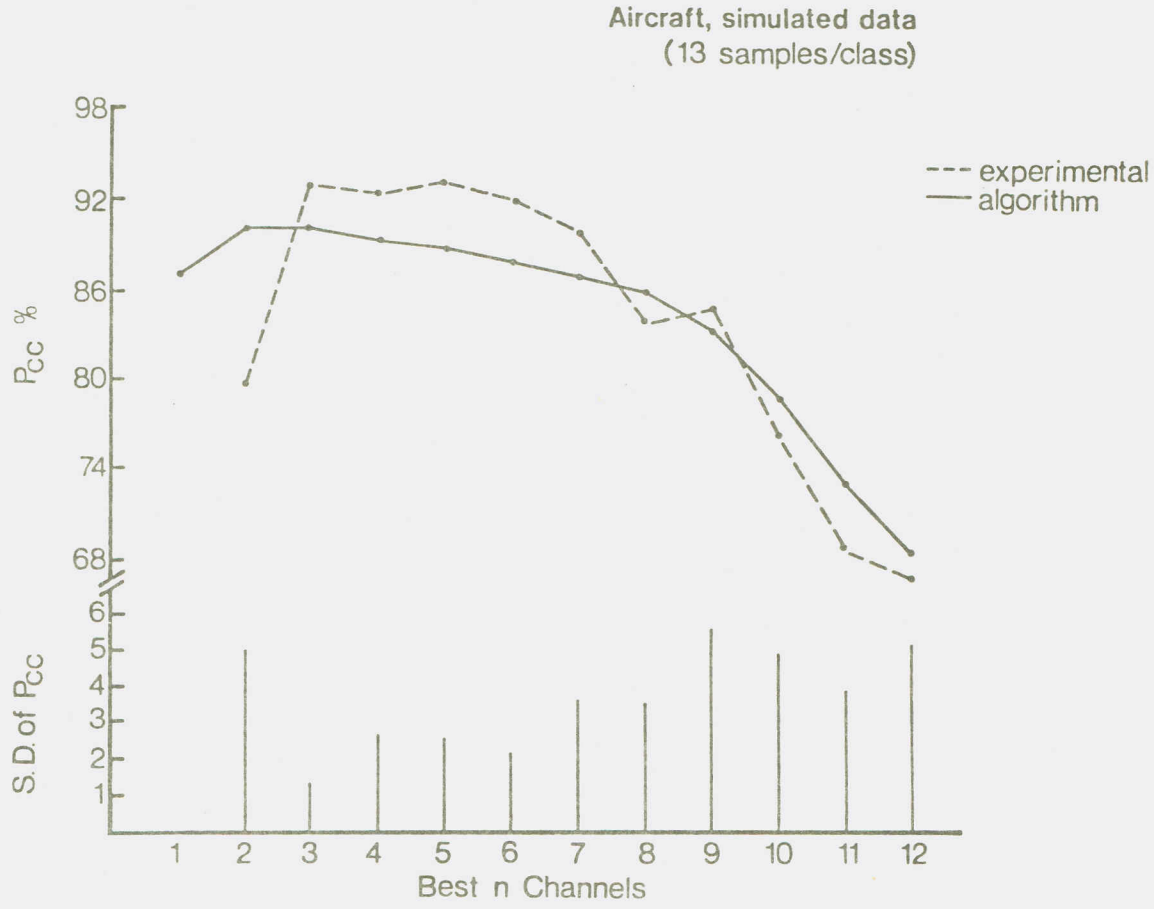


Figure 4.11 Classification Results of Aircraft, Simulated Data, Using 13 Samples per Class.

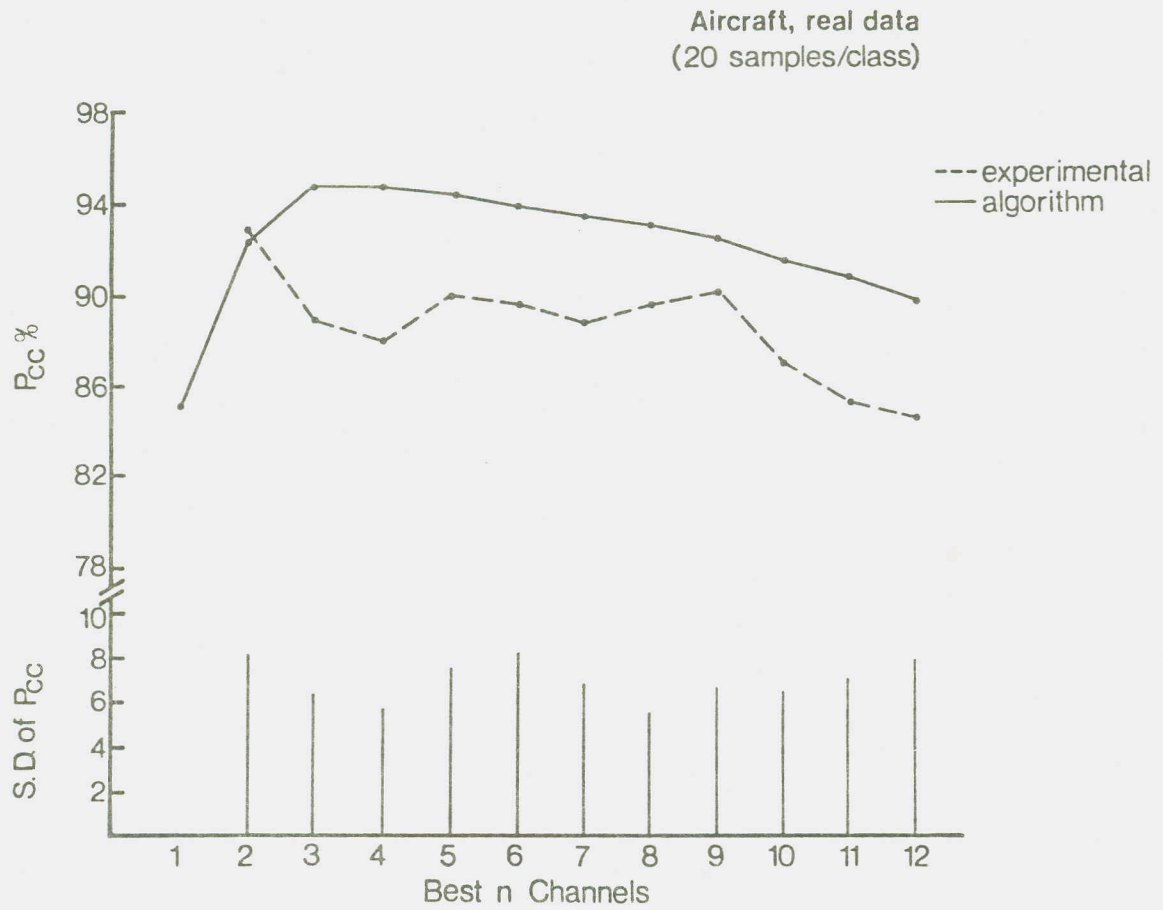


Figure 4.12 Classification Results of Aircraft, Real Data, Using 20 Samples per Class.

compare this curve with Figure 4.10, where the same conditions exist with the exception that the data is simulated. Because simulated data satisfies the assumptions made about the distributions of classes, it produces results that conform more with theory than real data does. The algorithm performance appears to be closer to what is expected, although in this case it does not quite follow the experimental curve. This "randomness" of the experimental curve is made more evident from looking at the standard deviations of P_{cc}, which do not seem to follow any general pattern and are all relatively large. This is a clear example of a case where deviations from the assumptions may obscure the action of a new proposed algorithm.

Experiment 4.12 (Aircraft, Real Data, 13 Samples per Class)

The same example used in Experiment 4.11 is used here, with 13 samples per class for training. Results are shown in Figure 4.13.

Experimental and algorithm results here are very close. Both peak at 3, and both are very close at high dimensionalities. The standard deviations of the errors are also increasing in general, particularly at high dimensionality. It is interesting to note that the standard deviation in almost all of the above four experiments starts increasing

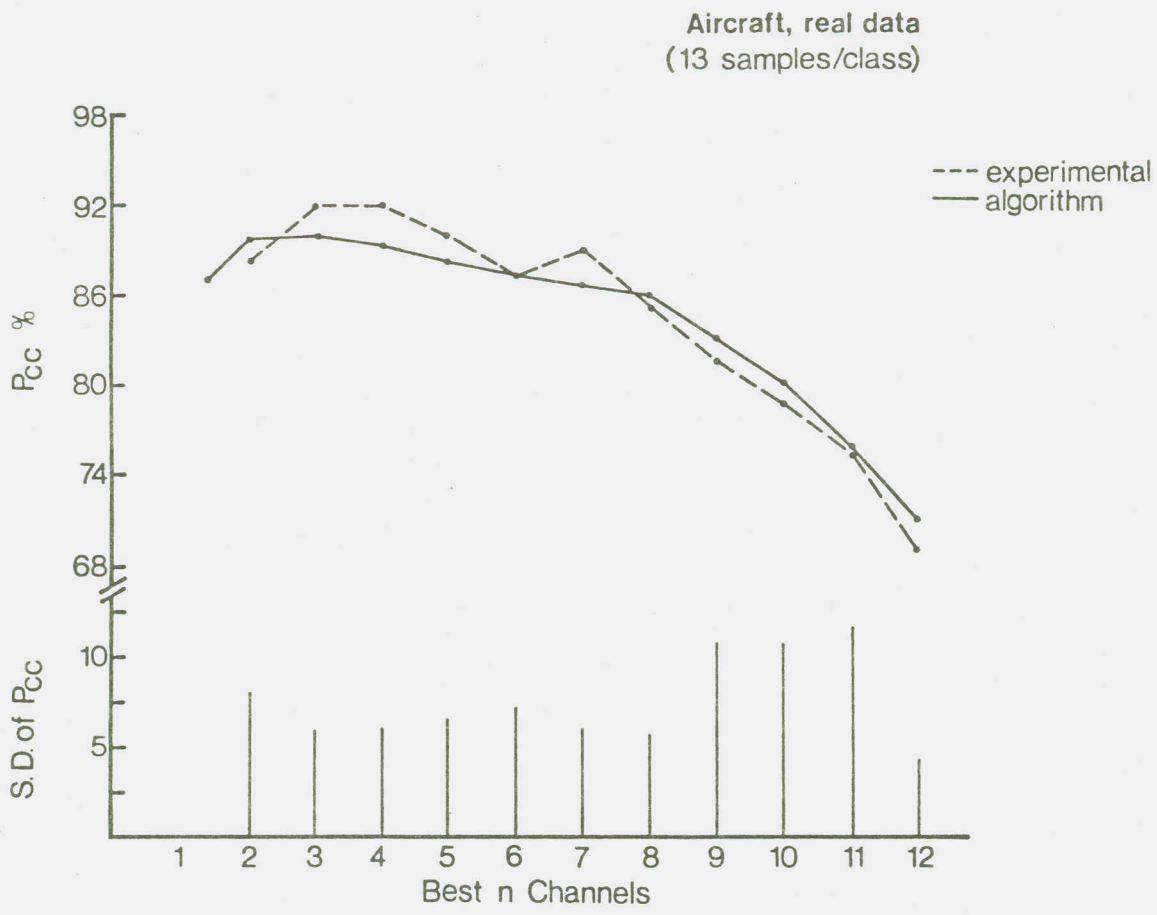


Figure 4.13 Classification Results of Aircraft, Real Data, Using 13 Samples per Class.

notably at about the same place the probability of correct classification starts dropping sharply. This supports the idea that at these dimensionalities, the randomness in the estimated statistics is so large that it pulls the curve down.

Experiment 4.13 (Landsat, Multitemporal, Simulated Data, 20 Samples per Class)

The data set used in this experiment is the same as that used in Experiment 4.6. It is obtained from Landsat, with four dates concatenated so that more features are presented. The 20 samples per class curve of Figure 4.6 is reproduced in Figure 4.14, together with the curve predicted by the algorithm.

The algorithm curve seems to drop in accuracy faster than the experimental curve, but both peak at around 4. The standard deviation of error also increases as more features are used.

Experiment 4.14 (Landsat, Multitemporal, Simulated Data, 13 Samples per Class)

The same data set used in Experiment 4.13 is used, but with 13 samples per class for training. Results appear in

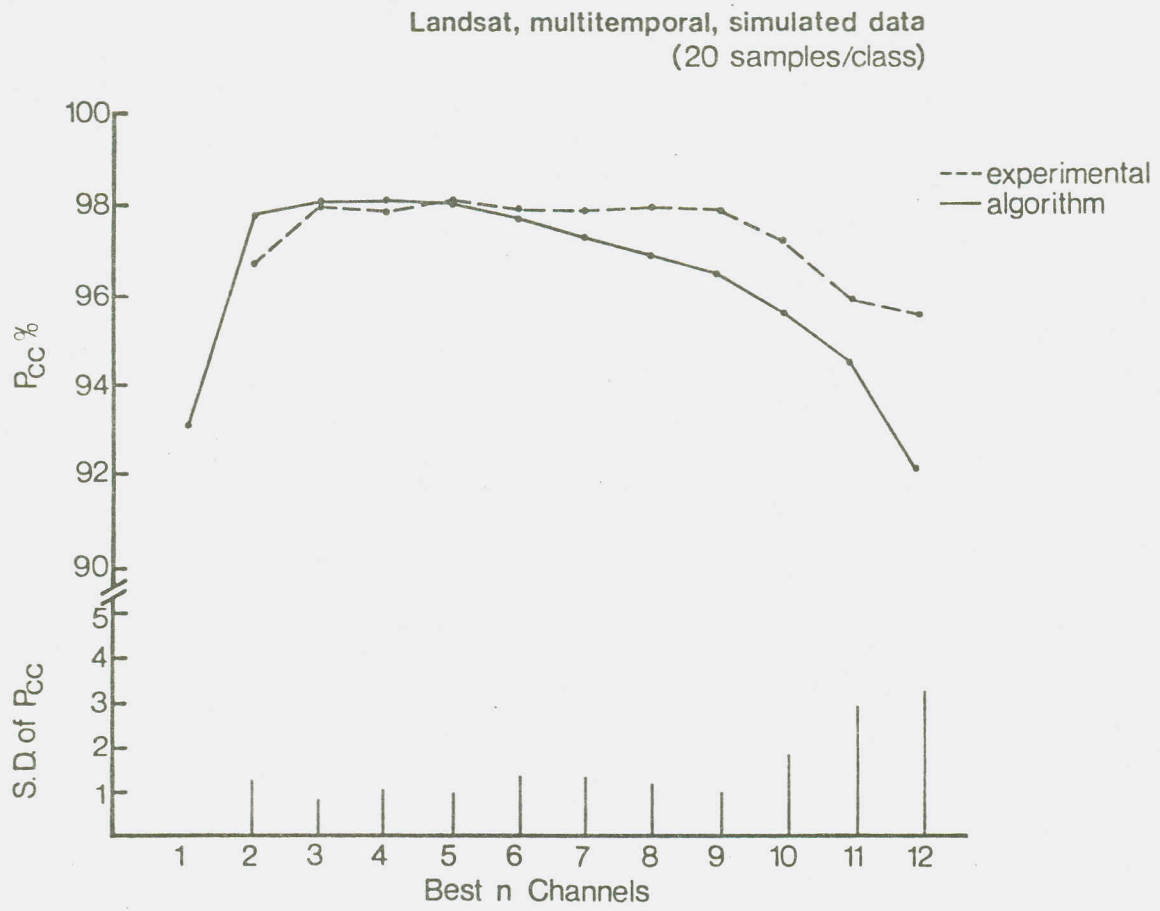


Figure 4.14 Classification Results of Landsat, Multitemporal, Simulated Data, Using 20 Samples per Class.

Figure 4.15. The increase in the variance of error with increasing dimensionality is very noticeable here. Again, the same observations apply, with both curves starting to drop in accuracy at the dimensionality of 4.

Experiment 4.15 (Landsat, Multitemporal, Real Data, 20 Samples per Class)

The Landsat data set is again used, but without any simulation. 20 samples per class are used for training, classification results are averaged and plotted in Figure 4.16.

While the algorithm predicts a somewhat better performance than the experimental curve, both have the same shape, and both are fairly constant until the first 7 or 8 features. This is due to the fact that the two classes in this set, corn and soybeans, are largely separable and hence the increase in the variance of the error with increasing dimensionality does not outweigh the large separability effect between these two classes.

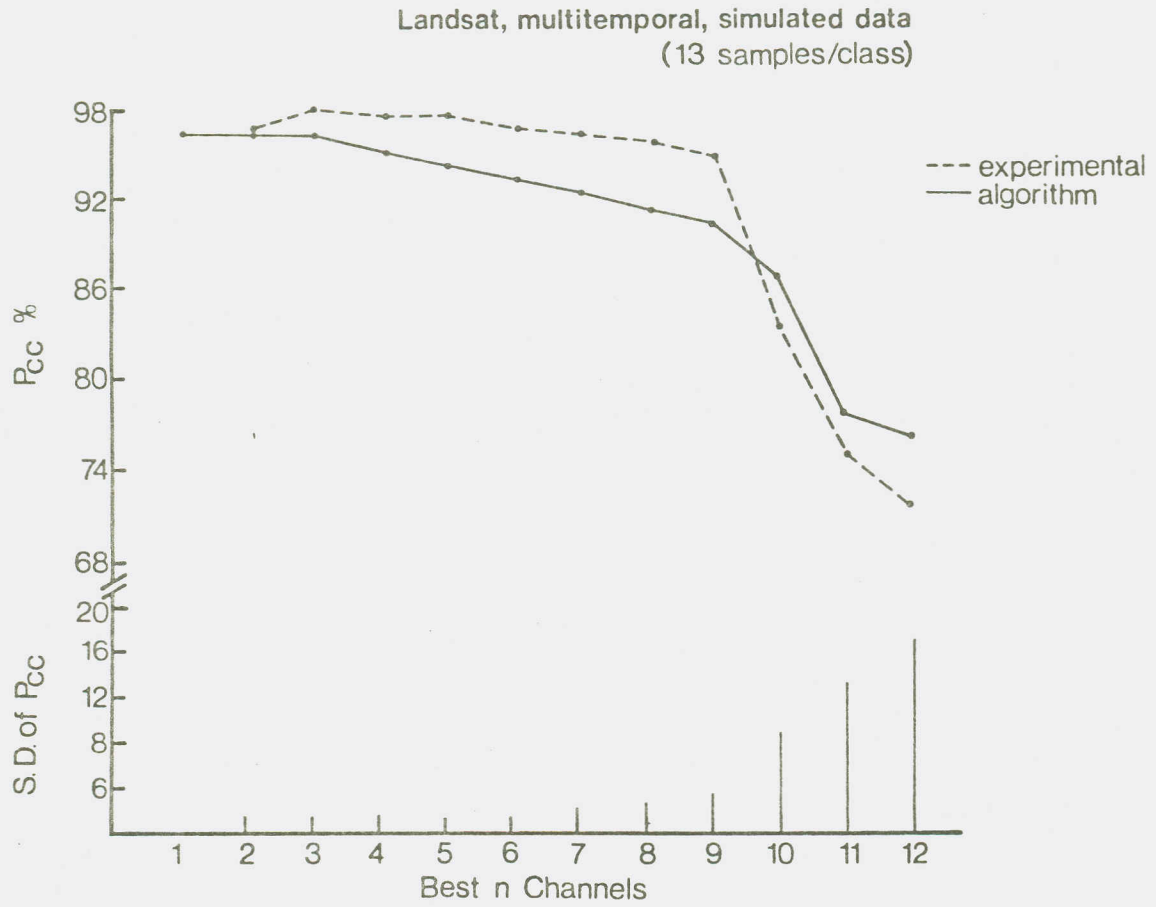


Figure 4.15 Classification Results of Landsat, Multitemporal, Simulated Data, Using 13 Samples per Class.

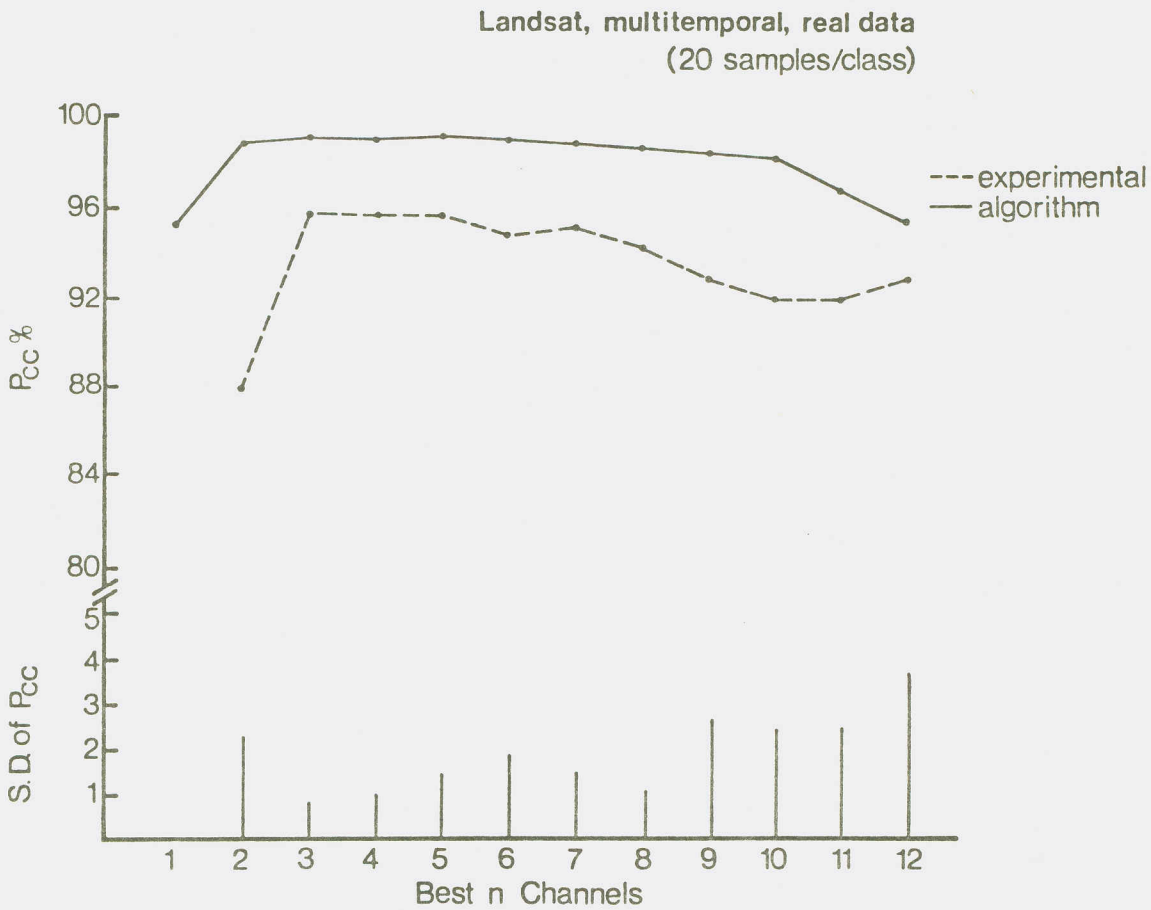


Figure 4.16 Classification Results of Landsat, Multitemporal, Real Data, Using 20 Samples per Class.

Experiment 4.16 (Landsat, Multitemporal, Real Data, 13 Samples per Class)

The Landsat, real data set is used in this experiment with 13 samples per class for training. Results are shown in Figure 4.17. The two curves have the same shape, and peak at the same place, 4, although again the algorithm predicts a better performance than does the experimental curve. The variance of error is again seen to be increasing with the number of features used.

To summarize the results of the last eight experiments (4.9-4.16), the probabilities of error predicted by the proposed algorithm as a function of dimensionality as compared to experimental observations are shown for aircraft and Landsat data. Results are obtained for both simulated and real data, using 20 and 13 samples per class for training. For each case, five different training sets are used, and classification results are averaged over these five sets. The standard deviations of errors for each feature subset are also plotted.

Results indicate that the algorithm predicts in most of the cases the best, or near best, subset of features to be used. While not always predicting closely the actual clas-

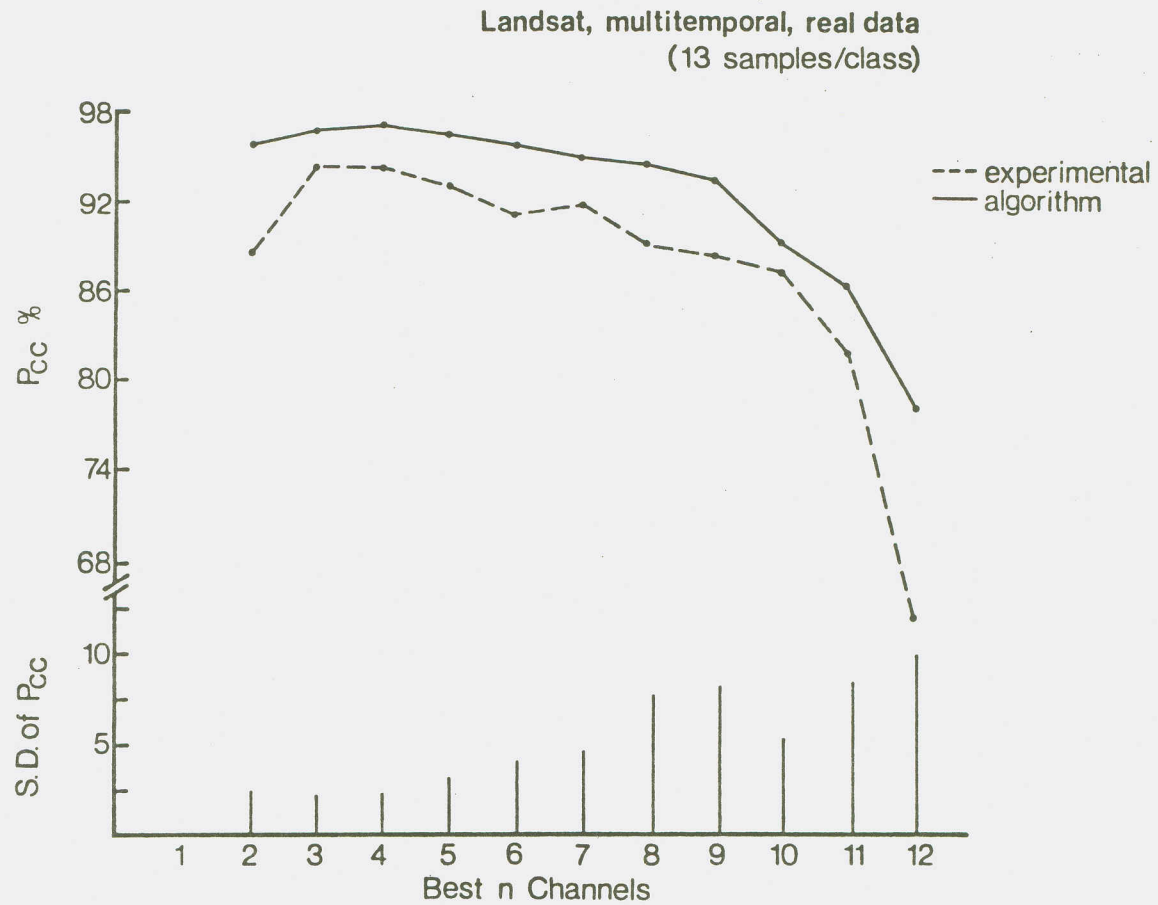


Figure 4.17 Classification Results of Landsat, Multitemporal, Real Data, Using 13 Samples per Class.

sification accuracies obtained from the experimental average curve, it has in most of the cases the same shape as the experimental curve and seems to follow any trends in performance that the experimental curve undertakes. Since the objective behind the algorithm is to predict the best feature dimensionality and specific subset to be used in classification rather than to predict the probability of error itself, the fact that the algorithm does not always accurately predict this probability of error is not of serious concern.

The standard deviations plotted seem to indicate that in general, an increase in dimensionality results in an increase in the variance of error, that increase becoming highly noticeable at high dimensionality, when the randomness in the estimated statistics, given a limited set of training samples, is large.

The next step is to incorporate this algorithm in a binary tree classification procedure, using more than two classes, and assess its performance. This is done in Section 4.5.

4.5 Experiments on a Binary Tree Classification Procedure

In this section, two data sets will be classified in a binary tree classification procedure, using the proposed algorithm to predict the optimal features at every node.

A complete design of a binary tree classification procedure should address the problem of how to separate the nodes in the tree effectively. Seprations should be sought that lead to meaningful classes at the intermediate and terminal nodes. This problem should be thoroughly studied before a solution can be arrived at.

It is not the purpose of this research to address this problem in any detail. Therefore, no attempt has been made here to dictate a particular procedure or claim any optimal, or close to optimal, one. The procedure that will be used is heuristic, the purpose of conducting the next two experiments is to illustrate the usefuleness of the proposed algorithm in predicting the optimal features to be used at every node. The problem of how to separate the nodes is left as a topic for future research.

Experiment 4.17

The Landsat, multitemporal, real data set used in Experiment 4.6 is used here again. Three informational

classes exist in the scene: corn, soybeans, and other. 13 samples per class are used for training, creating 3 spectral classes. The reason this is done is that in actual practice situations, it is almost impossible to distinguish spectral classes with only 13 training samples per class. A much larger, separate, set is used for testing (all training and test field descriptions are found in Appendix F). The binary tree is constructed by using a bottom-up procedure, combining the most separable classes. The criterion for measuring separability is that used by Whitsitt (9), and is defined as follows:

$$D_{erf} = \mathrm{erf}((2B)^{1/2}) \qquad (4.3)$$

where B is the Bhattacharyya distance and erf (.) is the gaussian error function. Whitsitt found that this measure is less ambiguous and more linear than the measure B. The measure is calculated using the first 12 features after a Karhunen-Loeve expansion was performed on the data. After the tree is constructed this way, the proposed algorithm was used to predict the optimal features to be used at every node.

The binary tree that resulted from the above procedure is shown in Figure 4.18. The algorithm predicts an optimal feature subset of 4 at the top, and a subset of 2 at the intermediate node. These appear below each node. Inside the node, the classes present are shown together with the total number of training samples present.

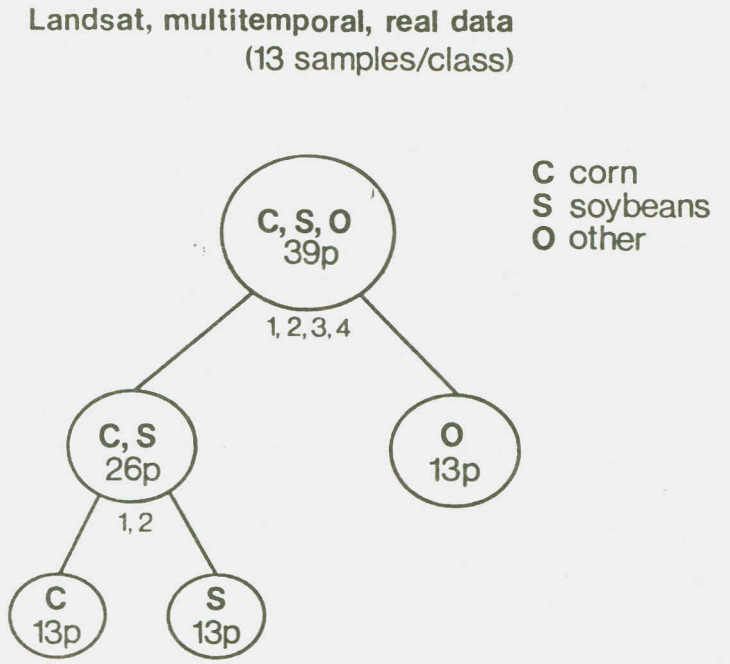


Figure 4.18 Binary Tree Design Structure of Landsat, Multitemporal, Real Data, Using 13 Training Samples per Class, With Numbers Inside Nodes Indicating Number of Training Samples Used.

A single-stage classification is then performed on the data using feature subsets of 2 to 12. This is done to compare the performance of the binary tree procedure to that of each of the feature subsets.

Results are plotted in Figure 4.19. The classification result obtained from the binary tree procedure is drawn in a dotted line across the page only to compare against the single-stage curve, and does not imply that all the feature subsets were used, or that the classification result is the same for all feature subsets.

The results indicate that using three classes, the single-stage curve has a peak at 4, and that by using all twelve features, the result is much poorer. The binary tree procedure, on the other hand, results in a classification accuracy that is almost as good as the best result obtained from using the best feature subset (which is unknown in an actual practice situation) in a single-stage classification. Thus, it appears that the algorithm is effective in predicting the best features to be used at each node.

Experiment 4.18

The aircraft, real data set used in Experiment 4.1 is used here. The data set has seven informational classes.

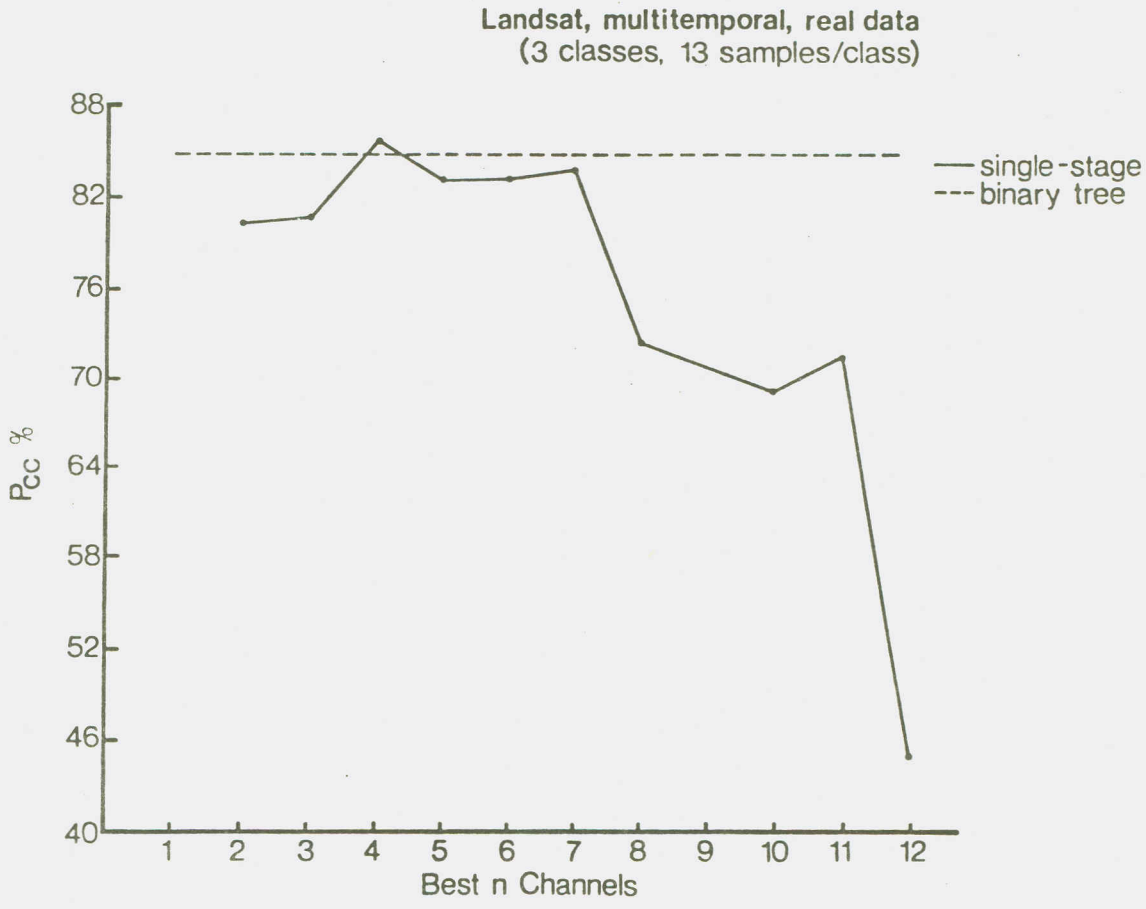


Figure 4.19 Single-Stage and Binary Tree Classification Results of Landsat, Multitemporal, Real Data, Using 13 Training Samples per Class.

In this experiment, supervised clustering (discussed in Section 1.2.1) is used to get 9 spectral classes, using an adequate number of training samples per class. 13 samples per class were then randomly chosen from the larger training set so that it is known that each set of these samples comes from one spectral class. The bottom-up procedure described in Experiment 4.17 was then used to build the binary tree, with the exception of class water, which was separated from the other classes at the beginning, as water has been known from experience to have spectral properties that are much different from other agricultural classes. The proposed algorithm is then used to predict the best features at each node. A single-stage classification is performed using feature subsets of 2 to 12, and then the same statistics were used in the binary tree classification procedure.

The resulting tree appears in Figure 4.20. Figure 4.21 shows the classification results obtained from the single-stage and the binary tree classifiers.

The binary tree procedure, using the proposed algorithm, performs better than any feature subset does in a single-stage procedure. The Hughes phenomenom is very evident here, as the overall classification accuracy for seven informational classes (9 spectral) drops sharply from a high of 69.4% to a low of 43.0%.

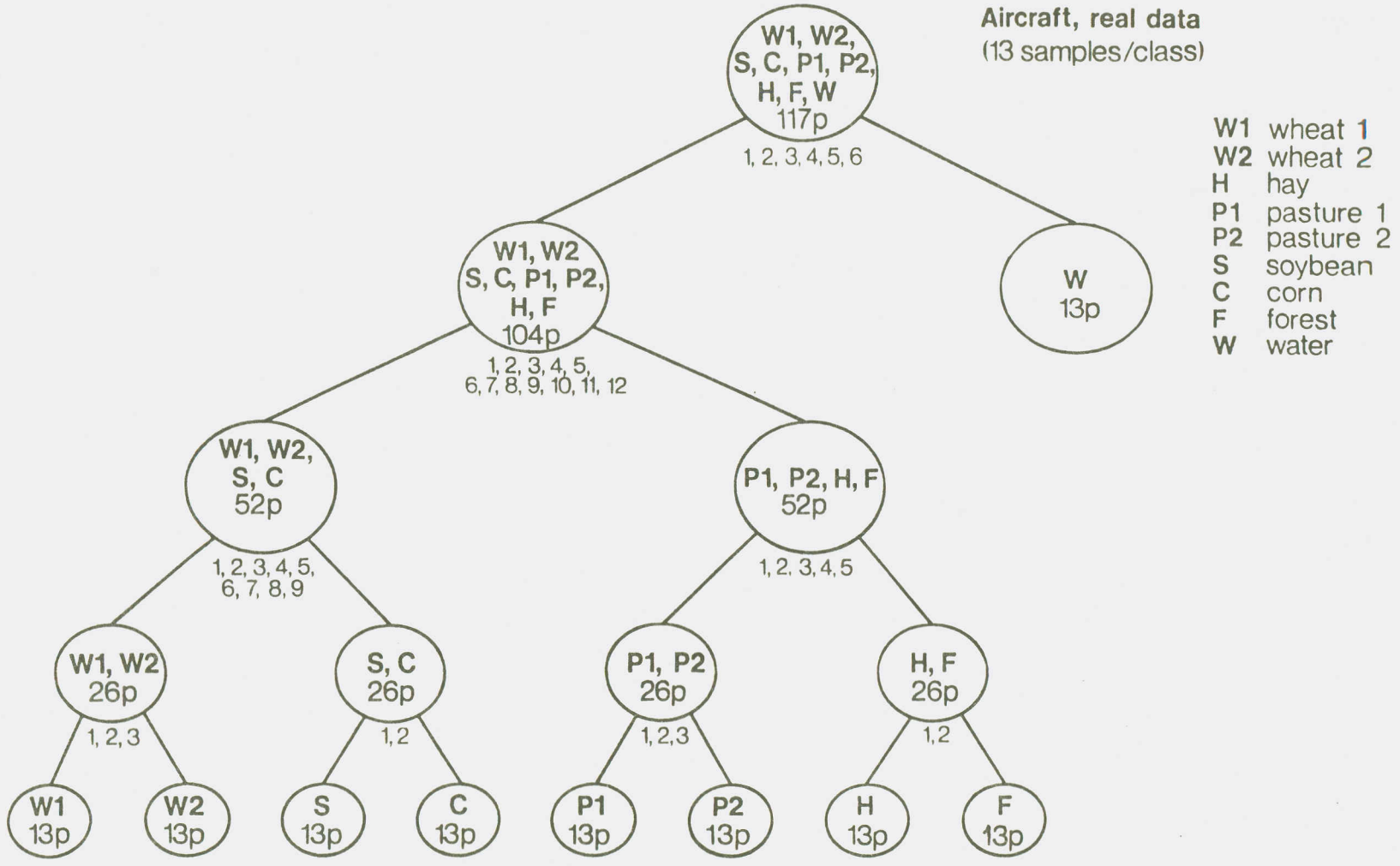


Figure 4.20 Binary Tree Design Structure of Aircraft, Real Data, Using 13 Training Samples per Class.

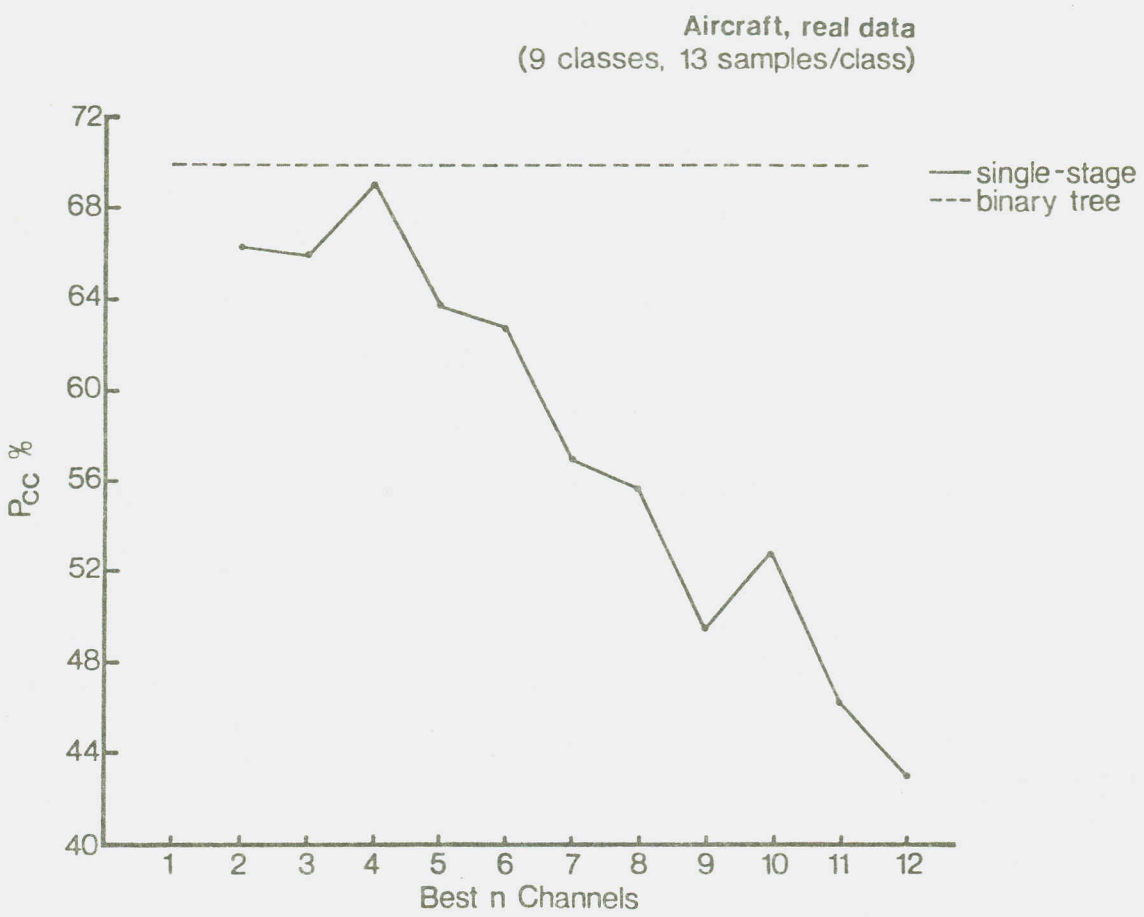


Figure 4.21 Single-Stage and Binary Tree Classification Results of Aircraft, Real Data, Using 13 Training Samples per Class.

Summarizing the results of the last two experiments, the proposed algorithm is shown to be effective in predicting feature subsets that lead to the maximum, or near maximum, accuracy possible using the Karhunen-Loeve expansion for ordering the features.

It is worthwhile to note that common belief is that few features need be used at the top of the tree to separate classes, and more features need be used deeper in the tree to distinguish between somewhat inseparable classes. However, if there are inadequate training samples present, then the number of training samples towards the bottom of the tree is less than that towards the top. Hence, less features should be used at the bottom to avoid the Hughes phenomenon. This is evident in the last two examples, particularly in Figure 4.20, where many features are used at the top, but only few at the bottom.

One point also worth mentioning is that in situations where a node is divided into two nodes of unequal training samples, one of them might have inadequate training samples while the other might have adequate ones. This situation is illustrated in Figure 4.20, where the top node is divided into water, and everything else. In this case, the number of features used is "intermediate", depending on the effect

of the degradation in the accuracy of the estimated statistics of the node with the inadequate number of training samples.

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 Summary of Results

The purpose of this research has been to develop an error estimator that will predict when/if the Hughes phenomenon occurs in multispectral data. Several significant results were arrived at and are summarized below.

The probability of error was studied through the likelihood ratio function, which offered the convenience of working with a one- dimensional variable, regardless of the number of features used in estimating the training statistics. An algorithm was then developed to estimate the statistics of this function, taking into account the number of training samples used to estimate these statistics. Several theoretical and experimental results were obtained on the Hughes phenomenon. These showed the dependency of the probability of error on the number of training samples and features used. The algorithm developed in Chapter 3 was shown to predict a suitable feature subset to be used at each node in a binary tree procedure. The algorithm was tested in

Chapter 4 by comparing it to experimental observations under different conditions, and was utilized in two binary tree classification procedures to demonstrate its practicality.

Some results were also shown, demonstrating the usefuleness of the K-L expansion over the whole data set in ordering features in the presence of a limited set of training samples. The procedure is used extensively in the research, and appears to have less variablity than other procedures under the conditions given.

Certain parts of the algorithm developed are heuristic in nature. Reasons why more theoretical solutions were not pursued were explained. These heuristic procedures often raise difficulty in verifying the validity of the algorithm strategy. The basic point is that when both a practical solution and theoretical perfection cannot be achieved simultaneously, one tends to choose the former. Experimental results in Chapter 4 demonstrated that the algorithm can be used practically to yield optimal, or near optimal, results.

5.2 Suggestions for Further Research

The main objective behind developing the error algorithm is to use it as a feature selection technique in a multi-stage classification procedure. In particular, the algorithm was developed to be used in a binary tree procedure. The design of such a procedure requires, in addition

to choosing the optimal features at each node, an effective design of separating the nodes. This question was only addressed superficially in this research, and could serve as a topic for another research project. An effective design for separating the nodes, coupled with the developed algorithm to choose the features, should lead to much higher accuracies than a single-stage classifier.

Several strategies developed in the research were heuristic in nature. Appendix B addresses the problem of why it is difficult to theoretically calculate the probability density function of the variances of the likelihood ratio function given either class one or two. If such a derivation is made possible, a much better and clearer idea will be obtained on how the variance of the likelihood ratio function is affected by the number of training samples, and the error algorithm can be made to more accurately predict the probability of error in the presence of a limited number of training samples.

The K-L expansion was used extensively as a feature selection technique in the presence of few training samples. This was based on experimental observations, but necessarily meant sacrificing the information found from the between classes variablity. A more detailed study of the relation of several feature selection techniques to the number of training samples can be very helpful.

LIST OF REFERENCES

LIST OF REFERENCES

1. Swain, P.H. and H. Hauska. 1977. The Decision Tree Classifier: Design and Potential. IEEE Trans. Geos. Elect. GE-15(3): 142-147.

2. Fukunaga, K. 1972. Introduction to Statistical Pattern Recognition. Academic Press, New York.

3. Duda, R.O. and P.E. Hart. 1973. Pattern Classification and Scene Analysis. Wiley, New York.

4. Fleming, M.D., J.S. Berkebile, and R.M. Hoffer. 1975. Computer-Aided Analysis of Landsat-1 MSS Data: A Comparison of Three Approaches, Including a 'Modified Clustering' Approach. 10p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 072475.

5. Fleming, M.D. and R.M. Hoffer. 1977. Computer-Aided Analysis Techniques for an Operational System to Map Forest Lands Utilizing Landsat MSS Data. 254p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 112277.

6. Fukunaga, K. and D.L. Kessell. 1973. Nonparametric Bayes Error Estimation Using Unclassified Samples. IEEE Trans. Infor. Theory. IT-19(7):434-400.

7. Mobasseri, B.G. and C.D. McGillem. 1979. Multiclass Bayes Error Estimation by a Feature Space Sampling Technique. IEEE Trans. Systems, Man and Cybernetics. SMC-9(10):660-665.

8. Moore, D.S., S.J. Whitsitt, and D.A. Landgrebe. 1976. Variance Comparisons of Unbiased Estimators of Probability of Correct Classification. IEEE Trans. Infor. Theory. IT-22(1):102-105.

9. Whitsitt, S.J. and D.A. Landgrebe. 1977. Error Estimation and Separability Measures in Feature Selection for Multiclass Pattern Recognition. 186p. Laboratory for Applications of Remote Sensing, Purdue University, West

Lafayette, Indiana. LARS Publication 082377. Also available as a Ph.D Thesis, TR-EE 77-34. Department of Electrical Engineering, Purdue University.

10. Wiersma, D.J. and D.A. Landgrebe. 1978. The Analytical Design of Spectral Measurements for Multispectral Remote Sensor Systems. 271p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Technical Report 122678. Also available as a Ph.D Thesis, TR-EE 79-13. Department of Elecrtical Engineering, Purdue University.

11. Mobasseri, B.G., D.J. Wiersma, E.R. Wiswell, D.A. Landgrebe, C.D. McGillem, and P.E. Anuta. 1978. A Multispectral Scanner System Parameter Study and Analysis Software System Description. 126p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Contract Report 112678.

12. Marill, T. and D.M. Green. 1963. On the Effectiveness of Receptors in Recognition Systems. IEEE Trans. Infor. Theory. IT-9(1):11-17.

13. Swain, P.H. and R.C. King. 1973. Two Effective Feature Selection Criteria for Multispectral Remote Sensing. 5p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 042673.

14. Jeffreys, H. 1948. Theory of Probability. Oxford University Press.

15. Matusita, K. 1951. On the Theory of Statistical Decision Functions. Ann. Institit. Stat. Math. (Tokyo). 3:17-35.

16. Bhattacharyya, A. 1943. On a Measure of Divergence Between Two Statistical Populations Defined by Their Probability Distributions. Bul. Calcutta Math. Soc. 35:99-109.

17. Mahalanobis, P.C. 1936. On the Generalized Distance in Statistics. Proc. National Inst. Sci. (India). 12:49-55.

18. Swain, P.H., T.V. Robertson, and A.G. Wacker. 1971. Comparison of the Divergence and B-Distance in Feature Selection. 12p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 020871.

19. Kailath, T. 1967. The Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Trans. Communication Technology. COM-14(1):52-60.

20. Karhunen, K. 1947. Uber Lineare Methoden in Der Wahrscheinlich- keitsrechnung. Amer. Acad. Sci., Fennicade. Ser. A,I, 37:3-79. (Transl: Rand Corp., Santa Monica, California, Rept. T-131, August 1960.)

21. Loeve, M. 1963. Probability Theory. Van Nostrand, Princeton, New Jersey.

22. Oja, E. and J. Karhunen 1980. Recursive Construction of Karhunen-Loeve Expansions For Pattern Recognition Purposes. Proceedings of Fifth International Conference on Pattern Recognition, Miami Beach, Florida, December 1-4. pp. 1215-1218.

23. Karhunen, J. and E. Oja 1980. Some Comments on the Subspace Methods of Classification. Proceedings of Fifth International Conference On Pattern Recognition, Miami Beach, Florida, December 1-4. pp. 1191-1194.

24. Kanal, L. 1976. Patterns in Pattern Recognition: 1968-1974. IEEE Trans. Infor. Theory. IT-20(6):697-722.

25. Hughes, G.F. 1968. On the Mean Accuracy of Statistical Pattern Recognizer. IEEE Trans. Infor. Theory. IT-14(1):55-63.

26. Abend, K. and T.J. Harley, Jr. 1969. Comments "On the Mean Accuracy of Statistical Pattern Recognizers." IEEE Trans. Infor. Theory (Correspondence). IT(5):420-421.

27. Chandrasekaran, B. and T.J. Harley, Jr. 1969. Comments "On the Mean Accuracy of Statistical Pattern Recognizer." IEEE Trans. Infor. Theory (Correspondence). IT(5):421-423.

28. Kanal, L. and B. Chandrasekaran. 1971. On Dimensionality and Sample Size in Statistical Pattern Classification. Pattern Recognition. 3:225-236.

29. Chandrasekaran, B. 1971. Independence of Measurements and the Mean Recognition Accuracy. IEEE Trans. Information Theory, Vol IT-7(4):452-456.

30. Foley, D.H. 1972. Considerations of Sample and Feature Size. IEEE Trans. Information Theory. IT-18(5):618-626.

31. Chandrasekaran, B. and A.K. Jain 1975. Independence, Measurement Complexity and Classification Performance. IEEE Trans. Systems, Man and Cybernatics. SMC-5(2):240-244

32. Duin, R.P. 1977. Comments on "Independence, Measurement Complexity, and Classification Performance". IEEE Trans. Systems Man and Cybernatics. (Correspondence) SMC(7):559-560.

33. Van Ness, J. 1977. Dimensionality and Classification Performance with Independent Coordinates. IEEE Trans. on Systems, Man and Cybernatics. (Correspondence) SMC(7):560-564.

34. Chandrasekaran, B. and A.K. Jain 1977. "Independence, Measurement Complexity and Classification Performance": An Ammendation. IEEE Trans. Systems, Man and Cybernatics. (Corrspondence) SMC(7):564-566.

35. Van Campenhout, J.M. 1978. On the Peaking of the Hughes Mean Recognition Accuracy: The Resolution of an Apparent Paradox. IEEE Trans. Systems, Man and Cybernetics. SMC-8(5):390-395.

36. Kulkarni, A.V. 1978. On the Mean Accuracy of Hierarchical Classifiers. IEEE Trans. Computers. C-27(8):771-776.

37. Raudys, S.J. 1979. Determination of Optimal Dimensionality in Statistical Pattern Classification. Pattern Recognition. 11:263-270.

38. Trunk, G.V. 1979. A Problem of Dimensionality: A Simple Example. IEEE Trans. Pattern Analysis and Machine Intelligence. PAMI-1:306-307.

39. Raudys, S. and V. Pikelis. 1980. On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithm in Pattern Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence. PAMI-2(3):242-252.

40. El-Sheikh, T.S. and A.G. Wacker. 1980. Effect of Dimensionality and Estimation on the Performance of Gaussian Classifiers. Pattern Recognition 12:115-126.

41. Wacker, A.G. and T.S. El-Sheikh 1980. Calculation of Probability of Correct Classification for Two-Class Gaussian Classifiers With Arbitrary Hyperquadratic Decision Boundaries. Machine Processing of Remotely Sensed Data Symposium. CHI533-9:94-302.

42. El-Sheikh,T.S. and A.G. Wacker 1980. Average Classification Accuracy Over Collections of Gaussian Problems. CHI1499-3:685-690.

43. Wald, A. 1947. Sequential Analysis. Wiley, New York.

44. Fu, K.S., Y.T. Chien and G.P. Cardillo. 1967. A Dynamic Programming Approach to Sequential Pattern Recognition. IEEE Trans. Computers. C-16(6):790-803.

45. Fu, K.S. 1968. Sequential Methods in Pattern Recognition and Machine Learning. Academic Press.

46. Dubes, R. and A.K. Jain. 1979. Validity Studies in Clustering Methodologies. Pattern Recognition. 11:235-254.

47. Lukasova, A. 1979. Hierarchical Agglomerative Clustering Procedure. Pattern Recognition. 11:365-381.

48. Nadler, M. 1971. Error and Reject Rates in a Hierarchical Pattern Recognizer. IEEE Trans. Computers. C-20:1598-1601.

49. Meisel, W.S. and D.A. Michalopoulos. 1973. A Partitioning Algorithm with Application in Pattern Classification and the Optimization of Decision Trees. IEEE Trans. Computers. C-22:93-103.

50. Wu, C.L., D.A. Landgrebe and P.H. Swain. 1974. The Decision Tree Approach to Classification. 194p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 090174. Also available as a Ph.D Thesis, TR-EE 75-17. Department of Electrical Engineering, Purdue University.

51. Swain, P.H., C.L. Wu, D.A. Landgrebe, and H. Hauska. 1975. Layered Classification Techniques for Remote Sensing Applications. 12p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 061275.

52. Bartolucci, L.A., P.H. Swain, and C.L. Wu. 1976. Selective Radiant Temperature Mapping Using a Layered Classifier. IEEE Trans. Geoscience Electronics. GE-14:101-106.

53. You, K.C. and K.S. Fu. 1976. An Approach to the Design of a Linear Binary Tree Classifier. Proc. Conference on Machine Processing of Remotely Sensed Data. June 29-July 1, 1976. IEEE Catalog No. 76CH1103-1 MPRSD.

54. Fletcher, R. and M.J.D. Powell. 1963. A Rapid Descent Method for Minimization. Computer Journal. 6:163-168.

55. Kulkarni, A.V. and L.N. Kanal. 1976. An Optimization Approach to Hierarchical Classifier Design. Proc. Third Int. Joint Conf. on Pattern Recognition (Coronado, California), IEEE Catalog No. 76CH/140-3C.

56. Parikh, J. 1977. A Comparative Study of Cloud Classification Techniques. Remote Sensing of Environment. 6:67-81.

57. Sethi, I.K. and B. Chatterjee. 1977. Efficient Decision Tree Design for Discrete Variable Pattern Recognition Problems. Pattern Recognition. 9:197-206.

58. Breiman, L. 1978. Growing Trees to Analyze High Dimensional Data. Technology Service Corporation Report. TSC-CSD-IN-024.

59. Sonquist, J.A., E.L. Baker, and J.N. Morgan. 1973. Searching for Structure. Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor, Michigan.

60. Rounds, E.M. 1979. A Combined Nonparametrical Approach to Feature Selection and Binary Decision Tree Design. Proc. IEEE Computer Society Conference on Pattern Recognition and Image Processing, Chicago, Illinois, August 6-8. CH1428-2:38-43.

61. Dattatreya, G.R., and V.V.S. Sarma, 1981. Bayesian and Decision Tree Approaches for Pattern Recoginition Including Feature Measurements Costs. IEEE Trans. Pattern Recoginition and Machine Intelligence. PAMI-3:293-298.

62. Wacker, A.G. and D.A. Landgrebe. 1971. The Minimum Distance Approach to Classification. 361p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Information Note 100771. Also available as a Ph.D Thesis, TR-EE 71-37, Department of Electrical Engineering, Purdue University.

63. Kullback, S. 1959. Information Theory and Statistics. p. 195. Wiley, New York.

64. Fukunaga, K. and T. Krile. 1969. Calculation of Bayes Recognition Error for Two Multivariate Gaussian Distributions. IEEE Trans. on Computers. C-18(3):220-229.

65. Swain, P.H. and S.M. Davis, eds. 1978. Remote Sensing: The Quantitative Approach. pp. 164-174. McGraw-Hill, Inc., New York.

66. Muasher, M. and P. Swain. 1980. A Multispectral Data Simulation Technique. 30p. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana. LARS Technical Report 070980.

67. Van Trees, H.L. 1968. Detection, Estimation and Modulation Theory, Part I. Wiley & Sons. New York.

68. Bickel, P.J. and K.A. Doksum 1977. Mathematical Statistics: Basic Ideas and Selected Topics. Holden-Day, San Francisco.

69. Papoulis, A. 1965. Probability, Random Variables, and Stochastic Processes. McGraw-Hill, New York.

APPENDICES

Appendix A

Generation of Normally Distributed Samples

Let U_1 and U_2 be two random variables independent and identically distributed Uniform (0,1). Then, let

$$Z_1 = (-2 \ln U_1)^{\frac{1}{2}} \cos 2\pi U_2 \qquad (A.1)$$

$$Z_2 = (-2 \ln U_1)^{\frac{1}{2}} \sin 2\pi U_2 \qquad (A.2)$$

then Z_1 and Z_2 are independent and identically distributed normal (0,1).

Proof:

$$f(U_1,U_2) = \begin{cases} 1 & 0 < U_1 < 1 \ , \quad 0 < U_2 < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (A.3)$$

is the probability density function of two independent uniforms.

$$U_1 = \exp\left[-\tfrac{1}{2}(Z_1^2 + Z_2^2)\right] \tag{A.4}$$

$$U_2 = \frac{1}{2\pi} \arctan\left(\frac{Z_2}{Z_1}\right) \tag{A.5}$$

The jacobian of the transformation is:

$$J = -\frac{1}{2\pi} \exp\left[-\tfrac{1}{2}(Z_1^2 + Z_2^2)\right]$$

$$f(Z_1,Z_2) = f(U_1,U_2) \cdot |J|$$

$$= \frac{1}{2\pi} \exp\left[-\tfrac{1}{2}(Z_1^2 + Z_2^2)\right] \qquad 0 < \exp\left[-\tfrac{1}{2}(Z_1^2 + Z_2^2)\right] < 1$$

$$0 < \frac{1}{2\pi} \arctan\left(\frac{Z_2}{Z_1}\right) < 1$$

$$= 0 \quad \text{otherwise} \tag{A.6}$$

$$f(Z_1) \sim N(0,1) \qquad f(Z_2) \sim N(0,1)$$

The side conditions give $-\infty < Z_1 < \infty$, $-\infty < Z_2 < \infty$. Strictly speaking, Z_1 cannot equal zero; however, prob(Z_1 = 0)=0 as we are working with continuous densities.

To test the effectiveness of the pseudo random vectors in the multivariate case, random vectors distributed $N(0,I_p)$ were generated and then tested with a Kolmogorov-Smirnov test. Since the multivariate normal cdf is difficult to evaluate, the sum of squares was calculated and compared to the χ_p^2 distribution.

For sample sizes greater than 100, the pseudo random vectors were distributed properly. For sample sizes less than 100, the K-S test is not valid. Since we would generally (over an entire area) be working with more than 100 points per class, this was not pursued further.

In addition, the sample covariance matrices were tested for homogeneity against the true class statistics. For sample runs of up to 2000 points, there were not significant differences at the $\alpha = 0.10$ level.

Appendix B

On The Probability Density Functions Of $\hat{\sigma}_1^2$ And $\hat{\sigma}_2^2$

Let us look at the expressions for $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$. From (3.55) and (3.58), we have:

$$\hat{\sigma}_1^2 = 2(\mathrm{tr}(I - \hat{\Sigma}_2^{-1}\hat{\Sigma}_1)^2 + 2\hat{m}_2\hat{\Sigma}_2^{-1}\hat{\Sigma}_1\hat{\Sigma}_2^{-1}\hat{m}_2) \qquad \text{(B.1)}$$

$$\hat{\sigma}_2^2 = 2(\mathrm{tr}(\hat{\Sigma}_1^{-1}\hat{\Sigma}_2 - I)^2 + 2\hat{m}_2\hat{\Sigma}_1^{-1}\hat{\Sigma}_2\hat{\Sigma}_1^{-1}\hat{m}_2) \qquad \text{(B.2)}$$

To be able to calculate the probability density functions of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$, one has to know those of $\hat{m}_2$, $\hat{\Sigma}_1$, $\hat{\Sigma}_1^{-1}$, $\hat{\Sigma}_2$, and $\hat{\Sigma}_2^{-1}$.

Before we proceed, we make the following assumptions:

1. $\hat{M}_1$ and $\hat{M}_2$, the means of the two classes at hand are constant. Experience has shown that one can estimate these two quantities relatively accurately with a small number of training samples. Henceforth, we will assume $\hat{m}_2$ $(=\hat{M}_1 - \hat{M}_2)$ to be constant and not a random variable.

2. $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are independent. We will ignore any relationships that might exist between the covariance matrices of the two classes.

Theorem B.1

$\hat{\Sigma}_1$, $\hat{\Sigma}_2$ are each Wishart distributed with parameters $\frac{1}{n_1}\Sigma_1$, n_1 and $\frac{1}{n_2}\Sigma_2$, n_2 respectively, where $n_i = N_i - 1$ and N_i is the number of samples used in estimating Σ_i.

Proof

See (B.1),pp.159.

Thus, $\hat{\Sigma}_i$, $i=1,2$, has the following Wishart distribution:

$$\hat{\Sigma}_i \sim \frac{(n_i)^{\frac{n_i}{2}} \left|\hat{\Sigma}_i\right|^{\frac{n_i-p-1}{2}} \exp(-\frac{1}{2}(n_i \operatorname{tr} \hat{\Sigma}_i^{-1} \hat{\Sigma}_i))}{2^{\frac{n_i p}{2}} \pi^{p(p-1)/4} \left|\Sigma_i\right|^{n_i/2} \prod_{k=1}^{p} \Gamma(\frac{1}{2}(n_i+1-k))} \qquad (B.3)$$

where p is the number of dimensions.

Theorem B.2

$\hat{\Sigma}_i^{-1}$ is again Wishart distributed with parameters $\frac{1}{n_i}\hat{\Sigma}_i^{-1}$, n_i.

Proof

See (B.2)

Theorem B.3

If A is distributed according to Wishart, $W(\Sigma, n)$, then $B = CAC^T$ is also distributed Wishart $W(\Phi, n)$, where $\Phi = C \Sigma C^T$.

Proof

See (B.1),pp.162.

From the above theorems, we see that $\hat{\Sigma}_1$, $\hat{\Sigma}_2$, $\hat{\Sigma}_1^{-1}$, and $\hat{\Sigma}_2^{-1}$ are Wishart distributed. Further, as $\hat{\Sigma}_1$ is transformed into the identity matrix I, and $\hat{\Sigma}_2$ is transformed into a diagonal matrix Λ, the new covariance matrices are also Wishart distributed. Hence, $\hat{\Sigma}_1$ is transformed into a diagonal matrix $\hat{I}$ that is distributed $W(1/n_1 I, n_1)$. We will call the diagonal elements of this matrix $\hat{\gamma}_i$. Similarly, $\hat{\Sigma}_2$ is transformed into a diagonal matrix $\hat{\Lambda}$, that is distributed $W(1/n_2 \Lambda, n_2)$. We will call the diagonal elements of this matrix $\hat{\lambda}_i$. $\hat{\Sigma}_1^{-1}$ is transformed into a diagonal matrix $\hat{I}^{-1}$ distributed $W(1/n_1 I, n_1)$, and $\hat{\Sigma}_2^{-1}$ is transformed into a diagonal matrix $\hat{\Lambda}^{-1}$ distributed $W(1/n_2 \Lambda, n_2)$.

Thus, after applying the simultaneous diagonalization transformation, $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ become:

$$\hat{\sigma}_1^2 = 2 \sum_{i=1}^{p} \left(1 - 2\frac{\hat{\gamma}_i}{\hat{\lambda}_i} + \frac{\hat{\gamma}_i^2}{\hat{\lambda}_i^2} + 2 d_i^2 \frac{\hat{\gamma}_i}{\hat{\lambda}_i^2}\right) \qquad (B.4)$$

$$\hat{\sigma}_2^2 = 2 \sum_{i=1}^{p} \left(\frac{\hat{\lambda}_i^2}{\hat{\gamma}_i^2} - 2\frac{\hat{\lambda}_i}{\hat{\gamma}_i} + 2 d_i^2 \frac{\hat{\lambda}_i}{\hat{\gamma}_i^2} + 1\right) \qquad (B.5)$$

Note that equations (B.4) and (B.5) are modified versions of equations (3.53) and (3.56).

We now look at the probability density functions of the one-dimensional elements $\hat{\lambda}_i$ and $\hat{\gamma}_i$.

Theorem B.4

If $\Sigma_{ij}=0$ for $i \neq j$, and if A is distributed according to $W(\Sigma, n)$, then $A_{11}, A_{22}, \ldots, A_{pp}$ are independently distributed and A_{jj} is distributed according to $W(\Sigma_{jj}, n)$.

Proof

See (B.1), pp.163.

Therefore, $\hat{\lambda}_1, \ldots, \hat{\lambda}_p$ are each distributed $W(\frac{\lambda_i}{n_2}, n_2)$ and $\hat{\gamma}_1, \ldots, \hat{\gamma}_p$ are each distributed $W(1/n_1, n_1)$. Hence,

$$\hat{\gamma}_i \sim \begin{cases} \dfrac{\hat{\gamma}_i^{(n_1-2)/2} \exp(-\frac{1}{2} n_1 \hat{\gamma}_i) \; (n_1/2)^{n_1/2}}{\Gamma(n_1/2)} & \hat{\gamma}_i > 0 \\ 0 & \hat{\gamma}_i < 0 \end{cases} \qquad (B.6)$$

A similar expression exists for $\hat{\gamma}_i^{-1}$, with $\hat{\gamma}_i$ replaced by $\hat{\gamma}_i^{-1}$.

$$\hat{\lambda}_i \sim \begin{cases} \dfrac{\hat{\lambda}_i^{(n_2-2)/2} \exp\left(-\tfrac{1}{2} n_2 \hat{\lambda}_i/\lambda_i\right) \left(n_2/2\right)^{n_2/2}}{\Gamma\left(n_2/2\right) \lambda_i^{n_2/2}} & \hat{\lambda}_i > 0 \\ 0 & \hat{\lambda}_i < 0 \end{cases} \qquad (B.7)$$

A similar expression exists for $\hat{\lambda}_i^{-1}$, with $\hat{\lambda}_i$, λ_i replaced by $\hat{\lambda}_i^{-1}$, λ_i^{-1} .

Looking at equations (B.5) and (B.6), we see that even though we know the individual distributions of $\hat{\lambda}_i$ and $\hat{\gamma}_i$, the calculation of the densities of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ is still a very involved and difficult process. An attempt to arrive at these densities directly from those expressions is almost impossible. However, the moments of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ can be calculated.

Since calculating the moments of $\hat{\lambda}_i$ (and $\hat{\lambda}_i^{-1}$, $\hat{\gamma}_i$, $\hat{\gamma}_i^{-1}$) involves the evaluation of an integral of the type $\int_0^\infty t^n e^{-at} dt$, and since such an integral does indeed exist, the task of calculating any moment of $\hat{\lambda}_i$, $\hat{\lambda}_i^{-1}$, $\hat{\gamma}_i$, and $\hat{\gamma}_i^{-1}$ is a very easy one.

From any integration table book, we find:

$$\int_0^\infty t^n \exp(-at)\, dt = \frac{\Gamma(n+1)}{a^{n+1}} \qquad (n > -1,\ a > 0) \qquad (B.8)$$

Thus, if x is distributed W(x/n,n), then:

$$E(\hat{x}) = x$$

$$E(\hat{x}^2) = (1+2/n)\, x^2$$

$$E(\hat{x}^3) = (1+6/n + 8/n^2)\, x^3 \qquad (B.9)$$

$$E(\hat{x}^4) = (1+12/n +44/n^2 + 48/n^3)\, x^4$$

Since any moment of $\hat{\sigma}_1^2$ or $\hat{\sigma}_2^2$ is a function of the moments of $\hat{\lambda}_i$, $\hat{\lambda}_i^{-1}$, $\hat{\gamma}_i$, and $\hat{\gamma}_i^{-]}$, it is theoretically possible to calculate any moment of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$. Thus, it is theoretically possible to calculate the characteristic function of $\hat{\sigma}_1^2$ or $\hat{\sigma}_2^2$ uniquely from these moments.

Papoulis (B.3) provides a way to estimate the probability density function of a random variable once its characteristic function is known. However, the convergence properties of calculating the characteristic function from the moments of a random variable are very slow. A large number of moments would have to be calculated. Looking at equations (B.4) and (B.5), it is evident that beyond the first few moments, the derivation becomes quite a formidable task, and is very impractical.

Because of these difficulties encountered, it was decided to calculate only the variances of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ and heuristically incorporate them into the algorithm developed.

REFERENCES

B.1 Anderson, T.W. 1958. Introduction to Multivariate Statistical Analysis. Wiley, New York.

B.2 Keehn, D.G. 1965. A Note On Learning For Gaussian Properties. IEEE Trans. Information Theory. pp. 126-132.

B.3 Papoulis, A. 1962. The Fourrier Integral And Its Applications. McGraw-Hill, New York.

Appendix C

Derivation of the Variances of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$

We look first at $\hat{\sigma}_1^2$

From Appendix B, equation (B.4), we have

$$\hat{\sigma}_1^2 = 2 \sum_{i=1}^{p} \left[1 - 2\frac{\hat{\gamma}_i}{\hat{\lambda}_i} + \frac{\hat{\gamma}_i^2}{\hat{\lambda}_i^2} + 2\, d_i^2\, \frac{\hat{\gamma}_i^2}{\hat{\lambda}_i^2} \right] \tag{C.1}$$

Noting the assumption that the $\hat{\lambda}_i$'s are independent from the $\hat{\gamma}_i^2$, and taking the expected value of $\hat{\sigma}_1^2$, we get

$$E(\hat{\sigma}_1^2) = 2 \sum_{i=1}^{p} \left[1 - 2\frac{E(\hat{\gamma}_i)}{E(\hat{\lambda}_i)} + \frac{E(\hat{\gamma}_i^2)}{E(\hat{\lambda}_i^2)} + 2\, d_i^2\, \frac{E(\hat{\gamma}_i^2)}{E(\hat{\lambda}_i^2)} \right] \tag{C.2}$$

Making use of the expressions in (B.9), we get

$$E(\hat{\sigma}_1^2) = 2 \sum_{i=1}^{p} \left[1 - \frac{2}{\lambda_i} + (1+\frac{2}{n_1})(1+\frac{2}{n_2})\, \frac{1}{\lambda_i^2} + 2\, d_i^2 (1+\frac{2}{n_2})\, \frac{1}{\lambda_i^2} \right] \tag{C.3}$$

Now note that $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ are the summation of uncorrelated random variables. Since $\hat{\lambda}_i$'s are independent, $\hat{\gamma}_i$'s are independent, and each $\hat{\lambda}_i$ is independent from each $\hat{\gamma}_i$, then any function of $\hat{\lambda}_i$'s and $\hat{\gamma}_i$'s in one dimension is uncorrelated with any other function of $\hat{\lambda}_i$'s and $\hat{\gamma}_i$'s in another dimension. Hence, the variances of $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ consist of the sum of the variances in each dimension (See (69), p. 211) and

do not have any cross-product terms between dimensions. Therefore, in the following derivations, we will not attempt to derive any cross-product terms as they will cancel out in the end result.

$$[E(\hat{\sigma}_1^2)^2] = 4E\left(\sum_{i=1}^{p}\left(1 - 2\frac{\hat{\gamma}_i}{\hat{\lambda}_i} + \frac{\hat{\gamma}_i^2 + 2d_i^2\hat{\gamma}_i}{\hat{\lambda}_i^2}\right)\right)^2 + \text{cross-product terms}$$

$$= 4E\sum_{i=1}^{p}\left(1 - 4\frac{\hat{\gamma}_i}{\hat{\lambda}_i} + 2\left(\frac{\hat{\gamma}_i^2 + 2d_i^2\hat{\gamma}_i}{\hat{\lambda}_i^2}\right) + 4\frac{\hat{\gamma}_i^2}{\hat{\lambda}_i^2} - 4\frac{\hat{\gamma}_i^3}{\hat{\lambda}_i^3} - 8\frac{d_i^2\hat{\gamma}_i^2}{\hat{\lambda}_i^3} + \frac{\hat{\gamma}_i^4 + 4d_i^2\hat{\gamma}_i^3 + 4d_i^4\hat{\gamma}_i^2}{\hat{\lambda}_i^4}\right) + \text{cross-product terms} \qquad (C.4)$$

Substituting the expressions of (B.9) into (C.4), we get

$$[E(\hat{\sigma}_1^2)^2] = 4\sum_{i=1}^{p}\left(1 - \frac{4}{\lambda_i} + 2\left(\frac{(1+2/n_1) + 2d_i^2}{\lambda_i^2}\right)(1 + \frac{2}{n_2})\right.$$

$$+ \frac{4}{\lambda_i^2}(1+\frac{2}{n_1})(1+\frac{2}{n_2}) - \frac{4}{\lambda_i^3}(1 + \frac{6}{n_1} + \frac{8}{n_1^2})(1 + \frac{6}{n_2} + \frac{8}{n_2^2})$$

$$- \frac{8d_i^2}{\lambda_i^3}(1+\frac{2}{n_1})(1 + \frac{6}{n_2} + \frac{8}{n_2^2}) + \frac{1}{\lambda_i^4}(1 + \frac{12}{n_2} + \frac{44}{n_2^2} + \frac{48}{n_2^3})$$

$$\left.\left((1 + \frac{12}{n_1} + \frac{44}{n_1^2} + \frac{48}{n_1^3}) + 4d_i^2(1 + \frac{6}{n_1} + \frac{8}{n_1^2}) + 4d_i^4(1 + \frac{2}{n_1})\right)\right)$$

$$+ \text{cross-product terms}$$

$$= 4\sum_{i=1}^{p}\left(1 - \frac{4}{\lambda_i} + \frac{2}{\lambda_i^2}\left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2} + 2d_i^2 + 4\frac{d_i^2}{n_2}\right)\right.$$

$$+ \left(4 + \frac{8}{n_1} + \frac{8}{n_2} + \frac{16}{n_1 n_2}\right)\frac{1}{\lambda_i^2} - \frac{4}{\lambda_i^3}\left(1 + \frac{6}{n_1} + \frac{6}{n_2} + \frac{8}{n_1^2} + \frac{8}{n_2^2} + \frac{36}{n_1 n_2}\right.$$

$$\left. + \frac{48}{n_1 n_2^2} + \frac{48}{n_1^2 n_2} + \frac{64}{n_1^2 n_2^2}\right) - 8\frac{d_i^2}{\lambda_i^3}\left(1 + \frac{2}{n_1} + \frac{6}{n_2} + \frac{12}{n_1 n_2} + \frac{8}{n_2^2} + \frac{16}{n_1 n_2^2}\right)$$

$$+ \frac{1}{\lambda_i^4}\left(1 + \frac{12}{n_1} + \frac{12}{n_2} + \frac{144}{n_1 n_2} + \frac{44}{n_1^2} + \frac{44}{n_2^2} + \frac{48}{n_1^3} + \frac{48}{n_2^3} + \frac{528}{n_1^2 n_2} + \frac{528}{n_2^2 n_1}\right.$$

$$+ \frac{1936}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_2^3 n_1} + \frac{2112}{n_1^3 n_2^2} + \frac{2112}{n_1^2 n_2^3} + \frac{2304}{n_1^3 n_2^3} + 4d_i^2\left(1 + \frac{6}{n_1} + \frac{12}{n_2}\right.$$

$$\left. + \frac{8}{n_1^2} + \frac{44}{n_2^2} + \frac{72}{n_1 n_2} + \frac{264}{n_1 n_2^2} + \frac{96}{n_1^2 n_2} + \frac{48}{n_2^3} + \frac{288}{n_1 n_2^3} + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_1^2 n_2^3}\right)$$

$$\left.\left. + 4d_i^4\left(1 + \frac{2}{n_1} + \frac{12}{n_2} + \frac{44}{n_2^2} + \frac{24}{n_1 n_2} + \frac{48}{n_2^3} + \frac{88}{n_1 n_2^2} + \frac{96}{n_1 n_2^3}\right)\right)\right)$$

$$+ \begin{matrix}\text{cross-product} \\ \text{terms}\end{matrix} \qquad \text{(C.5)}$$

$$[E(\hat{\sigma}_1^2)]^2 = 4\sum_{i=1}^{p}\left(1 - \frac{2}{\lambda_i} + \left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2}\right)\frac{1}{\lambda_i^2}\right.$$

$$\left. + \frac{2d_i^2}{\lambda_i^2}\left(1 + \frac{2}{n_2}\right)\right)^2 + \begin{matrix}\text{cross-product} \\ \text{terms}\end{matrix}$$

$$= 4\sum_{i=1}^{p}\left(1 - \frac{4}{\lambda_i} + \frac{2}{\lambda_i^2}\left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2} + 2d_i^2 + \frac{4d_i^2}{n_2} + 2\right)\right.$$

$$- \frac{4}{\lambda_i^3}\left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2} + 2d_i^2\left(1 + \frac{2}{n_2}\right)\right) + \frac{1}{\lambda_i^4}\left(1 + \frac{4}{n_1} + \frac{4}{n_2}\right.$$

$$+ \frac{4}{n_1^2} + \frac{4}{n_2^2} + \frac{16}{n_1 n_2} + \frac{16}{n_1^2 n_2} + \frac{16}{n_1 n_2^2} + \frac{16}{n_1^2 n_2^2} + 4d_i^2\left(1 + \frac{2}{n_1} + \frac{4}{n_2}\right.$$

$$\left.\left.\left. + \frac{8}{n_1 n_2} + \frac{4}{n_2^2} + \frac{8}{n_1 n_2^2}\right) + 4d_i^4\left(1 + \frac{4}{n_2} + \frac{4}{n_2^2}\right)\right)\right) + \text{cross-product terms} \qquad (C.6)$$

Now, $\text{Var}(\hat{\sigma}_1^2) = [E(\hat{\sigma}_1^2)^2] - [E(\hat{\sigma}_1^2)]^2$

or,

$$\text{Var}(\hat{\sigma}_1^2) = 4 \sum_{i=1}^{p} \left(\frac{2}{\lambda_i^2}\left(\frac{4}{n_1} + \frac{4}{n_2} + \frac{8}{n_1 n_2}\right) - \frac{4}{\lambda_i^3}\left(\frac{4}{n_1} + \frac{4}{n_2} + \frac{8}{n_1^2}\right.\right.$$

$$+ \frac{8}{n_2^2} + \frac{32}{n_1 n_2} + \frac{48}{n_1 n_2^2} + \frac{48}{n_1^2 n_2} + \frac{64}{n_1^2 n_2^2} + \frac{4d_i^2}{n_1} + \frac{8d_i^2}{n_2} + \frac{24d_i^2}{n_1 n_2} + \frac{16d_i^2}{n_2^2}$$

$$\left. + \frac{32d_i^2}{n_1 n_2^2}\right) + \frac{1}{\lambda_i^4}\left(\frac{8}{n_1} + \frac{8}{n_2} + \frac{128}{n_1 n_2} + \frac{40}{n_1^2} + \frac{40}{n_2^2} + \frac{48}{n_1^3} + \frac{48}{n_2^3} + \frac{512}{n_1^2 n_2}\right.$$

$$+ \frac{1920}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_2^3 n_1} + \frac{2112}{n_1^2 n_2^3} + \frac{2112}{n_1^3 n_2^2} + \frac{2304}{n_1^3 n_2^3} + 4d_i^2\left(\frac{4}{n_1} + \frac{8}{n_2}\right.$$

$$+ \frac{8}{n_1^2} + \frac{40}{n_2^2} + \frac{64}{n_1 n_2} + \frac{256}{n_1 n_2^2} + \frac{96}{n_1^2 n_2} + \frac{48}{n_2^3} + \frac{288}{n_1 n_2^3} + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_1^2 n_2^3}\Big)$$

$$\left.\left. + 4d_i^4\left(\frac{2}{n_1} + \frac{8}{n_2} + \frac{40}{n_2^2} + \frac{24}{n_1 n_2} + \frac{48}{n_2^3} + \frac{88}{n_1 n_2^2} + \frac{96}{n_1 n_2^3}\right)\right)\right) \qquad (C.7)$$

Next, we look at $\hat{\sigma}_2^2$

From Appendix B, equation (B.5) we have

$$\hat{\sigma}_2^2 = 2 \sum_{i=1}^{p} \left[\frac{\hat{\lambda}_i^2}{\hat{\gamma}_i^2} - 2\frac{\hat{\lambda}_i}{\hat{\gamma}_i} + 2\frac{d_i^2\hat{\lambda}_i}{\hat{\gamma}_i^2} + 1 \right] \tag{C.8}$$

$$E(\hat{\sigma}_2^2) = 2 \sum_{i=1}^{p} \left[\frac{E(\hat{\lambda}_i^2)}{E(\hat{\gamma}_i^2)} - 2\frac{E(\hat{\lambda}_i)}{E(\hat{\gamma}_i)} + \frac{2d_i^2 E(\hat{\lambda}_i)}{E(\hat{\gamma}_i^2)} + 1 \right]$$

$$= 2 \sum_{i=1}^{p} \left[(1+\frac{2}{n_1})(1+\frac{2}{n_2})\,\lambda_i^2 - 2\lambda_i + 1 + 2d_i^2(1+\frac{2}{n_2})\lambda_i \right] \tag{C.9}$$

$$[E(\hat{\sigma}_2^2)^2] = 4E \sum_{i=1}^{p} \left(\frac{\hat{\lambda}_i^2}{\hat{\gamma}_i^2} - \frac{2\hat{\lambda}_i}{\hat{\gamma}_i} + 1 + 2d_i^2\frac{\hat{\lambda}_i}{\hat{\gamma}_i^2} \right)^2 + \text{cross-product terms}$$

$$= 4E \sum_{i=1}^{p} \left(\frac{\hat{\lambda}_i^4}{\hat{\gamma}_i^4} + 4\lambda_i^3 \left(\frac{d_i^2}{\hat{\gamma}_i^4} - \frac{1}{\hat{\gamma}_i^3} \right) + 2\hat{\lambda}_i^2 \left(\frac{3}{\hat{\gamma}_i^2} - \frac{4d_i^2}{\hat{\gamma}_i^3} + \frac{2d_i^4}{\hat{\gamma}_i^4} \right) \right.$$

$$\left. + 4\hat{\lambda}_i \left(\frac{d_i^2}{\hat{\gamma}_i^2} - \frac{1}{\hat{\gamma}_i} \right) + 1 \right) + \text{cross-product terms}$$

$$= 4 \sum_{i=1}^{p} \left(\lambda_i^4 \left(1 + \frac{12}{n_1} + \frac{44}{n_1^2} + \frac{48}{n_1^3} \right) \left(1 + \frac{12}{n_2} + \frac{44}{n_2^2} + \frac{48}{n_2^3} \right) \right.$$

$$+ 4\lambda_i^3 \left(1 + \frac{6}{n_2} + \frac{8}{n_2^2} \right) \left(\left(1 + \frac{12}{n_1} + \frac{44}{n_1^2} + \frac{48}{n_1^3} \right) d_i^2 - \left(1 + \frac{6}{n_1} + \frac{8}{n_1^2} \right) \right)$$

$$+ 2\lambda_i^2 \left(1 + \frac{2}{n_2} \right) \left(3\left(1 + \frac{2}{n_1} \right) - 4d_i^2 \left(1 + \frac{6}{n_1} + \frac{8}{n_1^2} \right) + 2d_i^4 \left(1 + \frac{12}{n_1} \right.\right.$$

$$\left.\left. + \frac{44}{n_1^2} + \frac{48}{n_1^3} \right) \right) + 4\lambda_i \left(d_i^2 \left(1 + \frac{2}{n_1} \right) - 1 \right) + 1 \Bigg) + \text{cross-product terms}$$

$$= 4 \sum_{i=1}^{p} \Bigg[\lambda_i^4 \left(1 + \frac{12}{n_1} + \frac{12}{n_2} + \frac{144}{n_1 n_2} + \frac{44}{n_1^2} + \frac{44}{n_2^2} + \frac{48}{n_1^3} + \frac{48}{n_2^3} + \frac{528}{n_1^2 n_2} \right.$$

$$\left. + \frac{528}{n_2^2 n_1} + \frac{1936}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_2^3 n_1} + \frac{2112}{n_1^3 n_2^2} + \frac{2112}{n_1^2 n_2^3} + \frac{2304}{n_1^3 n_2^3} \right)$$

$$+ 4\lambda_i^3 \left(\left(1 + \frac{12}{n_1} + \frac{6}{n_2} + \frac{44}{n_1^2} + \frac{8}{n_2^2} + \frac{72}{n_1 n_2} + \frac{264}{n_1^2 n_2} + \frac{96}{n_2^2 n_1} + \frac{48}{n_1^3} + \frac{288}{n_2^3 n_1} \right.\right.$$

$$\left. + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_2^2 n_1^3} \right) d_i^2 - \left(1 + \frac{6}{n_1} + \frac{6}{n_2} + \frac{8}{n_1^2} + \frac{8}{n_2^2} + \frac{36}{n_1 n_2} + \frac{48}{n_1 n_2^2} + \frac{48}{n_1^2 n_2} \right.$$

$$\left.\left. + \frac{64}{n_1^2 n_2^2} \right) \right) + 2\lambda_i^2 \left(\left(3 + \frac{6}{n_1} + \frac{6}{n_2} + \frac{12}{n_1 n_2} \right) - 4d_i^2 \left(1 + \frac{2}{n_2} + \frac{6}{n_1} + \frac{12}{n_1 n_2} \right.\right.$$

$$\left.\left. + \frac{8}{n_1^2} + \frac{16}{n_1^2 n_2} \right) + 2d_i^4 \left(1 + \frac{2}{n_2} + \frac{12}{n_1} + \frac{44}{n_1^2} + \frac{24}{n_1 n_2} + \frac{48}{n_1^3} + \frac{88}{n_1^2 n_2} + \frac{96}{n_1^3 n_2} \right) \right)$$

$$+ 4\lambda_i \left(d_i^2 \left(1 + \frac{2}{n_1} \right) - 1 \right) + 1 \Bigg] + \text{cross-product terms} \qquad \text{(C.10)}$$

$$[E(\hat{\sigma}_1^2)]^2 = 4 \sum_{i=1}^{p} \Bigg[\left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2} \right) \lambda_i^2 - 2\lambda_i + 1$$

$$+ 2d_i^2 \left(1 + \frac{2}{n_1} \right) \lambda_i \Bigg]^2 + \text{cross-product terms}$$

$$= 4 \sum_{i=1}^{p} \Bigg[\left(1 + \frac{4}{n_1} + \frac{4}{n_2} + \frac{4}{n_1^2} + \frac{4}{n_2^2} + \frac{16}{n_1 n_2} + \frac{16}{n_1^2 n_2} + \frac{16}{n_1 n_2^2} + \frac{16}{n_1^2 n_2^2} \right) \lambda_i^4$$

$$+ 4\lambda_i^3 \left(d_i^2 \left(1 + \frac{4}{n_1} + \frac{2}{n_2} + \frac{8}{n_1 n_2} + \frac{4}{n_1^2} + \frac{8}{n_1^2 n_2} \right) - \left(1 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2} \right) \right)$$

$$+ 2\lambda_i^2\left(\left(3 + \frac{2}{n_1} + \frac{2}{n_2} + \frac{4}{n_1 n_2}\right) + 2d_i^4\left(1 + \frac{4}{n_1} + \frac{4}{n_1^2}\right) - 4d_i^2\left(1 + \frac{2}{n_1}\right)\right)$$

$$+ 4\lambda_i\left(d_i^2\left(1 + \frac{2}{n_1}\right) - 1\right) + 1\Bigg] + \text{cross-product terms} \qquad \text{(C.11)}$$

$$\text{Var}(\hat{\sigma}_2^2) = [E(\hat{\sigma}_2^2)^2] - [E(\hat{\sigma}_2^2)]^2 \qquad \text{or}$$

$$\text{Var}(\hat{\sigma}_2^2) = 4\sum_{i=1}^{p}\Bigg[\lambda_i^4\left(\frac{8}{n_1} + \frac{8}{n_2} + \frac{128}{n_1 n_2} + \frac{40}{n_1^2} + \frac{40}{n_2^2} + \frac{48}{n_1^3} + \frac{48}{n_2^3} + \frac{512}{n_1^2 n_2}\right.$$

$$\left. + \frac{512}{n_1 n_2^2} + \frac{1920}{n_1^2 n_2^2} + \frac{576}{n_1^3 n_2} + \frac{576}{n_1 n_2^3} + \frac{2112}{n_1^3 n_2^2} + \frac{2112}{n_1^2 n_2^3} + \frac{2304}{n_1^3 n_2^3}\right) + 4\lambda_i^3\left(d_i^2\left(\frac{8}{n_1}\right.\right.$$

$$\left. + \frac{4}{n_2} + \frac{8}{n_2^2} + \frac{40}{n_1^2} + \frac{64}{n_1 n_2} + \frac{256}{n_1^2 n_2} + \frac{96}{n_2^2 n_1} + \frac{48}{n_1^3} + \frac{288}{n_2^3 n_1} + \frac{352}{n_1^2 n_2^2} + \frac{384}{n_2^2 n_1^3}\right)$$

$$\left. - \left(\frac{4}{n_1} + \frac{4}{n_2} + \frac{8}{n_1^2} + \frac{8}{n_2^2} + \frac{32}{n_1 n_2} + \frac{48}{n_1 n_2^2} + \frac{48}{n_1^2 n_2} + \frac{64}{n_1^2 n_2^2}\right)\right) + 2\lambda_i^2\left(\left(\frac{4}{n_1}\right.\right.$$

$$\left. + \frac{4}{n_2} + \frac{8}{n_1 n_2}\right) + 2d_i^4\left(\frac{8}{n_1} + \frac{2}{n_2} + \frac{40}{n_1^2} + \frac{24}{n_1 n_2} + \frac{48}{n_1^3} + \frac{88}{n_1^2 n_2} + \frac{96}{n_1^3 n_2}\right)$$

$$\left. - 4d_i^2\left(\frac{2}{n_2} + \frac{4}{n_1} + \frac{12}{n_1 n_2} + \frac{8}{n_1^2} + \frac{16}{n_1^2 n_2}\right)\right)\Bigg] \qquad \text{(C.12)}$$

Because we do not know the true values of λ_i, we substitute for λ_i in equations (C.7) and (C.12) by $\hat{\lambda}_i$.

Appendix D

Classification Results Tables

Table D.1 Classification Results of Aircraft, Simulated Data, Using 20 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	85.4	82.9	81.1	81.0	79.5	82.0	92.4	2.26
1-3	95.9	95.5	92.4	93.5	93.0	94.0	94.0	1.39
1-4	92.5	95.7	91.6	94.0	93.5	93.4	93.4	1.56
1-5	90.8	96.5	90.7	96.8	94.2	93.8	93.4	3.00
1-6	89.5	96.4	89.3	97.2	93.4	93.2	93.2	3.71
1-7	90.3	96.7	86.5	96.1	94.0	92.7	93.2	4.29
1-8	89.7	96.2	86.4	97.4	95.4	93.0	93.0	4.74
1-9	90.1	95.2	86.5	97.9	95.9	93.1	92.0	4.69
1-10	89.4	96.0	87.5	97.5	95.7	93.2	91.7	4.46
1-11	88.0	95.7	84.1	94.3	96.1	91.6	90.0	5.33
1-12	87.0	94.9	83.7	94.0	94.5	90.8	87.0	5.14

Table D.2 Classification Results of Aircraft, Simulated Data, Using 13 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	76.3	77.9	84.2	75.1	86.1	79.9	90.5	4.92
1-3	93.3	94.7	94.3	91.5	93.1	93.4	90.5	1.25
1-4	92.3	95.2	94.8	93.1	88.5	92.8	89.7	2.67
1-5	92.8	96.3	93.2	95.2	89.8	93.5	89.2	2.52
1-6	90.1	95.0	91.1	94.4	91.4	92.4	88.5	2.16
1-7	91.0	94.0	93.0	89.2	85.0	90.4	87.3	3.56
1-8	88.0	87.6	80.7	83.9	80.9	84.2	86.3	3.51
1-9	89.0	87.4	75.4	88.1	85.5	85.1	83.6	5.56
1-10	74.6	82.8	76.1	79.3	69.9	76.5	79.0	4.87
1-11	67.9	66.1	65.2	69.4	75.0	76.5	73.3	3.87
1-12	70.8	65.1	58.4	69.8	69.8	66.8	68.4	5.18

Table D.3 Classification Results of Aircraft, Real Data, Using 20 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	79.0	98.1	97.8	96.1	94.3	93.1	90.4	8.00
1-3	81.5	97.1	86.2	86.9	93.0	89.0	95.0	6.13
1-4	80.0	97.6	81.5	87.9	93.7	88.1	95.0	5.49
1-5	84.6	97.8	80.3	92.9	95.3	90.2	94.8	7.39
1-6	86.4	96.6	77.0	92.8	95.4	89.6	94.2	8.09
1-7	87.9	94.6	78.0	90.2	93.7	88.9	93.7	6.65
1-8	88.8	94.1	80.3	90.7	94.9	89.8	93.3	5.26
1-9	87.2	96.3	80.7	91.2	95.8	90.2	92.6	6.50
1-10	82.6	96.7	80.7	89.0	86.7	87.1	91.6	6.27
1-11	79.5	96.3	79.4	86.8	84.9	85.4	91.0	6.93
1-12	77.1	96.5	78.0	87.0	84.9	84.7	90.0	7.86

Table D.4 Classification Results of Aircraft, Real Data, Using 13 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	95.8	83.4	89.2	78.6	97.6	88.9	90.3	8.06
1-3	90.0	94.5	98.3	83.7	95.5	92.4	90.4	5.70
1-4	90.1	97.6	98.8	84.3	90.8	92.3	89.7	5.94
1-5	91.2	95.0	98.5	84.7	83.3	90.6	88.7	6.52
1-6	93.3	96.5	81.9	87.5	79.6	87.8	87.8	7.21
1-7	94.6	96.1	83.0	88.7	83.8	89.2	87.3	6.01
1-8	83.0	94.7	83.8	87.2	79.9	85.7	86.3	5.65
1-9	64.7	89.4	91.9	85.7	78.8	82.1	83.5	10.91
1-10	62.7	81.6	92.1	83.4	75.6	79.1	80.4	10.90
1-11	62.7	67.9	94.4	79.3	74.8	75.8	76.3	12.12
1-12	66.8	65.1	68.0	70.0	76.1	69.2	71.0	4.25

Table D.5 Classification Results of Landsat, Multitemporal, Simulated Data, Using 20 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	97.4	96.1	97.0	97.9	95.2	96.7	97.8	1.08
1-3	98.7	97.4	98.7	97.9	97.5	98.0	98.1	0.63
1-4	98.7	96.5	98.7	97.9	97.9	97.9	98.1	0.90
1-5	99.0	97.2	98.9	97.6	97.7	98.1	98.0	0.82
1-6	99.0	96.0	98.9	98.1	97.4	97.9	97.8	1.24
1-7	98.9	96.0	98.9	98.2	97.6	97.9	97.3	1.20
1-8	98.7	96.3	98.9	98.2	97.7	98.0	96.9	1.04
1-9	98.7	96.6	98.6	98.2	97.6	97.9	96.5	0.87
1-10	98.7	94.3	98.5	97.1	97.5	97.2	95.6	1.76
1-11	96.0	91.4	98.6	95.5	98.1	95.9	94.5	2.85
1-12	95.6	90.3	98.1	96.1	97.9	95.6	92.1	3.16

Table D.6 Classification Results of Landsat, Multitempoal, Simulated Data, Using 13 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	98.0	97.0	97.4	95.8	97.7	97.2	96.5	0.86
1-3	99.0	98.7	96.9	98.2	98.7	98.3	96.5	0.83
1-4	98.0	98.4	96.3	98.0	98.7	97.9	95.6	0.93
1-5	97.4	98.6	97.4	97.9	98.7	98.0	94.7	0.63
1-6	97.1	97.6	96.2	97.5	98.6	97.4	94.0	0.87
1-7	97.5	96.7	93.6	98.1	98.7	96.9	92.7	2.00
1-8	96.0	96.7	93.1	97.2	98.9	96.4	91.8	2.12
1-9	90.6	96.1	93.3	97.2	98.9	95.2	90.8	3.29
1-10	90.4	84.0	82.4	91.3	69.6	83.5	87.0	8.71
1-11	68.1	63.7	77.0	97.1	70.5	75.3	78.0	13.11
1-12	53.4	60.1	72.3	97.7	76.7	72.0	76.7	17.10

Table D.7 Classification Results of Landsat, Multitemporal, Real Data, Using 20 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	88.3	85.9	91.3	86.0	87.9	87.9	99.0	2.20
1-3	94.7	96.1	95.8	96.6	96.2	95.9	99.2	0.72
1-4	94.3	96.1	96.0	95.8	96.7	95.8	99.1	0.89
1-5	93.5	96.2	95.8	97.0	96.6	95.8	99.3	1.37
1-6	94.0	92.3	95.4	97.0	95.7	94.9	99.1	1.79
1-7	95.4	94.3	93.7	97.1	96.1	95.3	99.0	1.36
1-8	94.0	95.7	93.6	94.8	93.6	94.3	98.8	0.91
1-9	88.8	95.6	94.1	94.0	91.9	92.9	98.4	2.63
1-10	88.3	91.3	93.6	94.3	91.9	91.9	98.2	2.34
1-11	89.2	89.7	93.9	94.5	92.6	92.0	97.0	2.42
1-12	86.7	94.7	93.9	95.8	93.2	92.9	95.5	3.58

Table D.8 Classification Results of Landsat, Multitemporal, Real Data, Using 13 samples per class.

Channels	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Average Exper.	Average Algorithm	S.D. of P_{cc} Exper.
1-2	87.5	85.7	91.1	89.7	90.6	88.9	96.3	2.27
1-3	96.0	96.1	96.6	91.8	93.6	94.8	97.0	2.05
1-4	96.1	95.6	97.0	92.3	93.2	94.8	97.6	2.00
1-5	96.6	94.7	95.7	90.7	89.7	93.5	97.0	3.09
1-6	93.1	94.9	90.0	85.4	94.0	91.5	96.3	3.87
1-7	92.2	94.9	94.5	84.4	94.5	92.1	95.4	4.43
1-8	93.0	85.7	96.1	77.8	94.9	89.5	95.0	7.69
1-9	93.0	87.2	95.3	75.0	93.5	88.8	94.0	8.30
1-10	91.8	87.9	87.7	79.5	92.4	87.9	89.6	5.15
1-11	72.1	84.4	79.7	80.1	95.0	82.3	86.7	8.39
1-12	50.3	72.2	58.4	74.2	67.1	64.4	78.3	10.00

Appendix E

Computer Program Listings

```
FILE: SWRITE   FORTRAN  A   LARS / PURDUE UNIVERSITY

C     ****************************************************************
C     WRITTEN BY: BILL PFAFF
C     EDITED BY: MARWAN MUASHER  JUNE 14, 1980
C     ****************************************************************
C
C
C     ****************************************************************
C          THIS PROGRAM GENERATES SIMULATED DATA BASED ON A
C     CLASSIFICATION MAP OR A GROUND TRUTH MAP.  EACH PIXEL
C     GENERATED THUS COMES FROM A KNOWN CLASS DISTRIBUTION.  THE
C     METHOD USED IS AS FOLLOWS:
C     1. A GOOD CLASSIFICATION IS CHOSEN AS A BASE FOR
C        SIMULATED DATA
C     2. FROM THIS CLASSIFICATION WE KNOW THE NUMBER OF CLASSES, THE
C        CLASS STATISTICS, AND THE CLASS OF EACH PIXEL IN THE
C        AREA CLASSIFIED.
C     3. A STREAM OF UNIFORM RANDOM NUMBERS IS GENERATED FOR
C        EACH CHANNEL.  THEY ARE CHANGED TO NORMAL (0,1) DEVIATES.
C     4. FOR EACH PIXEL, A RANDOM N(0,I)VECTOR IS TRANSFORMED TO
C        BE DISTRIBUTED ACCORDING TO THE CLASS STATISTICS OF THAT
C        PIXEL.  THIS IS THE SIMULATED DATA VECTOR.
C     5. AS EACH LINE IS COMPLETED, IT IS WRITTEN TO AN OUTPUT TAPE.
C          TO RUN THE PROGRAM, YOU NEED TO HAVE THE FOLLOWING
C     EXEC FILE ON YOUR DISK:
C
C     GETDISK LARSYS
C     GETDISK DVSYS
C     GLOBAL TXTLIB CMSLIB FORTRAN SSP370
C     FILEDEF 6 PRINTER
C     FILEDEF 16 TERMINAL
C     FILEDEF 12 TAP2
C     FILEDEF 11 TAP1 (RECFM VS LRECL 1500 BLKSIZE 1500)
C     LOAD SWRITE GLOCOM MMTAPE TAPOP BCDVAL GTSERL GTDATE MFSD
C     RANDU WRTMTX
C     START SWRITE
C
C          THE PROGRAM WILL ASK FOR INFORMATION SUCH AS
C     TAPE NUMBERS, FILE NUMBERS,  ..ETC.  FROM HERE ON, IT
C     SHOULD BE EASY TO FOLLOW.
C     ****************************************************************
C
C
C     ****************************************************************
C    VARIABLES USED IN TPRINT
C
C    A      =COVARIANCE STORAGE FOR FACTORING
C    AREANO=AREA NUMBER OF CLASSIFICATION
C    B      =COVARIANCE STORAGE FOR MULTIPLICATION
C    DATA   =DATA POINT STORAGE
C    DATVAL=LINE NUMBER AND ROLL PARAMETER
C    ICAL   =CALIBRATION INFORMATION
C    IDREC  =IDENTIFICATION RECORD STORAGE
C    ISTART=STARTING POINTS FOR GAUSS
C    LOGDAT=DATA POINTS IN LOGICAL FORMAT
C    NOCHAN=NUMBER OF CHANNELS IN CLASSIFICATION
C    NOCLAS=NUMBER OF CLASSES IN ORIGINAL STATISTICS
C    NOFLDS=NUMBER OF TEST FIELDS
C    NOPOOL=NUMBER OF POOLED CLASSES
C    PNTCLS=CLASSIFICATIONS ARRAY
C    Z      =STATISTICS STORAGE
C     ****************************************************************
C
C
C*****************************************************************
C  INITIALIZATION
C*****************************************************************
C
      INTEGER*2  I2, INTDAT, ICAL(3), ILIN(2), PNTCLS(1000), ISTAT(4),
     $           FETVC3(30)
      LOGICAL*1  L1(2), LOGDAT(2), LCAL(6), DATOUT(12000)
      REAL*4  A(720), A2(12), Z(2700), B(12,12), DATA(12),
     $        RMEAN(30,12), RVAR(30,12,12), ZZ(2700), FRQCAL(5,30)
      INTEGER*4  ISTART(12), EOS, INFO(17), AREANO, IDREC(200), TAPENO, THREE,
     $           CLAPNT(30), IMEAN(30,12), IVAR(30,12,12), YES, NO, DATE(3)
      INTEGER*4  RUNNO, FLGT
      EQUIVALENCE (I2,L1), (INTDAT,LOGDAT), (ICAL,LCAL), (LNWRT,ILIN)
      EQUIVALENCE(FRQCAL(1,1),IDREC(51))
      DATA EOS,S.AM /'EOS ',1.0,0.0 /
      DATA YES,NO,THREE /'YES ','NO  ','3'/
```

```
FILE: SWRITE   FORTRAN  A   LARS / PURDUE UNIVERSITY

      DATA  FLGT /'SIM '/
      EPS=1.E-5
C
C*************************************************************************
C LOAD TAPES AND READ PARAMETERS
C*************************************************************************
C
      WRITE(16,500)
  500 FORMAT(//5X,'SPECIFY TAPE NUMBER ON WHICH RESULTS FILE IS LOCATED
     $/5X,'(TYPE EIGHT DIGIT TAPE NUMBER)')
      READ(16,505)INTAP
  505 FORMAT(I8)
      WRITE(16,510)
  510 FORMAT(5X,'SPECIFY FILE NUMBER AT WHICH RESULTS FILE IS LOCATED'/
     $X,'(TYPE THREE DIGIT FILE NUMBER)')
      READ(16,515)IFILE
  515 FORMAT(I3)
      CALL MMTAPE(INTAP,IFILE,0)
      WRITE(16,570)
  570 FORMAT(//5X,'SPECIFY THE TAPE NUMBER ONTO WHICH SIMULATED DATA IS
     $TO BE WRITTEN'/5X,'(TYPE EIGHT DIGIT TAPE NUMBER)')
      READ(16,575)TAPENO
  575 FORMAT(I8)
      WRITE(16,580)
  580 FORMAT(5X,'SPECIFY FILE NUMBER AT WHICH SIMULATED DATA IS TO BE W
     $ITTEN'/5X,'(TYPE THREE DIGIT FILE NUMBER)')
      READ(16,585)JFILE
  585 FORMAT(I3)
      WRITE(16,590)
  590 FORMAT(//5X,'SPECIFY THE RUN NUMBER FOR THE SIMULATED DATA RUN'/
     1      5X,'(TYPE EIGHT DIGIT RUN NUMBER)')
      READ(16,575) RUNNO
      CALL MOUNT(TAPENO,12,'RI')
      MARG=JFILE-1
      IF(MARG.LE.0) GO TO 3
      DO 3 LIP=1,MARG
      CALL TOPFF(12)
    3 CONTINUE
    5 READ(11)I
      IF(I.NE.1) GO TO 310
      READ(11)I,J,NOCLAS,NOCHAN,NOFLDS,NOPOOL,(FETVC3(IX),IX=1,NOCHAN)
      NOCH=((NOCHAN+1)/2)*2
      NOCOMP=NOCHAN*(NOCHAN+1)/2
      ISTOP=NOCOMP*NOPOOL
      IEND=ISTOP+NOCHAN*NOPOOL
   15 READ(11)I,J,K
      IF(I.LT.3) GO TO 15
      IF(K.NE.EOS) GO TO 15
      READ(11)I,J,(Z(IX),IX=1,IEND)
      DO 17 IX=1,IEND
      Z2(IX)=Z(IX)
   17 CONTINUE
   45 READ(11)I,AREANO,NOPNTS,NOLINE,INFO,IDREC
      NOFET3=NOCHAN
      IF(I.NE.5) GO TO 45
      WRITE(6,520)
  520 FORMAT(1H1////5X,'+++++++++++++++++++++++++++++++++++++++++')
      WRITE(6,525)
  525 FORMAT(5X,'+DATA SIMULATION USING MCCABES EQUATION+')
      WRITE(6,530)
  530 FORMAT(5X,'+++++++++++++++++++++++++++++++++++++++++')
      WRITE(6,535) RUNNO, IDREC(3)
  535 FORMAT(////5X,'SIMULATED DATA RUN IS',I9,' FROM RUN',I9)
      WRITE(6,537)INFO(4),INFO(5),INFO(7),INFO(8)
  537 FORMAT(/5X,'LINE',I5,' TO LINE',I5,' AND COLUMN',I5,' TO COLUMN',
     $5)
      WRITE(6,540)INTAP,IFILE
  540 FORMAT(/5X,'INPUT RESULTS FILE IS ON TAPE',I9,' FILE',I4)
      WRITE(6,545)TAPENO,JFILE
  545 FORMAT(/5X,'SIMULATED DATA IS ON TAPE',I9,' FILE',I4)
      WRITE(6,550)
  550 FORMAT(/5X,'CHANNELS USED')
      DO 560 IX=1,NOCHAN
      WRITE(6,555)FETVC3(IX),FRQCAL(1,IX),FRQCAL(2,IX)
  555 FORMAT(5X,I2,2X,F5.2,'-',F5.2)
  560 CONTINUE
      CALL GTDATE(DATE)
      WRITE(6,565)DATE
  565 FORMAT(/5X,'DATE OF SIMULATION IS ',3A4)
```

```
FILE:  SWRITE   FORTRAN  A   LARS / PURDUE UNIVERSITY

      IBEG=1
C
C*********************************************************************
C FACTOR COVARIANCE MATRICES
C*********************************************************************
C
      DO 30 IX=1,NOPOOL
      IDONE=IBEG+NOCOMP-1
      K=0
      DO 20 IY=IBEG,IDONE
      K=K+1
   20 A(K)=Z(IY)
      CALL MFSD(A,NOCHAN,EPS,IER)
      IF(IER.EQ.-1) GO TO 300
      IF(IER.GE.1) GO TO 310
      K=0
      DO 25 IY=IBEG,IDONE
      K=K+1
   25 Z(IY)=A(K)
   30 IBEG=IBEG+NOCOMP
C
C*********************************************************************
C GENERATE STARTING POINTS
C*********************************************************************
C
   29 WRITE(16,31)
   31 FORMAT(5X,'DO YOU WANT TO SPECIFIY THE STARTING POINTS FOR THE'/5
     $,'RANDOM NUMBER GENERATOR? (TYPE YES OR NO)')
      READ(16,32)INPUT
   32 FORMAT(A4)
      IF(INPUT.EQ.NO) GO TO 36
      IF(INPUT.EQ.YES) GO TO 33
      GO TO 29
   33 DO 39 IX=1,NOCHAN
      WRITE(16,41)IX
   41 FORMAT(5X,'SPECIFY STARTING POINT FOR CHANNEL',I3/5X,'(TYPE A NIN
     $ DIGIT ODD NUMBER)')
      READ(16,42)ISTART(IX)
   42 FORMAT(I9)
   39 CONTINUE
      GO TO 43
   36 CALL GTSERL(ISERL)
      ISERL=(ISERL/10)*8+1
      DO 40 I=1,NOCH
      ISERL=ISERL+1000000
      ISTART(I)=ISERL
   40 CONTINUE
   43 WRITE(6,34)
   34 FORMAT(////5X,'STARTING POINTS FOR RANDOM NUMBER GENERATOR'//)
      DO 44 I=1,NOCHAN
      WRITE(6,35)I,ISTART(I)
   35 FORMAT(5X,'STARTING POINT FOR CHANNEL ',I2,' IS ',I9)
   44 CONTINUE
C
C*********************************************************************
C READ CLASSIFICATIONS
C*********************************************************************
C
      IDREC(1)=TAPENO
      IDREC(2)=JFILE
      IDREC(3) = RUNNO
      NOLD = IDREC(5)
      IDREC(5) = NOCHAN
      IDREC(6) = 4*((NOPNTS + 9)/4)
      NOSAM = IDREC(6)
      IDREC(7) = FLGT
      DO 141 II=1,3
      IDREC(II+16) = DATE(II)
  141 CONTINUE
      IDREC(20) = NOLINE
C     DO 145 II= 1, NOCHAN
C     INEW = FETVC3(II)
C     DO 145 II2 = 1,5
C     FRQCAL(II2,II) = FRQCAL(II2,INEW)
C 145 CONTINUE
C     LIP = NOCHAN + 1
C     DO 150 II = LIP,NOLD
C     DO 150 II2 = 1,5
C     FRQCAL(II2,II) = 0.0
```

```
FILE:  SWRITE   FORTRAN  A   LARS / PURDUE UNIVERSITY

C 150 CONTINUE
C
      CALL TOPWR(12,800,IER,IDREC)
      IF(IER.NE.0) WRITE(16,234)IER
      IF(IER.GT.0) GO TO 310
      DO 50 MA=1,NOCLAS
      CLAPNT(MA)=0
      DO 50 MB=1,NOCHAN
      IMEAN(MA,MB)=0
      RMEAN(MA,MB)=0.0
      DO 50 MC=1,NOCHAN
      IVAR(MA,MB,MC)=0
   50 RVAR(MA,MB,MC)=0.0
       LNWRT = 0
   55 READ(11)J,K,LINENO,(PNTCLS(IX),IX=1,NOPNTS)
      IF(J.GT.6) GO TO 95
      LNWRT=LNWRT+1
      IF (MOD(LNWRT,25).EQ.0) WRITE(16,57)LNWRT,NOLINE
   57 FORMAT(5X,I4,' LINES OUT OF ',I4,' ARE COMPLETED')
C
C*********************************************************************
C GENERATE AND WRITE DATA POINTS
C*********************************************************************
C
   60 I2=ILIN(2)
      DATOUT(1)=L1(1)
      DATOUT(2)=L1(2)
      I2=32767
      DATOUT(3)=L1(1)
      DATOUT(4)=L1(2)
      I2=0
      ICOUNT=4
      DO 90 IX=1,NOPNTS
      ICOUNT=ICOUNT+1
      I2=PNTCLS(IX)
      L1(1)=.FALSE.
      IPOL=(I2-1)*NOCHAN
      IBEG=(I2-1)*NOCOMP
      K=IBEG
      DO 65 IY=1,NOCHAN
      DO 65 IZ=1,IY
      K=K+1
      B(IY,IZ)=Z(K)
      IF(IY.EQ.IZ) GO TO 65
      B(IZ,IY)=0.0
   65 CONTINUE
      DO 70 IY=1,NOCH
      CALL RANDU(ISTART(IY),NXINP,A2(IY))
      ISTART(IY)=NXINP
      CALL RANDU(ISTART(IY),NXINP,A(IY))
      ISTART(IY) = NXINP
      A(IY)=SQRT(-2.*ALOG(A2(IY)))*COS(6.28318*A(IY))
   70 CONTINUE
      CLAPNT(I2)=CLAPNT(I2)+1
      DO 80 IY=1,NOCHAN
      DATA(IY)=0.0
      IQ=NOPOOL*NOCOMP+IPOL+IY
      DO 75 IZ=1,NOCHAN
   75 DATA(IY)=DATA(IY)+B(IY,IZ)*A(IZ)
      DATA(IY)=DATA(IY)+Z(IQ)
      INTDAT=DATA(IY)+.5
      IF(INTDAT.LT.0) INTDAT=0
      IF(INTDAT.GT.255) INTDAT=255
      ISTAT(IY)=INTDAT
      DATOUT((IY-1)*NOSAM+ICOUNT)=LOGDAT(2)
      DO 92 IZ=1,6
   92 DATOUT((IY-1)*NOSAM+ICOUNT+IZ)=.FALSE.
   80 CONTINUE
      DO 90 II=1,NOCHAN
      IMEAN(I2,II)=IMEAN(I2,II)+ISTAT(II)
      DO 90 JJ=II,NOCHAN
      IVAR(I2,II,JJ)=IVAR(I2,II,JJ)+ISTAT(II)*ISTAT(JJ)
   90 CONTINUE
      NOBYTE=4+NOCHAN*NOSAM
      CALL TOPWR(12,NOBYTE,IER,DATOUT)
      IF(IER.NE.0) WRITE(16,234)IER
      IF(IER.GT.0) GO TO 310
      GO TO 55
   95 CONTINUE
```

```
FILE:  SWRITE   FORTRAN  A   LARS / PURDUE UNIVERSITY

      DO 100 IP=1,NOCLAS
      DO 100 IO=1,NOCHAN
      IF(CLAPNT(IP).LE.0) GO TO 98
      RMEAN(IP,IO)=FLOAT(IMEAN(IP,IO))/FLOAT(CLAPNT(IP))
   98 DO 100 IT=IO,NOCHAN
      IF(CLAPNT(IP).LE.1) GO TO 100
      REPNT=FLOAT(CLAPNT(IP))
      REVAR=FLOAT(IVAR(IP,IO,IT))
      REMEAN=FLOAT(IMEAN(IP,IO))
      SEMEAN=FLOAT(IMEAN(IP,IT))
      RVAR(IP,IO,IT)=(1./(REPNT-1.))*(REVAR-REMEAN*SEMEAN/REPNT)
      RVAR(IP,IT,IO)=RVAR(IP,IO,IT)
  100 CONTINUE
      DO 645 IP=1,NOCLAS
      WRITE(6,605)IP,CLAPNT(IP)
  605 FORMAT(1H1/5X,'CLASS NUMBER',I3,5X,I8,' POINTS'///)
      WRITE(6,610)
  610 FORMAT(37X,'ACTUAL',4X,'SIMULATED')
      WRITE(6,615)
  615 FORMAT(38X,'MEAN',7X,'MEAN'/)
      DO 622 IX=1,NOCHAN
      NINC=NOCOMP*NOCLAS+(IP-1)*NOCHAN
      WRITE(6,620)FETVC3(IX),FRQCAL(1,FETVC3(IX)),FRQCAL(2,FETVC3(IX)),
     $Z(NINC+IX),RMEAN(IP,IX)
  620 FORMAT(5X,'CHANNEL',I3,2X,'(',F5.2,'-',F5.2,')',5X,F8.3,3X,F8.3)
  622 CONTINUE
      WRITE(6,625)
  625 FORMAT(/////5X,'ACTUAL COVARIANCE MATRIX')
      DO 630 NO=1,NOCOMP
      NINC=(IP-1)*NOCOMP
  630 A(NO)=Z2(NINC+NO)
      CALL WRTMTX(A,NOCHAN,FRQCAL,THREE,FETVC3)
      WRITE(6,635)
  635 FORMAT(////5X,'SIMULATED COVARIANCE MATRIX')
      NO=0
      DO 640 IO=1,NOCHAN
      DO 640 IN=1,IO
      NO=NO+1
  640 A(NO)=RVAR(IP,IO,IN)
      CALL WRTMTX(A,NOCHAN,FRQCAL,THREE,FETVC3)
  645 CONTINUE
      CALL TOPEF(12,IER)
      DO 650 IX=3,200
  650 IDREC(IX)=0
      CALL TOPWR(12,800,IER,IDREC)
      IF(IER.NE.0) WRITE(16,234)IER
      IF(IER.GT.0) GO TO 310
      GO TO 320
  234 FORMAT(5X,'ERROR IS',I5)
C
C*************************************************************************
C ERROR MESSAGES
C*************************************************************************
C
  300 WRITE(6,305)
  305 FORMAT(5X,'ERROR -1')
  310 WRITE(6,315)
  315 FORMAT(5X,'ERROR GT 1')
  320 STOP
      END
```

FILE: HUGHES FORTRAN A LARS / PURDUE UNIVERSITY

```
C*************************************************************
C
C         HUGHES FORTRAN
C       PROGRAM TO CALCULATE THE PROBABLITY OF ERROR FOR TWO CLASSES
C        PROGRAM REQUIRES AS INPUT A DECK IN THE READER FILE AS
C        FOLLOWS:
C        - FIRST CARD: NUMBER OF TRAINING SAMPLES OF CLASS 1
C          (FORMAT I3)
C        - SECOND CARD: NUMBER OF TRAINING SAMPLES OF CLASS 2
C          (FORMAT I3)
C        - MEANS AND COVARIANCE MATRICES OF CLASS 1 AND 2 IN
C          LARSYS FORMAT
C          THE PROGRAM GIVES AS AN OUTPUT THE PROBABILITY OF CORRECT
C        CLASSIFICATION FOR EACH CHANNEL (FOR CHANNEL1, CHANNEL 1,2
C        CHANNELS 1,2,3, ETC.), THE TRANSFORMATION MATRIX AND THE
C        NEW MEAN AND COVARIANCE MATRICES.
C        THE PROGRAM REQUIRES THE FOLLOWING EXEC FILE
C
C        GETDISK IMSL
C        GLOBAL TXTLIB FORTMOD2 CMSLIB DIMSLIB SIMSLIB
C        LOAD HUGHES
C        START
C
C*************************************************************************
C
C
C*************************************************************************
C
C        LIST OF VARIABLES
C
C        N1: NUMBER OF TRAINING SAMPLES OF CLASS 1
C        N2: NUMBER OF TRAINING SAMPLES OF CLASS 2
C        EGVAL1: EIGENVALUE VECTOR OF Σ1, Σ2 AFTER TRANSFORMATION
C        DD1: NEW MEAN VECTOR OF CLASS 1
C        DD2: NEW MEAN VECTOR OF CLASS 2
C        VSGMA1: VARIANCE OF η1 = VAR H(X/W1)
C        VSGMA2: VARIANCE OF η2 = VAR H(X/W2)
C        TRANS1: TRANSFORMATION MATRIX
C        SS1NEW: NEW COVARIANCE MATRIX OF CLASS 1
C        SS2NEW: NEW COVARIANCE MATRIX OF CLASS 2
C        CONST: MULTIPLICATIVE FACTOR OF VAR (η1) AND VAR (η2)
C
C*************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 SIGMA1(78),SIGMA2(78),AINV(78),WK(102),PS1S2(12,12),
     *WR(168),M1(12),M2(12),PERROR,EGVECS(12,1),EGVECT(1,12),CC(1,12),
     *EGVALR(24),EGVECR(288),SIGM1S(12,12),AA(1,1),DEGVEC(12,12),
     *EGVAL1(12),BATACH(12),TEMP1(12),DDEGVC(12,12),MEANR(2),MEANS(2),
     *SGMR(2),SGMS(2),GAMAR(2),GAMAS(2),ALPHR(2),ALPHS(2),
     *CR(2),CS(2),A(2),B(2),DELTAR(2),DELTAS(2),DIST(2),ERROR(2),
     *SS1NEW(78),SS2NEW(78),ASGMS(2),ASGMR(2),
     *SIGMAS(12,12),DD1(12),DD2(12),TRANS(12,12),TRANS1(12,12),
     *LAMBDA,WORK(400),MEANS1(12,2),MEANR1(12,2),SGMS1(12,2),SGMR1(12,2)
     *DSGN1(12),VSGMA(2)
      COMPLEX*16 EGVAL(12),EGVEC(12,12),ZN,
     *X1,X2,D1(12),D2(12)
      EQUIVALENCE (EGVAL(1),EGVALR(1)),(EGVEC(1,1),EGVECR(1))
C
C*************************************************************
C
C        READ NUMBER OF TRAINING SAMPLES OF CLASS 1 AND 2
C        READ MEAN VECTORS OF CLASSES 1 AND 2
C        READ COVARIANCE MATRICES OF  CLASS 1 AND 2
C
C*************************************************************
C
       READ(5,967)N1
       READ (5,967)N2
 967     FORMAT(I3)
      READ (5,130)M1
      READ (5,130)M2
      READ (5,130) SIGMA1
      READ(5,130)SIGMA2
 130  FORMAT(2X,5E14.7)
      N = 12
C
C*************************************************************
C       COMPUTE INVERSE OF COVARIANCE MATRIX OF CLASS I
C*************************************************************
```

```
FILE: HUGHES   FORTRAN  A   LARS / PURDUE UNIVERSITY

     5 +(1.0/EGVAL1(I)**4)*(8.0/N1 +8.0/N2 +128.0/(N1*N2) +40.0/N1**2
     6 +40.0/N2**2 +48.0/N1**3 +48.0/N2**3 +512.0/(N1**2*N2)
     7 +512.0/(N1*N2**2) +1920.0/(N1**2*N2**2) +576.0/(N1**3*N2)
     8 +576.0/(N2**3*N1) +2112.0/(N1**2*N2**3) +2112.0/(N1**3*N2**2)
     9 +2304.0/(N1**3*N2**3) +4.0*DSQR21(I)*(4.0/N1 +8.0/N2
     * +8.0/N1**2 +40.0/N2**2 +64.0/(N1*N2) +256.0/(N1*N2**2)
     * +96.0/(N1**2*N2) +48.0/N2**3 +288.0/(N1*N2**3) +352.0/(N1**2*N2
     * **2) +384.0/(N1**2*N2**3)) +4.0*DSQR21(I)**2*(2.0/N1 +8.0/N2
     * +40.0/N2**2 +24.0/(N1*N2) +48.0/N2**3 +88.0/(N1*N2**2)
     * +96.0/(N1*N2**3))))
      VSGMA(2)=VSGMA(2)+4.0*((EGVAL1(I)**4)*(8.0/N1 +8.0/N2
     1 +128.0/(N1*N2) +40.0/N1**2 +40.0/N2**2 +48.0/N1**3 +48.0/N2**3
     2 +512.0/(N1**2*N2) +512.0/(N1*N2**2) +1920.0/(N1**2*N2**2)
     3 +576.0/(N1**3*N2) +576.0/(N1*N2**3)
     4 +2112.0/(N1**3*N2**2) +2112.0/(N1**2*N2**3) +2304.0/(N1**3*N2**3
     5 )) +(4.0*EGVAL1(I)**3)*(DSQR21(I)*(8.0/N1 +4.0/N2 +8.0/N2**2
     6 +40.0/N1**2 +64.0/(N1*N2) +256.0/(N1**2*N2) +96.0/(N2**2*N1)
     7 +48.0/N1**3 +288.0/(N2**3*N1) +352.0/(N1**2*N2**2) +
     8 384.0/(N2**2*N1**3)) -(4.0/N1 +4.0/N2 +8.0/N1**2 +8.0/N2**2
     9 +32.0/(N1*N2) +48.0/(N1*N2**2) +48.0/(N1**2*N2)
     * +64.0/(N1**2*N2**2))) +(2.0*EGVAL1(I)**2)*(4.0/N1 +4.0/N2
     * +8.0/(N1*N2) +(2.0*DSQR21(I)**2)*(8.0/N1 +2.0/N2 +40.0/N1**2
     * +24.0/(N1*N2) +48.0/N1**3 +88.0/(N1**2*N2) +96.0/(N1**3*N2))
     * -4.0*DSQR21(I)*(2.0/N2 +4.0/N1 +12.0/(N1*N2) +8.0/N1**2
     * +16.0/(N1**2*N2**2))))
      DO 141 J=1,2
      IF(A(J).GT.0.0)GO TO 979
      MEANS(J)=MEANS(J)+A(J)*(1.0+B(J)**2)
      SGMS(J)=SGMS(J)+2.0*((A(J)**2)*(1.0+2.0*(B(J))**2))
 974  FORMAT(10X,'SGMRB = ',F20.4)
C
C*************************************************************
C
C        CALCULATE MULTIPLICATIVE FACTOR AND NEW σ̂ AND σ̂
C
C*************************************************************
C
979   XZX=DFLOAT(I)
      CONST=0.1+2.0*(XZX**2)/(N1*N2)
 143  ASGMS(J)=SGMS(J)+CONST*DSQRT(VSGMA(J))
  192 ALPHS(J)=((MEANS(J)**2)/ASGMS(J))-1.0
      IF(ALPHS(J).GE.0.35)GAMAS(J)=1.0
      CS(J)=(MEANS(J))-DSQRT((GAMAS(J)+1.0)*ASGMS(J))
      DELTAS(J)=ASGMS(J)/(MEANS(J)-CS(J))
      CS(J)=-MEANS(J)-DSQRT((GAMAS(J)+1.0)*ASGMS(J))
      IF(A(J).GT.0.0)GO TO 142
       GO TO 144
 142  MEANR(J)=MEANR(J)+A(J)*(1.0+B(J)**2)
      SGMR(J)=SGMR(J)+2.0*((A(J)**2)*(1.0+2.0*(B(J))**2))
874    FORMAT(30X,F20.4)
873    FORMAT(10X,F20.4)
       XZX=DFLOAT(I)
 144    ASGMR(J)=SGMR(J)+CONST*DSQRT(VSGMA(J))
  193 ALPHR(J)=((MEANR(J)**2)/ASGMR(J))-1.0
      IF(ALPHR(J).GE.0.35)GAMAR(J)=1.0
      CR(J)=MEANR(J)-DSQRT((GAMAR(J)+1.0)*ASGMR(J))
      DELTAR(J)=ASGMR(J)/(MEANR(J)-CR(J))
 141  CONTINUE
C
C*************************************************************
C
C    CALCULATE PROBABILITY OF ERROR
C
C*************************************************************
C
      PSI=PSI+DLOG(EGVAL1(I))+((DD2(I)-DD1(I))**2)/(EGVAL1(I)-1.0)
      DO 145 J=1,2
      DIST(J)=PSI-(CR(J)-CS(J))
 145  CONTINUE
      DO 146 K =1,2
      IF(DIST(K).LT.0.0)GO TO 147
      IF(DELTAR(K).EQ.0.0)GO TO 148
     ERROR(K)=1.0-((DELTAR(K)/(DELTAR(K)+DELTAS(K)))**
     1(GAMAS(K)+1.0))*(((DIST(K)/DELTAR(K))+1.0+((GAMAR(K)+
     2GAMAS(K))*DELTAS(K))/(DELTAR(K)+DELTAS(K)))**GAMAR(K))*
     3DEXP(-DIST(K)/DELTAR(K))
      GO TO 146
 148  ERROR(K)=1.0
      GO TO 146
```

```
FILE: HUGHES   FORTRAN  A   LARS / PURDUE UNIVERSITY

C
      CALL LINV2P(SIGMA1,N,AINV,IDGT,EE1,EE2,WK,IER)
       WRITE(6,117)IER
 117   FORMAT(' ',I3)
C
C*************************************************************
C     COMPUTE INVERSE OF COVARIANCE MATRIX 1 MULTIPLIED
C     BY COVARIANCE MATRIX 2
C*************************************************************
C
      CALL VMULSS(AINV,SIGMA2,N,PS1S2,N)
C
C*************************************************************
C    COMPUTE EIGENVALUES AND EIGENVECTORS OF (INVERSE(
C    SIGMA1)) (SIGMA2)
C*************************************************************
C
      CALL EIGRF(PS1S2,N,N,2,EGVALR,EGVECR,N,WR,IERR)
       WRITE(6,117)IERR
       WRITE(6,126)WR(1)
126     FORMAT(' ',F6.1)
C
C*************************************************************
C     NORMALIZING EIGENVECTORS (SEE FUKUNAGA,
C    PAGE 35)
C*************************************************************
C
       CALL VCVTSF(SIGMA1,N,SIGM1S,N)
       CALL VCVTSF(SIGMA2,N,SIGM2S,N)
      DO 10 I = 1,N
      DO 20 J = 1,N
      EGVECT(1,J) = DREAL(EGVEC(J,I))
        EGVECS(J,1)=DREAL(EGVEC(J,I))
 20    CONTINUE
      M = N
      NN = N
      CALL VMULFF(EGVECT,SIGM1S,1,M,NN,1,N,CC,1,IEER)
       WRITE(6,126)IEER
      CALL VMULFF(CC,EGVECS,1,M,NN,1,N,AA,1,IIER)
       WRITE(6,126)IIER
      AA(1,1) = DSQRT(AA(1,1))
      DO 30 K = 1,N
      EGVEC(K,I) = EGVEC(K,I)/AA(1,1)
 30     CONTINUE
 10    CONTINUE
C
C*************************************************************
C
C*************************************************************
C    CALCULATE NEW MEAN VECTOR DI = EGVEC*MI
C*************************************************************
C
      DO 90 I = 1,N
      D1(I)=(0.0,0.0)
      D2(I)=(0.0,0.0)

 90   CONTINUE
C
C*************************************************************************
C    CALCULATE NEW MEAN VECTORS
C*************************************************************************
C
      DO 95 I =1,N
      DO 95 J =1,N
         DEGVEC(I,J) = DREAL(EGVEC(J,I))
      EGVAL1(I) = DREAL(EGVAL(I))
      DDEGVC(I,J) = DREAL(EGVEC(I,J))
      D1(I)=EGVEC(J,I)*M1(J)+D1(I)
      D2(I)=EGVEC(J,I)*M2(J)+D2(I)
      TRANS1(I,J)=0.0
 95   CONTINUE
 183   FORMAT(5F14.7)
       DO 777 I =1,N
       DD1(I)=DREAL(D1(I))
       DD2(I)=DREAL(D2(I))
 777   CONTINUE
C
C*************************************************************
C
```

```
FILE: HUGHES   FORTRAN  A   LARS / PURDUE UNIVERSITY

C    ORDER THE EIGENVALUES AND EIGENVECTORS ACCORDING TO
C         MAXIMUM EIGENVALUE
C
C*****************************************************************
C

      DO 120 I=1,N
      DO 120 J=1,N
      IF(EGVAL1(I)-EGVAL1(J))120,120,131
 131  TEMP=EGVAL1(I)
      TEMPP=DD1(I)
      TTEMP=DD2(I)
      EGVAL1(I)=EGVAL1(J)
      DD1(I)=DD1(J)
      DD2(I)=DD2(J)
      EGVAL1(J)=TEMP
      DD1(J)=TEMPP
      DD2(J)=TTEMP
      DO 132 K=1,N
      TEMP1(K)=DDEGVC(K,I)
      DDEGVC(K,I)=DDEGVC(K,J)
      DDEGVC(K,J)=TEMP1(K)
 132  CONTINUE
 120  CONTINUE
C*****************************************************************
C
C   INITIALIZE ALL PARAMETERS UNDER CONSIDERATION
C
C*****************************************************************
C
      WRITE(6,136)
       DO 134 I=1,N
       DO 134 J=1,N
       TRANS(I,J)=DDEGVC(J,I)
 134   CONTINUE
      DO 135 II =1,2
      MEANR(II)=0.0
      MEANS(II)=0.0
      SGMR(II)=0.0
      SGMS(II)=0.0
      GAMAR(II)=0.0
      GAMAS(II)=0.0
      ALPHR(II)=0.0
      ALPHS(II)=0.0
      DELTAR(II)=0.0
      DELTAS(II)=0.0
      CR(II)=0.0
      CS(II)=0.0
      PSI=0.0
      VSGMA(II)=0.0

 135   CONTINUE
C
C*****************************************************************
C
C   CALCULATE PARAMETERS OF GAMMA DISTRIBUTIONS
C
C*****************************************************************
C
 136  FORMAT(' ',10X,'FIRST N DIMENSIONS',10X,'PROBABILITY OF ERROR')
      DO 140 I = 1,N
      A(1)=1.0-1.0/EGVAL1(I)
      B(1)=(DD1(I)-DD2(I))/(EGVAL1(I)-1.0)
      A(2)=EGVAL1(I)-1.0
      B(2)=(DSQRT(EGVAL1(I))*(DD1(I)-DD2(I)))/(EGVAL1(I)-1.0)
      DSQR21(I)=(DD1(I)-DD2(I))**2
C
C*****************************************************************
C
C         CALCULATE VAR (σ̂ᵢ) AND VAR (σ̂ᵢ²)
C
C*****************************************************************
C
      VSGMA(1)=VSGMA(1)+4.0*((2.0/EGVAL1(I)**2)*(4.0/N1 +4.0/N2
     1 +8.0/(N1*N2)) -(4.0/EGVAL1(I)**3) * (4.0/N1 +4.0/N2 +
     2 8.0/N1**2 +8.0/N2**2 +32.0/(N1*N2) +48.0/(N1*N2**2)
     3 + 48.0/(N1**2*N2) +64.0/(N1**2*N2**2) + 4.0*DSQR21(I)*
     4 (1.0/N1 +2.0/N2 +6.0/(N1*N2) +4.0/N2**2 +8.0/(N1*N2**2)))
```

FILE: HUGHES FORTRAN A LARS / PURDUE UNIVERSITY

```
 147  IF(DELTAS(K).EQ.0.0)GO TO 149
      ERROR(K)=((DELTAS(K)/(DELTAS(K)+DELTAR(K)))**(GAMAR(K)+1.0))*
     1((-((DIST(K))/DELTAS(K))+1.0+((GAMAR(K)+GAMAS(K))*DELTAR(K))/
     2(DELTAR(K)+DELTAS(K)))**GAMAS(K))*DEXP(DIST(K)/DELTAS(K))
      GO TO 146
 149  ERROR(K)=0.0
 146  CONTINUE
      XXX=1.0-ERROR(2)
      PERROR = 0.5*(1.0-ERROR(1)+ERROR(2))
      PCC=1.0-PERROR
 159  FORMAT(44X,F20.4)
      WRITE(6,150)I,PCC
 150  FORMAT(' ',16X,I2,25X,F7.5)
 151  FORMAT(' ',5X,F10.3,5X,F10.3)
 152  CONTINUE
      WRITE(6,155)
 155  FORMAT(/)
 140    CONTINUE
 190  CONTINUE
C
C*****************************************************************
C
C       PRINT TRANSFORMATION MATRIX AND NEW MEAN AND COVARIANCE
C       MATRICES
C
C*****************************************************************
C
       WRITE(6,919)
 919   FORMAT(10X,'TRANSFORMATION VECTOR')
       WRITE(6,183)((TRANS(I,J),J=1,N),I=1,N)
       WRITE(6,920)
 920   FORMAT(//)
       WRITE(6,921)
 921   FORMAT(10X,'NEW MEAN VECTORS AND COVRIANCE MATRICES OF CLASS 1
     * AND 2')
       WRITE(6,165)(DD1(I),I=1,N)
       WRITE(6,165)(DD2(I),I=1,N)
 165   FORMAT('MN',5E14.7)
       DO 748 I=1,N
       DO 746 J=I,N
       IF(I.NE.J)GO TO 747
       SS1NEW(I+(I*(I-1))/2)=1.0
       SS2NEW(I+(I*(I-1))/2)=EGVAL1(I)
       GO TO 746
 747   SS1NEW(I+(J*(J-1))/2)=0.0
       SS2NEW(I+(J*(J-1))/2)=0.0
 746   CONTINUE
 748   CONTINUE
       NMN=N*(N+1)/2
       WRITE(6,175)(SS1NEW(I),I=1,NMN)
       WRITE(6,175)(SS2NEW(I),I=1,NMN)
 175   FORMAT('CV',5E14.7)
 432   STOP
       END
```

Appendix F

Description of Data Sets For Experiments

F.1 Training and Test Fields for Aircraft, Simulated Data Set (Tape 203, file 3)

Training Fields

```
CLASS CORN
RUN(71053900),LINE(304,326,2),COL(109,133,2)
RUN(71053900),LINE(512,528,1),COL(87,93,1)
RUN(71053900),LINE(620,636,1),COL(107,123,2)
RUN(71053900),LINE(656,676,2),COL(33,59,2)
CLASS FOREST
RUN(71053900),LINE(798,812,1),COL(141,161,2)
RUN(71053900),LINE(704,720,1),COL(147,155,1)
RUN(71053900),LINE(726,736,1),COL(81,95,1)
```

Test Fields (Also Area Classified)

```
TEST CORN
RUN(71053900),LINE(143,154,1),COL(42,57,1)
RUN(71053900),LINE(305,318,1),COL(116,132,1)
RUN(71053900),LINE(403,413,1),COL(17,33,1)
RUN(71053900),LINE(643,657,1),COL(121,127,1)
RUN(71053900),LINE(684,691,1),COL(11,30,1)
RUN(71053900),LINE(857,866,1),COL(34,53,1)
TEST FOREST
RUN(71053900),LINE(424,430,1),COL(161,173,1)
RUN(71053900),LINE(521,531,1),COL(142,162,1)
RUN(71053900),LINE(711,728,1),COL(149,158,1)
RUN(71053900),LINE(769,779,1),COL(127,148,1)
RUN(71053900),LINE(837,851,1),COL(155,162,1)
RUN(71053900),LINE(923,931,1),COL(70,79,1)
```

F.2 Training and Test Fields for Aircraft, Real Data Set (Tape 203, file 1)

Training Fields

```
CLASS CORN
RUN(71053900),LINE(304,326,2),COL(109,133,2)
RUN(71053900),LINE(512,528,1),COL(87,93,1)
RUN(71053900),LINE(620,636,1),COL(107,123,2)
RUN(71053900),LINE(656,676,2),COL(33,59,2)
CLASS FOREST
RUN(71053900),LINE(798,812,1),COL(141,161,2)
RUN(71053900),LINE(704,720,1),COL(147,155,1)
RUN(71053900),LINE(726,736,1),COL(81,95,1)
```

Test Fields (Also Area Classified)

```
TEST CORN
RUN(71053900),LINE(227,247,1),COL(81,96,1)
RUN(71053900),LINE(334,351,1),COL(66,100,3)
RUN(71053900),LINE(452,474,2),COL(108,119,1)
RUN(71053900),LINE(597,611,1),COL(137,153,2)
RUN(71053900),LINE(646,664,1),COL(101,128,2)
RUN(71053900),LINE(711,721,1),COL(102,113,1)
TEST FOREST
RUN(71053900),LINE(241,249,1),COL(27,45,1)
RUN(71053900),LINE(509,527,1),COL(181,193,1)
RUN(71053900),LINE(729,751,2),COL(201,217,1)
RUN(71053900),LINE(765,803,2),COL(191,203,2)
RUN(71053900),LINE(833,855,2),COL(151,171,2)
RUN(71053900),LINE(989,1005,1),COL(141,155,2)
```

F.3 Training and Test Fields for Landsat, Multitemporal, Simulated Data Set (Tape 203, file 6)

Training Fields

```
CLASS CORN
78843016      25    32    1    33    42    1
78843016      62    67    1   133   141    1
78843016      30    33    1    87   102    1
78843016      91    97    1    79    86    1
CLASS SOYB
78843016       9    12    1    61    77    1
78843016      74    82    1    51    64    1
78843016     110   117    1   167   172    1
```

Test Fields (Also Area Classified)

```
TEST CORN
RUN(78843016),LINE(2,12,1),COL(30,34,1)
RUN(78843016),LINE(38,46,1),COL(18,26,1)
RUN(78843016),LINE(55,58,1),COL(103,117,1)
RUN(78843016),LINE(16,22,1),COL(123,127,1)
RUN(78843016),LINE(70,73,1),COL(80,89,1)
RUN(78843016),LINE(85,93,1),COL(47,50,1)
RUN(78843016),LINE(102,104,1),COL(140,155,1)
RUN(78843016),LINE(107,115,1),COL(11,15,1)
TEST SOYBEANS
RUN(78843016),LINE(1,4,1),COL(91,100,1)
RUN(78843016),LINE(16,20,1),COL(56,70,1)
RUN(78843016),LINE(32,34,1),COL(114,126,1)
RUN(78843016),LINE(49,51,1),COL(113,125,1)
RUN(78843016),LINE(76,84,1),COL(31,40,1)
RUN(78843016),LINE(99,106,1),COL(127,132,1)
RUN(78843016),LINE(106,114,1),COL(53,59,1)
```

F.4 Training and Test Fields for Landsat, Multitemporal, Real Data Set (Tape 203, file 5)

Training Fields

```
CLASS CORN
78843016      26   32   1    32   42   1
78843016      91   98   1    79   86   1
78843016      62   67   1   134  141   1
78843016      30   34   1    91  102   1
CLASS SOYB
78843016       9   13   1    68   78   1
78843016      74   82   1    51   63   1
78843016     100  105   1   120  132   1
```

Test Fields (Also Area Classified)

```
TEST CORN
RUN(78843016),LINE(2,11,1),COL(27,32,1)
RUN(78843016),LINE(38,46,1),COL(19,25,1)
RUN(78843016),LINE(103,106,1),COL(140,156,1)
RUN(78843016),LINE(101,115,1),COL(12,17,1)
RUN(78843016),LINE(78,86,1),COL(124,128,1)
RUN(78843016),LINE(67,74,1),COL(94,98,1)
RUN(78843016),LINE(35,41,1),COL(123,127,1)
TEST SOYBEANS
RUN(78843016),LINE(41,44,1),COL(67,79,1)
RUN(78843016),LINE(79,84,1),COL(31,40,1)
RUN(78843016),LINE(106,114,1),COL(54,59,1)
RUN(78843016),LINE(44,51,1),COL(118,123,1)
RUN(78843016),LINE(1,4,1),COL(90,100,1)
RUN(78843016),LINE(109,113,1),COL(132,147,1)
RUN(78843016),LINE(44,47,1),COL(155,161,1)
```

F.5 Training and Test Fields for Aircraft Binary Tree Example (Tape 203, file 1)

Training Fields

CLASS WHT1							
71053900	11	626	626	1	162	162	1NS-
71053900	12	627	627	1	164	164	1NS-
71053900	14	628	628	1	159	159	1NS-
71053900	16	629	629	1	163	163	1NS-
71053900	22	635	635	1	167	167	1NS-
71053900	3	461	461	1	71	71	2NS-
71053900	4	461	461	1	79	79	2NS-
71053900	9	463	463	1	75	75	2NS-
71053900	4	621	621	1	167	167	1NS-
71053900	10	624	624	1	159	159	1NS-
71053900	20	633	633	1	161	161	1NS-
71053900	21	634	634	1	163	163	1NS-
71053900	27	639	639	1	163	163	1NS-
CLASS WHT2							
71053900	3	314	314	1	163	163	1NS-
71053900	6	316	316	1	166	166	1NS-
71053900	7	317	317	1	159	159	1NS-
71053900	8	318	318	1	157	157	1NS-
71053900	10	319	319	1	157	157	1NS-
71053900	17	324	324	1	167	167	1NS-
71053900	18	325	325	1	165	165	1NS-
71053900	21	327	327	1	167	167	1NS-
71053900	22	328	328	1	158	158	1NS-
71053900	7	462	462	1	79	79	2NS-
71053900	10	463	463	1	77	77	2NS-
71053900	17	469	469	1	67	67	2NS-
71053900	21	471	471	1	75	75	2NS-
CLASS HAY							
71053900	2	484	484	1	55	55	2NS-
71053900	1	880	880	1	132	132	1NS-
71053900	3	882	882	1	126	126	1NS-
71053900	7	883	883	1	126	126	1NS-
71053900	14	886	886	1	128	128	1NS-
71053900	15	887	887	1	133	133	1NS-
71053900	18	889	889	1	134	134	1NS-
71053900	19	890	890	1	135	135	1NS-
71053900	20	891	891	1	128	128	1NS-
71053900	30	895	895	1	132	132	1NS-
71053900	13	488	488	1	41	41	2NS-
71053900	16	490	490	1	43	43	2NS-
71053900	19	894	894	1	135	135	1NS-

CLASS PAS1							
71053900	2	402	402	1	157	157	2NS-
71053900	32	417	417	1	153	153	2NS-
71053900	34	418	418	1	149	149	2NS-
71053900	1	1012	1012	1	101	101	1NS-
71053900	1	1012	1012	1	102	102	1NS-
71053900	1	1012	1012	1	107	107	1NS-
71053900	5	1014	1014	1	101	101	1NS-
71053900	6	1015	1015	1	103	103	1NS-
71053900	7	1016	1016	1	102	102	1NS-
71053900	10	1017	1017	1	113	113	1NS-
71053900	10	1017	1017	1	115	115	1NS-
71053900	12	1018	1018	1	112	112	1NS-
71053900	15	1020	1020	1	107	107	1NS-
CLASS PAS2							
71053900	0	418	418	1	147	147	2
71053900	0	588	588	1	67	67	2
71053900	0	589	589	1	65	65	2
71053900	0	589	589	1	67	67	2
71053900	0	589	589	1	69	69	2
71053900	0	589	589	1	75	75	2
71053900	0	593	593	1	71	71	2
71053900	0	595	595	1	61	61	2
71053900	0	595	595	1	71	71	2
71053900	0	596	596	1	57	57	2
71053900	0	596	596	1	59	59	2
71053900	0	596	596	1	67	67	2
71053900	0	597	597	1	63	63	2
CLASS SOY							
71053900	4	424	424	2	125	125	2NS-
71053900	3	336	336	2	165	165	2NS-
71053900	22	352	352	2	165	165	2NS-
71053900	1	488	488	2	123	123	2NS-
71053900	2	488	488	2	133	133	2NS-
71053900	22	500	500	2	127	127	2NS-
71053900	9	312	312	2	63	63	2NS-
71053900	10	312	312	2	67	67	2NS-
71053900	5	424	424	2	131	131	2NS-
71053900	7	426	426	2	113	113	2NS-
71053900	11	426	426	2	137	137	2NS-
71053900	41	440	440	2	137	137	2NS-
71053900	23	502	502	2	119	119	2NS-
CLASS CRN							
71053900	8	516	516	1	93	93	1NS-
71053900	10	518	518	1	87	87	1NS-
71053900	17	521	521	1	93	93	1NS-
71053900	11	623	623	1	121	121	2NS-
71053900	15	625	625	1	123	123	2NS-
71053900	3	656	656	2	53	53	2NS-
71053900	23	322	322	2	119	119	2NS-
71053900	29	326	326	2	111	111	2NS-
71053900	19	527	527	1	90	90	1NS-
71053900	8	660	660	2	35	35	2NS-
71053900	16	664	664	2	45	45	2NS-
71053900	24	668	668	2	55	55	2NS-
71053900	29	672	672	2	41	41	2NS-
CLASS FST							
71053900	11	731	731	1	85	85	1NS-
71053900	13	709	709	1	154	154	1NS-
71053900	17	711	711	1	151	151	1NS-
71053900	32	718	718	1	147	147	1NS-
71053900	3	726	726	1	90	90	1NS-
71053900	4	726	726	1	95	95	1NS-
71053900	27	732	732	1	95	95	1NS-
71053900	32	735	735	1	82	82	1NS-
71053900	15	803	803	1	149	149	2NS-
71053900	20	805	805	1	145	145	2NS-
71053900	30	809	809	1	141	141	2NS-
71053900	11	709	709	1	151	151	1NS-
71053900	28	718	718	1	151	151	1NS-

CLASS WAT								
71053900		5	888	888	1	165	165	1NS-
71053900		8	891	891	1	162	162	1NS-
71053900		9	892	892	1	164	164	1NS-
71053900		1	936	936	1	139	139	1NS-
71053900		3	938	938	1	141	141	1NS-
71053900		3	938	938	1	143	143	1NS-
71053900		6	939	939	1	143	143	1NS-
71053900		6	939	939	1	146	146	1NS-
71053900		8	941	941	1	140	140	1NS-
71053900		10	943	943	1	138	138	1NS-
71053900		11	944	944	1	140	140	1NS-
71053900		14	947	947	1	141	141	1NS-
71053900		15	948	948	1	141	141	1NS-

Test Fields (Also Area Classified)

TEST WHEAT								
71053900		304	312	1	155	161	1	WHEATCUT
71053900	UU6	839	848	1	67	70	1	WHEATCUT
71053900	U6	854	861	1	73	77	1	WHEATCUT
71053900	UU7	829	851	2	73	91	2	WHEAT
71053900	HH3	619	641	2	151	161	1	WHEAT
71053900	GG2	569	575	1	145	148	1	OATSCUT
71053900	FF9	459	475	2	81	99	1	OATS
TEST HAY								
71053900	Z22	873	887	1	19	67	2	HAY
71053900	L8	899	923	2	85	99	1	HAY
71053900	C4	252	275	2	33	35	1	HAY
71053900	G2	659	661	1	92	96	1	HAY
71053900	Q5	713	715	1	39	50	1	HAY
71053900	CC2	361	387	2	155	165	1	HAY
71053900	BB9	313	327	1	173	185	1	HAY
TEST PASTURE								
71053900	L2	589	599	1	77	93	1	PASTURE
71053900	Z21	1021	1031	1	103	117	1	PASTURE
71053900	O1	731	743	1	31	55	2	PASTURE
71053900	I2	669	675	1	101	123	2	PASTURE
71053900	T9	1013	1037	2	201	211	1	PASTURE
71053900	HH9	683	693	1	97	129	2	PASTURE
71053900	EE5	421	439	2	177	191	1	PASTURE
71053900	Z20	423	445	2	11	27	1	PASTURE
TEST SOYBEANS								
71053900	DD6	593	613	1	101	127	2	SOYBEANS
71053900	G4	649	687	2	77	83	1	SOYBEANS
71053900	RR2	861	867	1	123	149	2	SOYBEANS
71053900	II5	649	671	2	177	191	1	SOYBEANS
71053900	OO2	479	519	2	105	139	2	SOYBEANS
71053900	R7	449	475	2	27	55	2	SOYBEANS
71053900	Z9	205	231	2	195	211	2	SOYBEANS

TEST CORN								
71053900	A3	227	247	1	81	96	1	CORN
71053900	A5	225	247	1	49	59	1	CORN
71053900	C1	283	295	1	67	95	2	CORN
71053900	F5	374	387	1	89	99	1	CORN
71053900	DD3	452	474	2	108	119	1	CORN
71053900	HH1	597	611	1	137	153	2	CORN
71053900	JJ1	711	721	1	102	113	1	CORN
71053900	Z15	481	515	2	3	21	2	CORN
71053900	F6	373	387	1	47	79	2	CORN
71053900	Z16	305	327	2	191	205	1	CORN
TEST FOREST								
71053900	A10	241	249	1	27	45	2	FOREST
71053900	Z6	729	751	2	201	217	2	FOREST
71053900	Z3	765	803	2	191	203	1	FOREST
71053900	EE4	522	525	1	155	159	1	FOREST
71053900	RR4	833	855	1	151	171	1	FOREST
71053900	HH10	765	799	2	139	159	2	FOREST
71053900	M3	783	795	1	49	81	2	FOREST
71053900	Z18	375	387	1	191	201	1	FOREST
TEST WATER								
71053900	A9	205	209	1	34	38	1	PONDWATR
71053900	U2	817	819	1	49	51	1	PONDWATR
71053900	A7	221	224	1	27	29	1	PONDWATR
71053900	W3	1000	1004	1	51	54	1	WATER
71053900	W2	1010	1014	1	36	39	1	WATER
71053900	QQ7	969	973	1	126	131	1	WATER
71053900	W7	849	855	1	201	205	1	WATER
71053900	W6	873	879	1	185	191	1	WATER
71053900	W5	977	983	1	113	119	1	WATER
71053900	W4	1041	1047	1	11	15	1	WATER

F.6 Training and Test Fields for Landsat, Multitemporal Binary Tree Example (Tape 203, file 5)

Training Fields

CLASS CORN							
78843016	0	28	28	1	33	33	1
78843016	0	29	29	1	35	35	1
78843016	0	30	30	1	37	37	1
78843016	0	30	30	1	42	42	1
78843016	0	32	32	1	34	34	1
78843016	0	32	32	1	35	35	1
78843016	0	32	32	1	39	39	1
78843016	0	64	64	1	134	134	1
78843016	0	64	64	1	137	137	1
78843016	0	65	65	1	141	141	1
78843016	0	30	30	1	93	93	1
78843016	0	30	30	1	96	96	1
78843016	0	34	34	1	102	102	1
CLASS SOYBEANS							
78843016	0	11	11	1	69	69	1
78843016	0	13	13	1	72	72	1
78843016	0	74	74	1	57	57	1
78843016	0	74	74	1	63	63	1
78843016	0	75	75	1	52	52	1
78843016	0	76	76	1	56	56	1
78843016	0	76	76	1	61	61	1
78843016	0	77	77	1	53	53	1
78843016	0	80	80	1	60	60	1
78843016	0	81	81	1	59	59	1
78843016	0	82	82	1	58	58	1
78843016	0	100	100	1	125	125	1
78843016	0	101	101	1	130	130	1
CLASS ELSE							
78843016	0	51	51	1	154	154	1
78843016	0	52	52	1	154	154	1
78843016	0	52	52	1	160	160	1
78843016	0	53	53	1	158	158	1
78843016	0	55	55	1	161	161	1
78843016	0	91	91	1	180	180	1
78843016	0	91	91	1	182	182	1
78843016	0	92	92	1	177	177	1
78843016	0	94	94	1	178	178	1
78843016	0	95	95	1	188	188	1
78843016	0	52	52	1	39	39	1
78843016	0	1	1	1	50	50	1
78843016	0	7	7	1	49	49	1

Test Fields (Also Area Classified)

```
TEST CORN
RUN(78843016),LINE(2,11,1),COL(27,32,1)
RUN(78843016),LINE(38,46,1),COL(19,25,1)
RUN(78843016),LINE(103,106,1),COL(140,156,1)
RUN(78843016),LINE(101,115,1),COL(12,17,1)
RUN(78843016),LINE(78,86,1),COL(124,128,1)
RUN(78843016),LINE(67,74,1),COL(94,98,1)
RUN(78843016),LINE(35,41,1),COL(123,127,1)
TEST SOYBEANS
RUN(78843016),LINE(41,44,1),COL(67,79,1)
RUN(78843016),LINE(79,84,1),COL(31,40,1)
RUN(78843016),LINE(106,114,1),COL(54,59,1)
RUN(78843016),LINE(44,51,1),COL(118,123,1)
RUN(78843016),LINE(1,4,1),COL(90,100,1)
RUN(78843016),LINE(109,113,1),COL(132,147,1)
RUN(78843016),LINE(44,47,1),COL(155,161,1)
TEST ELSE
RUN(78843016),LINE(33,42,1),COL(137,141,1)
RUN(78843016),LINE(54,57,1),COL(39,52,1)
RUN(78843016),LINE(55,59,1),COL(136,149,1)
RUN(78843016),LINE(95,109,1),COL(191,194,1)
RUN(78843016),LINE(108,114,1),COL(83,89,1)
```