4-1-1996

HIGH DIMENSIONAL FEATURE REDUCTION VIA PROJECTION PURSUIT

Luis O. Jimenez
Purdue University School of Electrical and Computer Engineering

David Landgrebe
Purdue University School of Electrical and Computer Engineering

High Dimensional Feature Reduction Via Projection Pursuit

Luis O. Jimenez
David Landgrebe

TR-ECE 96-5
April 1996

School of Electrical
and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285

HIGH DIMENSIONAL FEATURE REDUCTION VIA PROJECTION PURSUIT

Luis O. Jimenez
David Landgrebe

TR-ECE 96-5
April 1995

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE IN 47907-1285

Table of Contents

ABSTRACT

The recent development of more sophisticated sensors for remote sensing systems enables the measurement of radiation in many more spectral intervals than previous possible. An example of this technology is the AVIRIS system., which collects image data in 220 bands. The increased dimensionality of such hyperspectral data provides a challenge to the current techniques for analyzing such data.

Our experience in three dimensional space tends to mislead our intuition of geometrical and statistical properties in high dimensional space, properties that must guide our choices in the data analysis process. Using Euclidean and Cartesian geometry, high dimensional space properties are investigated and their implication for high dimensional data and its analysis are studied in order to illuminate the differences between conventional spaces and hyperdimensional space.

Supervised classification techniques use labeled samples in order to train the classifier. Usually the number of such samples is limited, and as the number of bands available increases, this limitation becomes more severe, and can become dominate over the projected added value of having the additional bands available. This suggests the need for reducing the dimensionality via a preprocessing method which takes into consideration high dimensional space properties. Such reduction should enable the estimation of feature extraction parameters to be more accurate. Using a technique referred to as Projection Pursuit, two parametric projection pursuit algorithms have been developed: Parallel Parametric Projection Pursuit and Sequential Parametric Projection Pursuit. In the present work both methods are presented, and an iterative procedure of the Sequential Approach that mitigates the computation time problem is shown.

Parametric Projection Pursuit' methods requires the use of a numerical optimization algorithm. A method to estimate an initial value that can more quickly lead to the global maximum is presented for projection pursuit using Bhattacharyya distance as the Projection Index. This method leads also to a high dimensional version of a feature selection algorithm, which requires significantly less computation than the normal procedure.

1. INTRODUCTION

1.1 Background

Multispectral image data consist of a set of measurements containing information from the scene at a number of different spectral wavelengths. Remote Sensing rnultispectral data may include measurements from ultraviolet, visible near, middle, and thermal infrared and microwave ranges of wavelengths. The different ranges of wavelengths characterize the interaction mechanism between electromagnetic radiation and the materials illuminated. The reflected energy measured by the sensors depends on such properties as pigmentation, moisture content and cellular structure of vegetation, mineral and moisture content of soil, the level of sedimentation of water, and the heat capacity of material surfaces among others [1]. On the basis that every material will have a different spectral response, one expects to be able to classify the scene into different materials or regions. This type of process is used, for example, by agricultural analysts in the classification of crops. The purpose of acquiring remote sensing image data is to identify and classify different surface materials by their spatial and spectral distribution of energy [2].

In the present research, multispectral data will be modeled as rnultivariate data distributions, and this will allow us to use the theory of stochastic or random processes [3]. On the basis of this representation, multivariate statistical analysis will be used to produce quantitative results. Specifically, we will use statistical pattern recognition to categorize each elementary observation into one of a limited number of discrete pre-specified classes. The pattern recognition and classification model contains three parts: a transducer, a feature extractor and a classifier [4] (see Figure 1.1). The transducer is the sensor that produces the multispectral image data. The feature extractor extracts relevant information of the input data. The classifier assigns the observation to one of the possible classes. The classification performs a partition in the feature space into different regions and assigns the observations to each one of the classes depending in the region of the feature space where they are localized. That

partition will be developed with the objective of minimizing the probability of error in the process of classification. We expect that each class will have different statistical properties, in their spectral response for a particular scene. As a consequence we will be able to separate them into different classes.

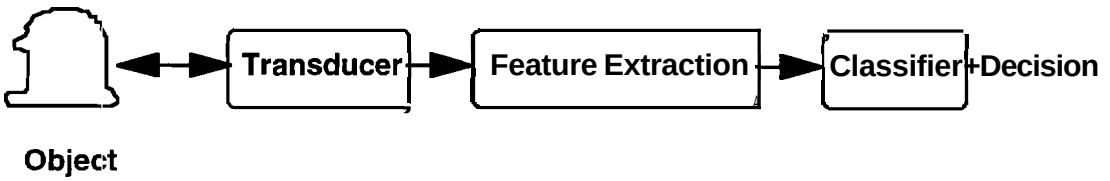


Fig. 1.1. Classical pattern recognition and classification model.

1.2 Statement of the Problem

The recent development of more sophisticated remote sensing systems enables the measurement of radiation in many more spectral intervals than possible previously. An example of this technology is the AVIRIS system, which collects image data in 220 bands. As the number of dimensions of high spectral resolution data increases;, the capability to detect more detailed classes should also increase. Although, with the increment of the number of features, the cost and complexity of the feature extractor and classifier increase, it is expected that the classification accuracy will increase as well. In statistical pattern recognition, supervised classification techniques use labeled samples available for training the classifier and estimates its performance. Even if the classifier has good performance on the training samples that is not guaranteed in new samples. That is the reason the labeled samples have been divided in two independent sets: one for learning (training samples) and the other for estimating its classification accuracy (test samples). Usually the number of such samples is limited. It has been observe frequently in practice that beyond a certain point, if the number of training samples per feature is small, the addition of more dimensions leads to a worst performance in terms of a penalty in the test samples classification accuracy. Hughes proved that the basic source of the problem is the limited number of training samples [5]. The penalty becomes more serious in high dimensional cases. In other terms, as the number of dimensions and classes increase with the number of training samples being fixed the problem get worse. That is why the optimum number of features for classification is limited by the nurnber of training samples [6]. In order to avoid what has been named the Hughes phenomena, there had been some empirical and analytical research in the adequate proportion of the number of training samples per number of features. Fukunaga [7] proved that the

required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier. In terms of nonparametric classifiers the situation is even worse. It has been estimated that as tlie number of dimensions increases the training sample:; size need to increases exponentially in order to have an effective estimate of the multivariate densities needed to perform a nonparametric classification [8] [9]. These limitations are what had been called the curse of dimensionality [4, pp. 95]. That condition had restricted severely the practical applications of statistical pattern recognition procedures in high dimensional data.

The previous discussion shows the need to reduce the dimensionality of the data. A number of techniques for feature extraction have been developed to reduce dimensionality. Among these techniques are Principal Components, Discriminant Analysis, and Decision Boundary Feature Extraction [10]. These techniques estimate the statistics at full dimensionality in order to extract relevant features for classification. If the number of training samples is not adequately large the estimation of parameters in high dimensional data will not be accurate enough. As a result,, the estimated features nnay not be reliable. The use of a data preprocessing algorithm before the use of any feature extraction algorithm had been proposed in order to reduce the dimensionality [11]. In the present work a different preprocessing algorithm is proposed,,It will produce a linear combination of features that reduces dimensionality, but by performing the computation at a lower dimensional space, consequently avoiding what had been named the curse of dimensionality. That reduction enables the estimation of parameters to be more accurate for feature extraction with classification purposes (see Figure 2).

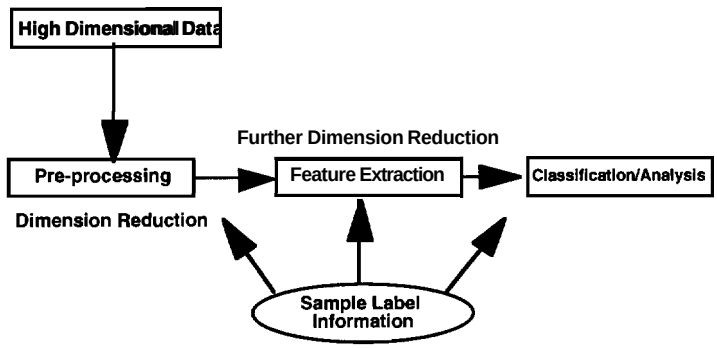


Fig. 1.2. Preprocessing of high dimensional data.

The preprocessing method developed in the present work will take into account a priori, problem specific information. It will be developed after considering some characteristics of high dimensional space geometry and statistics of multispectral data. Its objective is to linearly combines features, at the same time preserving the distance between classes.

1.3 Thesis Organization

Our familiarity with a three spatial dimensions world is based on our experience. At the same time we are not capable of imagining a high dimensional space in order to develop some intuition of its differences and similarities with the known three dimensional Euclidean space. Still we can grasp some insights of high dimensional spaces with the use of some mathematical tools. Chapter 2 will study some patterns of high dimensional space and their implication for hiigh dimensional data and its analysis. That will provide the rationale, the need, and the requisites of a preprocessing block.

In chapter 3 a study and evaluation of different feature extraction techniques will be done. It will show the development of the algorithm that will accomplish the objective of the preprocessing block fulfilling the requisites established in chapter 2. That algorithm is based on a technique developed in statistics named Projection Pursuit. Based on the fact that the algorithm will do its computation at a lower dimensional subspace, it will require the use of a numerical optimization method. Chapter 4 will show a further development that has the objective of avoiding local optima. Finally chapter 5 will provide a summary of the conclusions and suggestions for further work. Experimental results for different classifiers and feature extraction methods are provided throughout the thesis.

2. HIGH DIMENSIONAL SPACE PROPERTIES

2.1 Introduction

The complexity of dimensionality has been known for more than three decades, and its impact varies from one field to another. In combinatorial optimization over many dimensions, it is seen as an exponential growth of the computational effort with the number of dimensions. In statistics, it manifests itself as a problem with parameter or density estimation due to the paucity of data. The negative effect of this paucity results from some geometrical, statistical and asymptotical properties of high dimensional feature space. These characteristics exhibit surprising behavior of data in higher dimensions.

There are many assumptions that we make about characteristics of lower dimensional spaces based on our experience in three dimensional Euclidean space. There is a conceptual barrier that makes it difficult to have proper intuition of the properties of high dimensional space and its consequences in high dimensional data behavior. Most of the assumptions that are important for statistical purposes we tend to relate to our three dimensional space intuition, for example, as to where the concentration of volume is of such figures as cubes, spheres, and ellipsoids or where the data concentration is in known density function families such as normal and uniform. Other important perceptions that are relevant for statistical analysis are, for example, how the diagonals relate to the coordinates, the number of labeled samples required for supervised classification, the assumption of normality in data, and the importance of mean and covariance difference in the process of discrimination among different statistical classes. In the next section some characteristics of high dimensional space will be studied, and their impact in supervised classification data analysis will be discussed. Most of these properties do not fit our experience in three dimensional Euclidean space as mentioned before.

2.2 Geometrical, Statistical And Asymptotical Properties

In this section we illustrate some unusual or unexpected hyperspace characteriistics including a proof and discussion. These illustrations are intended to show that higher dimensional space is quite different from the dimensional space with which we are familiar.

As dimensionality increases:

A. The volume of a hypercube concentrates in the comers [8, pp. 29].

It has been shown [12] that the volume of the hypersphere of radius r and dimension d is given by the equation:

$$V_s(r) = volume - sphere = \frac{2r^d}{d}\frac{\pi^{\bar{2}}}{\Gamma\left(\frac{d}{2}\right)} \tag{2.1}$$

and that the volume of a hypercube in $[-r, r]^d$ is given by the equation:

$$V_c(r) = volume - cube = (2r)^d \tag{2.2}$$

The fraction of the volume of a hypersphere inscribed in a hypercube is:

$$f_{d1} = \frac{V_s(r)}{V_C(r)} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma\left(\frac{d}{2}\right)} \tag{2.3}$$

where d is the number of dimensions. We see in Figure *2.1* how *(2.3)* decreases as the dimensionality increases.

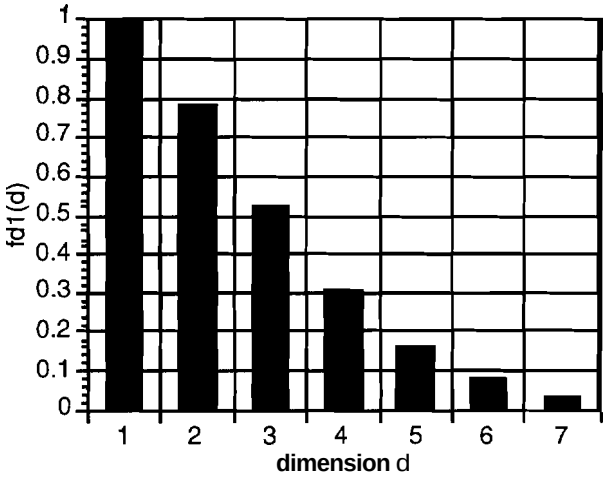


Fig. 2.1. Fractional volume of a hypersphere inscribed in a hypercube as a function of dimensionality.

Note that $lim_{d\to\infty} f_{d1} = 0$ which implies that the volume of the hypercube is increasingly concentrated in the corners as d increases.

B. The volume of a hypersphere concentrates in an outside shell **[8, pp. 29]** [13].

The fraction of the volume of an outside shell of a sphere of radius r-ε inscribed in a sphere of radius r is:

$$f_{d2} = \frac{V_d(r) - V_d(r-\varepsilon)}{V_d(r)} = \frac{r^d - (r-\varepsilon)^d}{r^d} = 1 - \left(1 - \frac{\varepsilon}{r}\right)^d \qquad (2.4)$$

In Figure 2.2 we can observe, for the case $\varepsilon = r/5$, how as the dimension increases the volume concentrates in the outside shell.

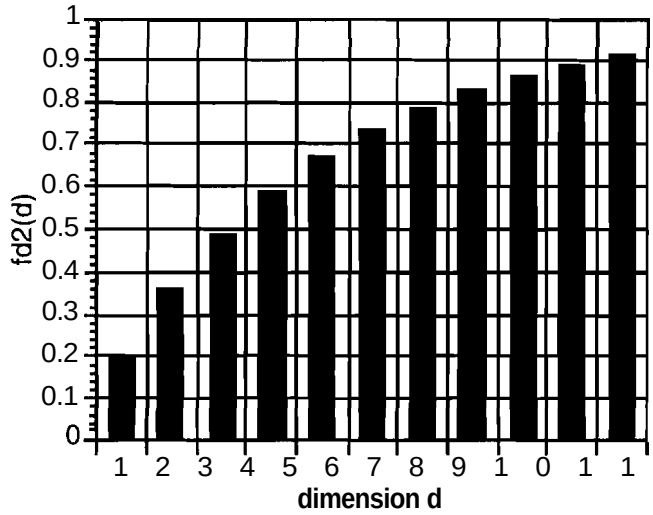


Fig. 2.2. Volume of a hypersphere contained in the outside shell as a function of dimensionality.

Note that $lim_{d\to\infty} f_{d2} = 1, \forall \varepsilon > 0$, implying that most of the volume of a hypersphere is concentrated in an outside shell.

C. The volume of a hyperellipsoid concentrates in an outside shell.

Here the previous result will be generalized to a hyperellipsoid. Let the equation of a hyperellipsoid in ddimensions be written as:

$$\frac{X_1^2}{\lambda_1^2} + \frac{X_2^2}{\lambda_2^2} + \cdots + \frac{X_d^2}{\lambda_d^2} = 1 \qquad (2.5)$$

The volume is calculated by the equation [12, pp. 36]:

$$V_e(\lambda_i) = \frac{2\prod_{i=1}^{\bar{}} \lambda_i}{d} \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}\right)} \qquad (2.6)$$

The volume of a hyperellipsoid defined by the equation:

$$\frac{X_1^2}{(\lambda_1 - \delta_1)^2} + \frac{X_2^2}{(\lambda_2 - \delta_2)^2} + \cdots + \frac{X_d^2}{(\lambda_d - \delta_d)^2} = 1 \qquad (2.7)$$

where $0 \leq 6, < \lambda_i, \forall i$, is calculated by:

$$V_e(\lambda_i - \delta_i) = \frac{2\prod_{i=1}^{d}(\lambda_i - \delta_i)}{d} \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}\right)} \tag{2.8}$$

The fraction of the volume of $V_e(\lambda_i - \delta_i)$ inscribed in the volume $V_e(\lambda_i)$ is:

$$f_{d3} = \frac{\prod_{i=1}^{d}(\lambda_i - \delta_i)}{\prod_{i=1}^{d}\lambda_i} = \prod_{i=1}^{d}\left(1 - \frac{\delta_i}{\lambda_i}\right) \tag{2.9}$$

Let $\gamma_{min} = min\left(\frac{\delta_i}{\lambda_i}\right)$, then

$$f_{d3} = \prod_{i=1}^{d}\left(1 - \frac{\delta_i}{\lambda_i}\right) \le \prod_{i=1}^{d}(1 - \gamma_{\min}) = (1 - \gamma_{\min})^d \tag{2.10}$$

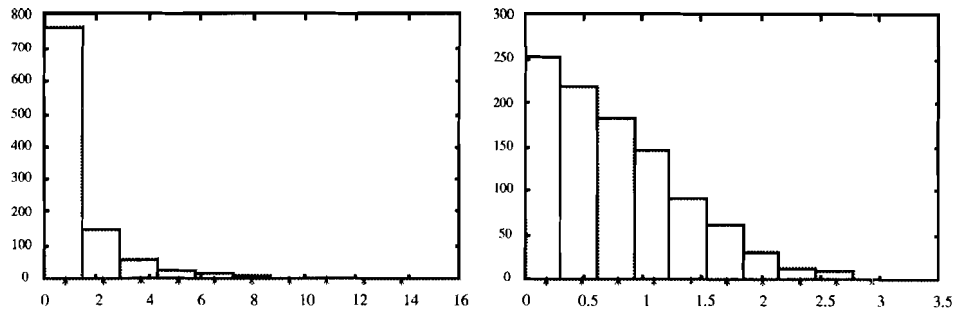
Using the fact that $f_{d3} \ge 0$ it is concluded that $\lim_{d\to\infty} f_{d3} = 0$.

The characteristics previously mentioned have two important consequences for high dimensional data that appear immediately. The first one is that high dimensional space is mostly empty, which implies that multivariate data in R^d is usually in a lower dimensional structure. As a consequence high dimensional data can be projected to a lower dimensional subspace without losing significant information in terms of separability among the different statistical classes. The second consequence of the foregoing, is that normally distributed data will have a tendency to concentrate in the tails; similarly, uniformly distributed data will be more likely to be collected in the corners, making density estimation more difficult. Local neighborhoods are almost surely empty, requiring the bandwidth of estimation to be large and producing the effect of losing detailed density estimation.

Support for this tendency can be found in the statistical behavior of normally and uniformly distributed multivariate data at high dimensionality. It is expected that as the dimensionality increases the data will concentrate in an outside shell. As the number of dimensions increases that shell will increase its distance from the origin as well.

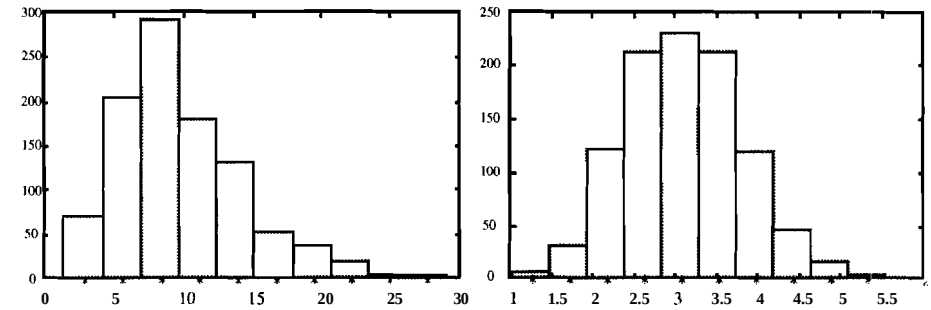
To show this specific multivariate data behavior, an experiment was developed. Multivariate normal and uniform distributed data were generated. The normal and uniform variables are independent identically distributed samples from the distributions N(0,1) and U(-1,1), respectively. Figures 2.3 and 2.4 illustrate the histograms of random variables, the distance from the zero coordinate and its square, that are functions of normal or uniform vectors at different number of dimensions.

Normal, dimensions = 1



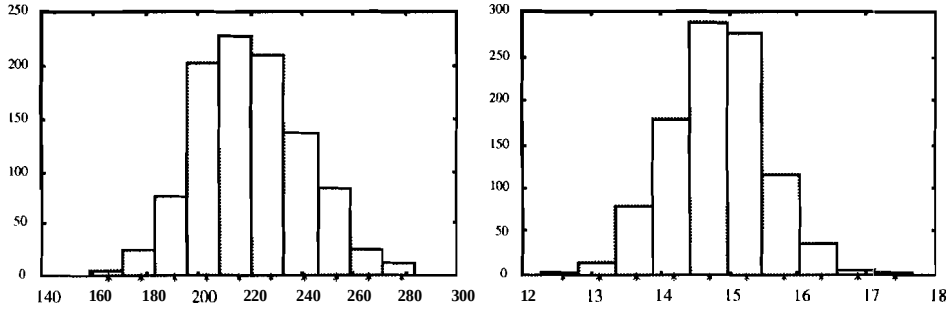
$\sum_i x_i^2, \mu = 1.0168, \sigma = 1.4017$ $\sqrt{\sum_i x_i^2}, \mu = .7736, \sigma = .5737$

Normal, dimensions = 10



$\sum_i x_i^2, \mu = 9.8697, \sigma = 4.5328$ $\sqrt{\sum_i x_i^2}, \mu = 3.1026, \sigma = 0.7042$

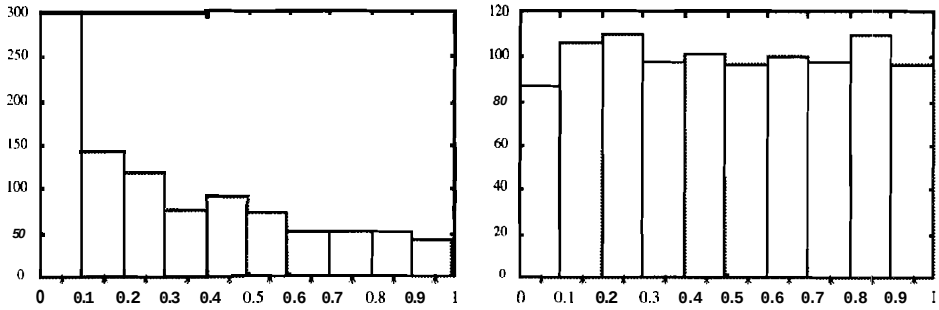
Normal, dimensions = 220



$\sum_i x_i^2, \mu = 220.3732, \sigma = 20.7862$ $\sqrt{\sum_i x_i^2}, \mu = 14.8497, a = 0.7129$

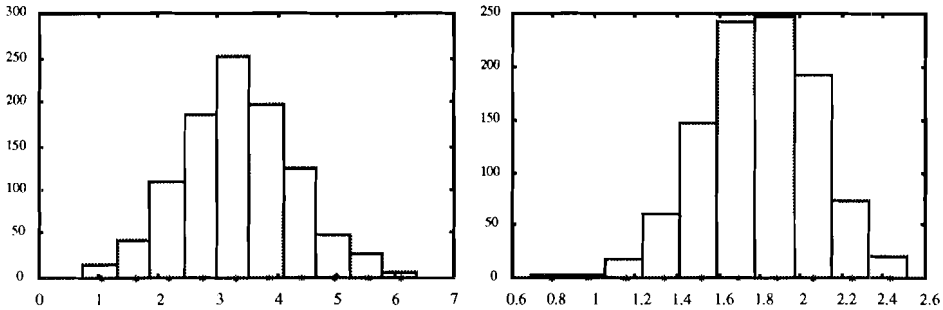
Fig. 2.3. Histograms of functions of Normally distributed random variables.

Uniform, dimensions = 1



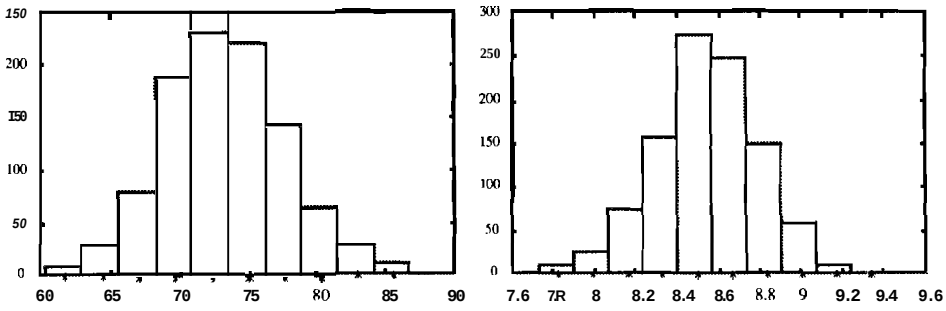
$\sum_i x_i^2, \mu = 0.3277, \sigma = 0.2883$ $\sqrt{\sum_i x_i^2}, \mu = 0.5041, \sigma = 0.2887$

Uniform, dimensions = 10



$\sum_i x_i^2, \mu = 3.3444, \sigma = 0.9390$ $\sqrt{\sum_i x_i^2}, \mu = 1.8010, \sigma = 0.2678$

Uniform, dimensions = 220



$\sum_i x_i^2, \mu = 73.3698, \sigma = 4.3854$ $\sqrt{\sum_i x_i^2}, \mu = 8.5488, = 0.2505$

Fig. 2.4. Histograms of functions of Uniformly distributed random variables.

From Figure 2.3, the data increasingly concentrate in an outside shell with the growth of dimensions. It can be observed that the concentration of points moves out from zero coordinates as the dimensionality increments.

These experiments show how the means and the standard deviations are functions of the number of dimensions. As the dimensionality increases the data concentrates in an outside shell. The mean and standard deviation of two random variables:

$$r = \sqrt{\sum_{i=1}^{d} x_i^2} \tag{2.11}$$

$$R = \sum_{i=1}^{d} x_i^2 \tag{2.12}$$

are computed. These variables are the distance and the square of the! distance of the random vectors. The values of the parameters and the histograms of the random variables are shown in Figure 2.3 and 2.4 for normal and uniform distribution of the data. As the dimensionality increases the distance from the zero coordinate of both random variables increases as well. These results show that the data have a tendency to concentrate in an outside shell and how the shell's distance from the zero coordinate increases with the increment of the number of dimensions.

Note that (2.12) has a chi-square distribution with *d* degrees of freedom when the x_i's are samples from the N(0,1) distribution. The mean and variance of Rare: $E(R) = d$, $Var(R) = 2d$ [14]. This conclusion supports the previous thesis.

Under these circumstances it would be difficult to implement any density estimation procedure and to obtain accurate results. Generally nonparametric approaches will have even greater problems with high dimensional data.

D. *The diagonals are nearly orthogonal to all coordinate axis* [8, pp. 27-31] [13].

The cosine of the angle between any diagonal vector and a Euclidean coordinate axis is:

$$cos(\theta_d) = \pm\frac{1}{\sqrt{d}} \tag{2.13}$$

Figure 2.5 illustrates how the angle between the diagonal and the coordinates, *theta(d)*, approaches 900 with increases in dimensionality.

Note that $lim_{d\to\infty} cos(\theta_d) = 0$, which implies that in high dimensional space the diagonals have a tendency to become orthogonals to the Euclidean coordinates.

This result is important because the projection of any cluster onto any diagonal, e.g., by averaging features, could destroy information contained in multispectral data.

In order to explain this, let a_{diag} be any diagonal in a *d* dimensional space. Let ac_i be the ith coordinate of that space. Any point in the space can be represented by the form:

$$P = \sum_{i=1}^{d} \alpha_i ac_i \qquad (2.14)$$

The projection of P over a_{diag}, P_{diag} is:

$$P_{diag} = \left(P^T a_{diag}\right) a_{diag} = \sum_{i=1}^{d} \alpha_i \left(ac_i^T a_d\right) a_d \qquad (2.15)$$

But as *d* increases $ac_i^T a_{dhg} \approx 0$ which implies that $P_{diag} \approx 0$. As a consequence P_{diag} is being projected to the zero coordinate, losing information about its location at the *d* dimensiorial space.

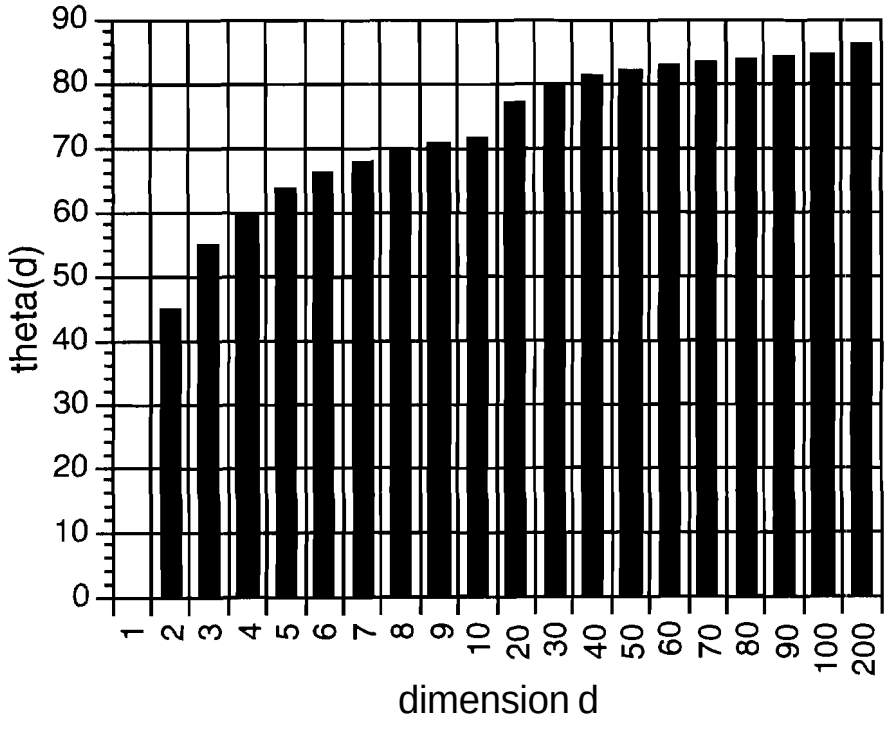


Fig. 2.5. Angle (in degrees) between a diagonal and a Euclidean coordinate vs. dimensionality.

E. The required number of labeled samples for supervised classification increases as a function of dimensionality.

Fukunaga [7] proves that the required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier. That fact is very relevant, especially since experiments have demonstrated that there are circumstances where second order statistics are more relevant than first order statistics in discriminating among classes in high dimensional data [15]. In terms of nonparametric classifiers the situation is even

more severe. It has been estimated that as the number of dimensions increases, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities [8, pp 208-212] [9].

It is to be expected that high dimensional data contains more information. At the same time the above characteristics tell us that it is difficult with the current techniques, which are usually based on computations at full dimensionality, to extract such information unless the available labeled data is substantial. A concrete example of this is the so-called Hughes phenomena. Hughes proved that with a limited number of training samples there is a penalty in classification accuracy as the nurnber of features increases beyond some point [5].

F. *For most high dimensional data sets', low linear projections have the tendency to be normal, or a combination of normal distributions, as the dimension increases.*

That is a significant characteristic of high dimensional data that is quite relevant to its analysis. It has been proved [16] [17] that as the dimensionality tends to infinity, lower dimensional linear projections will approach a normality model with probability approaching one (see Figure 2.6). Normality in this case implies a normal or a combination of normal distributions.

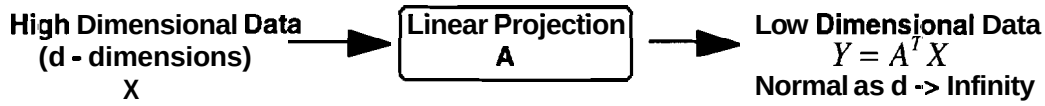


Fig. 2.6. The tendency of lower dimensional projections to be Normal.

Several experiments will illustrate this with simulated and real data. The procedure in these experiments is to project the data from a high dimensional space to a one dirnensional subspace. We examine the behavior of the projected data as the number of dimensions in the original high dimensional space increases from one to ten and finally to one hundred. The method of projecting the data is to multiply it with a normal vector with random angles from the coordinates. A histogram is used to observe the data distribution. A normal density function is plotted with the histogram to compare the results to normal.

Figure 2.7 shows the case of generated data from a uniform distribution. As the number of dimensions increases in the original space the projected data's histogram has a tendency to be normal. Figure 2.8 shows the results of the same experiment with real AVIRIS data with one soybeans class. Note that the results are similar to the generated data.

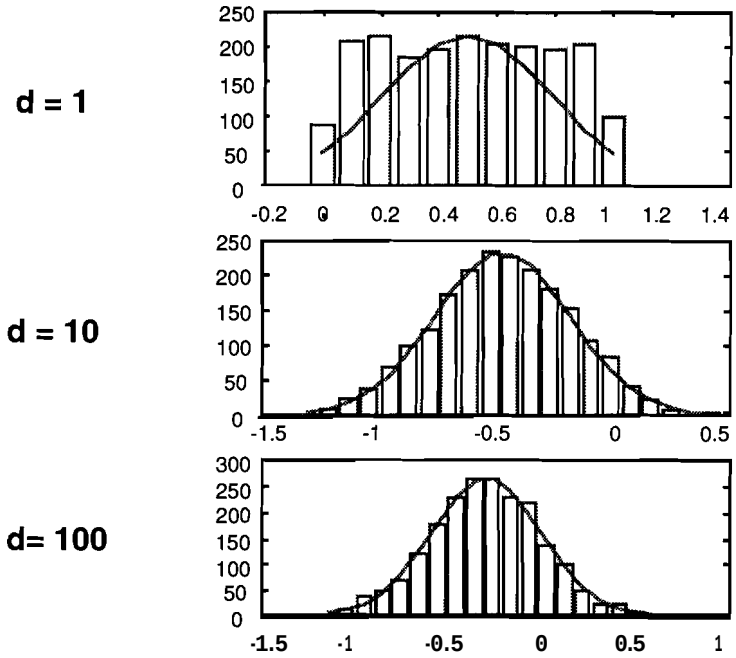


Fig. 2.7. Generated data: One class with Uniform distribution.

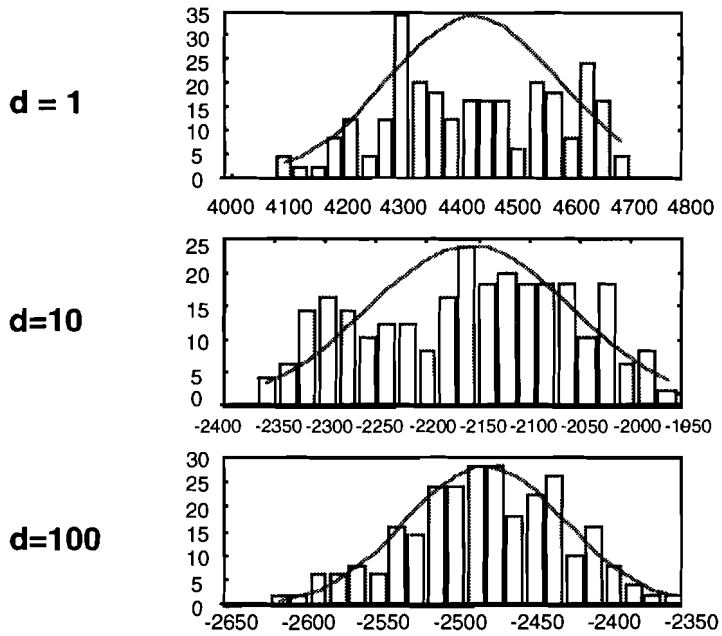


Fig. 2.8. AVIRIS Multispectral data: One class, soybeans.

These results tempt us to expect that the data can be assume to be a combination of normal distributions in the projected subspace without any problem. Other experiments show that a combination of normal distributions where each one represenits a different statistical class could collapse into one normal distribution. That

will imply loss of information. Figure 2.9 and 2.10 show the result of repeating the experiments for a two class problem. Both show the risk of damaging data projecting it into one normal distribution loosing separability and information. In the case of Figure 2.10 we have real AVIRIS data with a corn and a soybeans class.

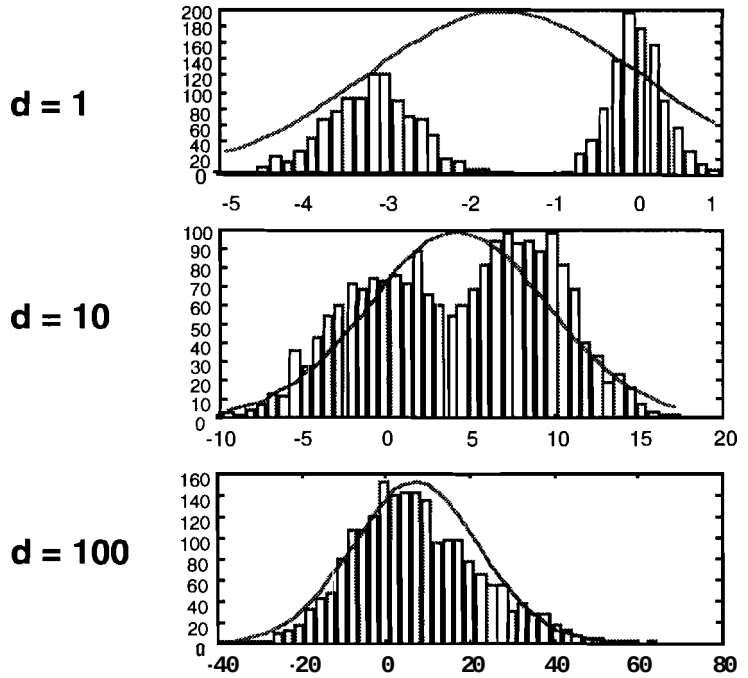


Fig. **2.9. Generated data: Two classes with Normal distributions.**

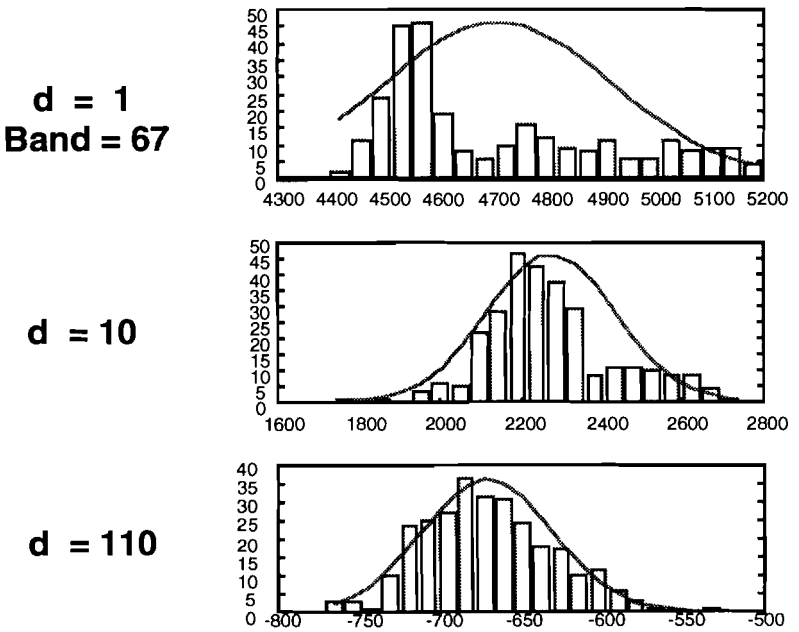


Fig. 2.10. AVIRIS Multispectral data: Two classes, corn and soybeans.

In all the cases above we can see the advantage of developing an algorithm that will estimate the projection directions that separate the explicitly defined classes, doing the computations at a lower dimensional space. The vectors that it computes will separate the classes, and at the same time, the explicitly defined classes will behave asymptotiically more like a normal distribution. The assumption of normality will be better grounded in the projected subspace than at full dimensionality.

2.3 Asymptotical First And Second Order Statistics Properties

Lee and Landgrebe [15] performed an experiment where they classified some high dimensional data in order to see the relative role that first and second order statistics played. To accomplish this objective the experiment compared three classifier!;. The first was an ML classification which uses class mean and class covariance information. The second was an ML classifier constrained to use only covariance differences among classes. The last one was a minimum distance classifier that uses only first order statistics. Figure 2.11 shows their result.

In that particular experiment as the number of dimension grew the role played by the second order statistics increased in discriminating among classes. The authors gave a rational explanation for that particular characteristic based on the fact that there are circumstances where there is a high correlation between adjacent bands and that most data are distributed along a few major components producing a hyperellipsoid shaped clata distribution. Under these circumstances the shape of the distribution given by the second order statistics becomes extremely important.

Hen?a more general basis will be given for the role of the first and second order statistics in hyperspectral data where adjacent bands could be correlated in any way. The results will be based on the asymptotic behavior of high dimensional data. This will aid in the understanding of the conditions required for the predominance of either first order or second order statistics in the discrimination among the sitatistical classes in high dimensional space.

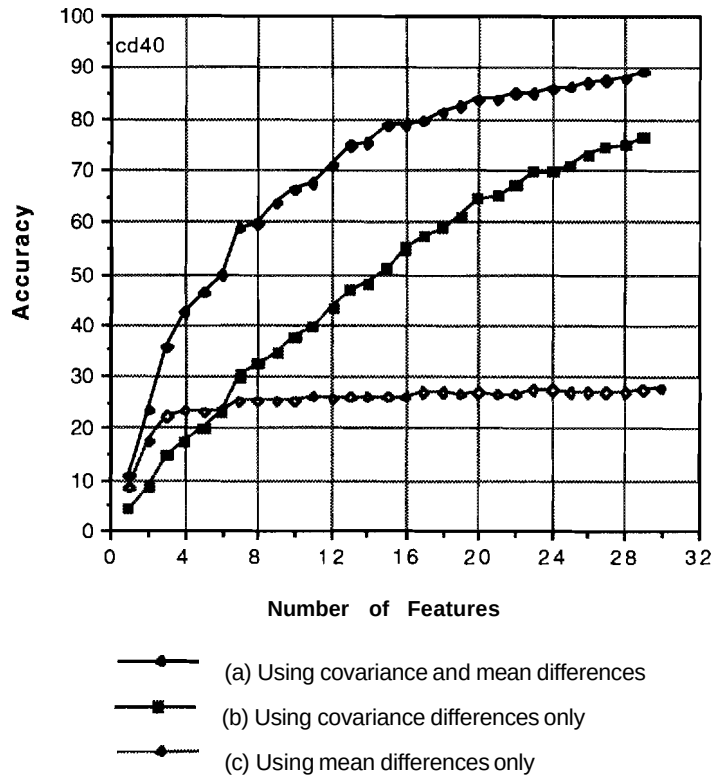


Fig. 2.11. Performance comparison of Normal ML, Normal ML with zero mean data, and the Minimum Distance classifier, each with 12 multitemporal classes.

It is expected that, as the number of features increases, the information contained in multispectral data increases as well. In supervised classification that increment of information is translated to the number of statistical classes and their separability. There are different measures of distance and separability among statistical classes in use. The choice here will be Bhattacharyya distance. It is used because it provides a bound of classification accuracy. In addition it takes into account first order and second order statistics. Bhattacharyya distance is the sum of two components, one based on mean differences and the other based on covariance differences.

The Bhattacharyya distance under the assumption of normality is computed by the equation:

$$\mu = \frac{1}{8}\left(M_2 - M_1\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(M_2 - M_1\right) + \frac{1}{2} ln \left(\frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{\left|\Sigma_1\right|\left|\Sigma_2\right|}}\right) \qquad (2.15)$$

The mean difference component of the Bhattacharyya distance is:

$$\mu_M = \frac{1}{8}\left(M_2 - M_1\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(M_2 - M_1\right) \tag{2.16}$$

and the covariance difference component of the Bhattacharyya distance is:

$$\mu_C = \frac{1}{2} ln \left(\frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{\left|\Sigma_1\right|\left|\Sigma_2\right|}} \right) \tag{2.17}$$

In order to see how Bhattacharyya distance and its mean and covariance components can aid in the understanding of the role of first and second order statistics, two experiment were developed. The first one has conditions where second order statistics are more relevant in discriminating among the classes. The second experiment has conditions for the predominance of first order statistics.

Experiment 1

In this experiment data was generated for two statistical classes. Both classes belong to normal distributions with different means and covariances in a 15 dimensional space. Each class has 500 points. Their respective parameters are:

$$\mathbf{M}_1 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\mathbf{M}_2 = [1.5 \quad 1 \quad .5 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & & & & & & & & & 0 \\ & 1 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \\ 0 & & & & & & & & & 1 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1.5 & & & & & & & & & 0 \\ & 1.9 & & & & & & & & \\ & & 3 & & & & & & & \\ & & & 3 & & & & & & \\ & & & & 3 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 3 & & & \\ & & & & & & & 3 & & \\ & & & & & & & & 3 & \\ 0 & & & & & & & & & 3 \end{bmatrix}$$

The data was classified according to three classifiers. The first was the ML classifier, the second was the ML (ML Cov) classifier constrained to use only covariance difference, and the third was minimum distance classifier (Min Dist). This enables us to have similar conditions to Lee and Landgrebe's experiment. The results is shown in Figure 2.12.

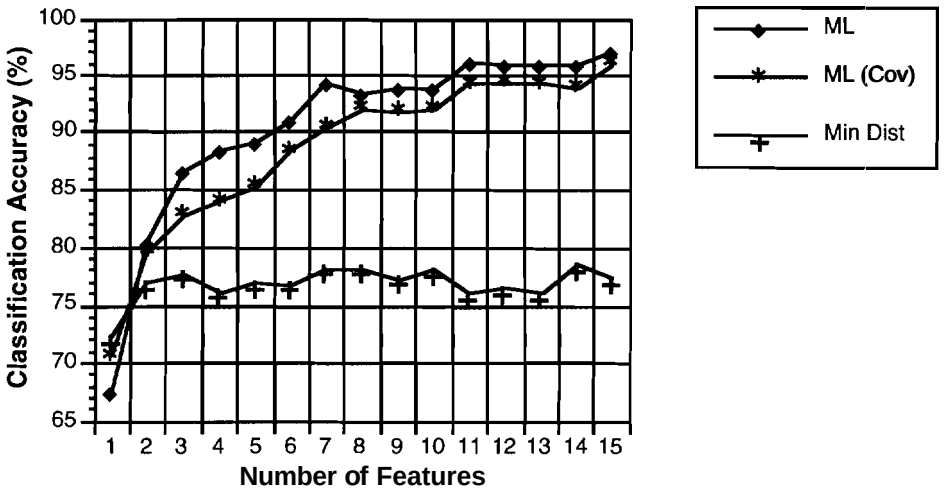


Fig. 2.12. Performance comparison of Normal ML, Normal ML with zero mean data, and Minimum Distance classifier. Two generated classes.

Observe how the results resemble Lee and Landgrebe's results. In order to have an understanding of the roles played by first and second order statistics the mean (Bhatt Mean) and covariance (Bhatt Cov) components of Bhattacharyya distance and its sum were computed and are shown in Figure 2.13. Their ratio of Bhatt Mean / Bhatt Cov was calculated and shown in Figure 2.14.

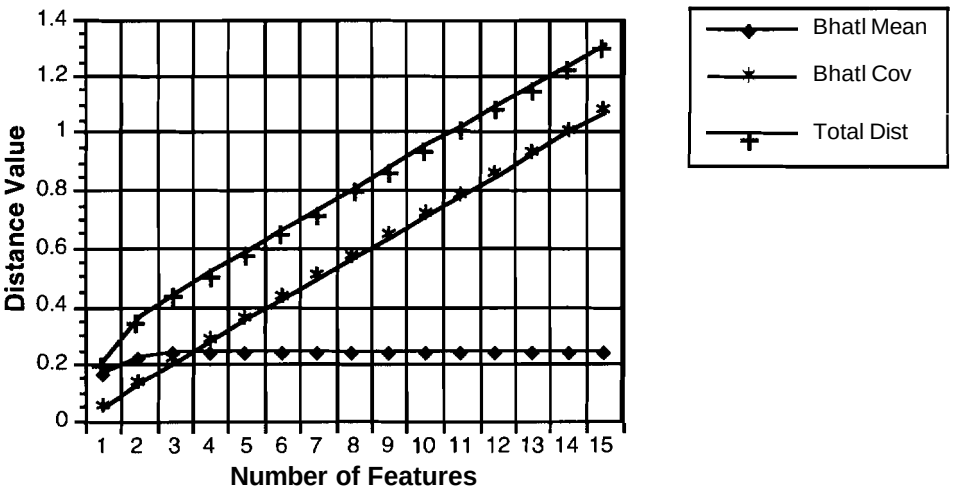


Fig. 2.13. Bhattacharyya distance and its mean and covariance components.

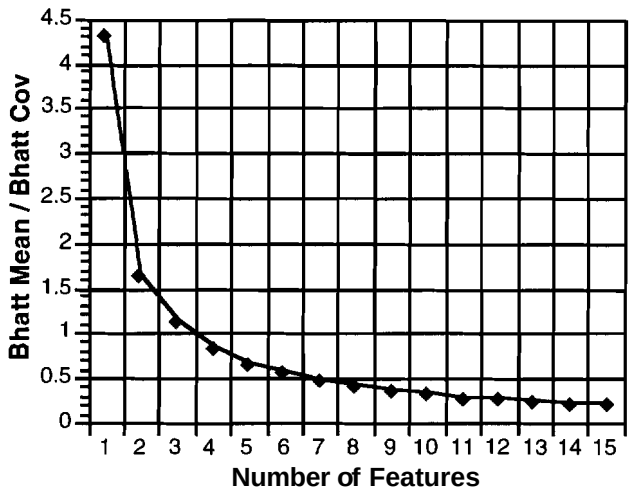


Fig. 2.14. Ratio of Bhattacharyya distance mean component over the covariance component.

Both figures show that there is a relation between second order statistics predominance and Bhatt Cov relevance. As the number of dimensions increases the ratio Bhatt Mean / Bhatt Cov decreases significantly and ML Cov classifier becomes more relevant than Min Dist. That shows that if as the dimensionality increases the ratio Bhatt Mean / Bhatt Cov decreases then second order statistics are more relevant in high dimensional data even when that could not be the case in low dimensionality.

Experiment 2

This experiment is similar to the previous one. The difference is in the fact that first order statistics are predominant in this case. The parameters of the two statistical classes are:

$$\mathbf{M}_1 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\mathbf{M}_2 = [1.5 \quad 1 \quad .5 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & & & & & & & 0 \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & & & & & & 1 \\ 0 & & & & & & & 1 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 2.5 & & & & & & & 0 \\ & 2 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ & & & & & & & 1 \\ 0 & & & & & & & 1 \end{bmatrix}$$

The classification results are shown in Figure 2.15. Observe that Min Dist classifier becomes more accurate than Min Cov after six dimensions.

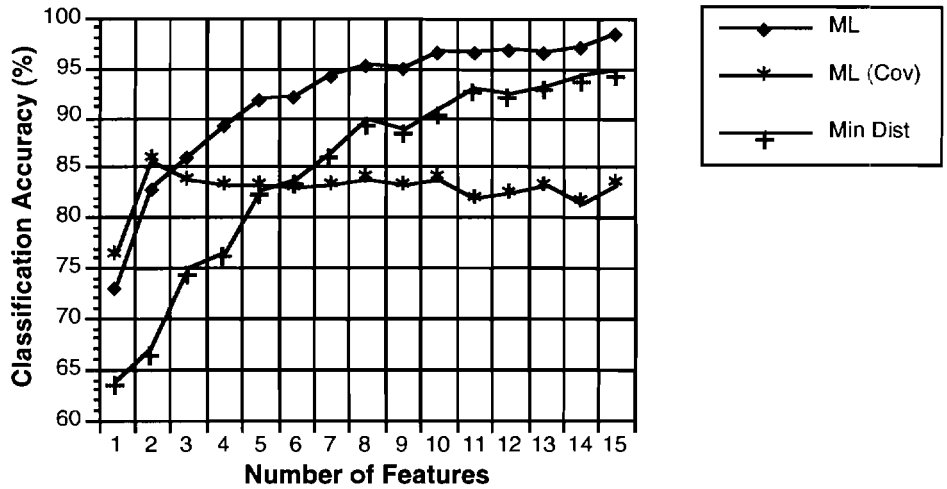


Fig. 2.15. Performance comparison of Normal ML, Normal ML with zero mean data, and Minimum Distance classifier. Two generated classes.

The mean (Bhatt Mean) and covariance (Bhatt Cov) components of Bhattacharyya distance and their sum were computed and are shown in Figure 2.16. Their ratio of Bhatt Cov / Bhatt Mean was calculated and shown in Figure 2.17. As the number of dimensions increases the ratio Bhatt Cov / Bhatt Mean decreases showing that first order statistics are more relevant in the classification of data.

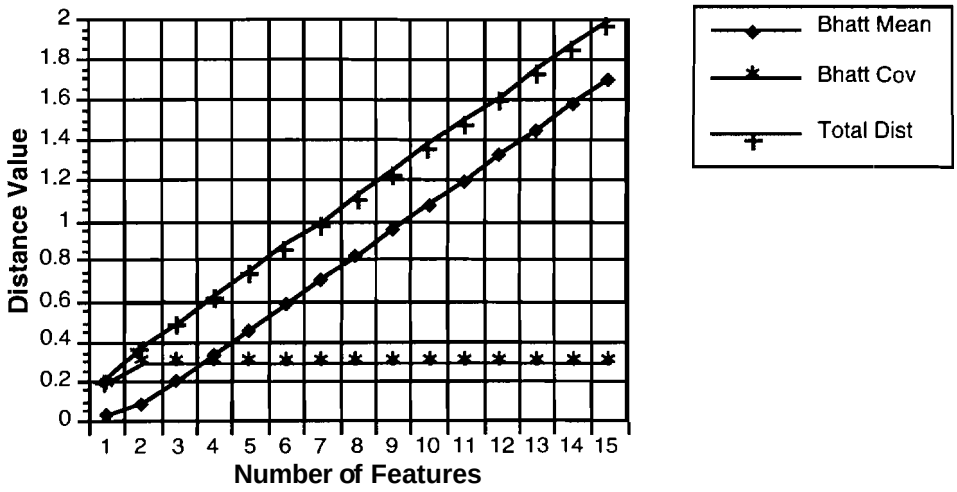


Fig. 2.16. Bhattacharyya distance and its mean and covariance components.

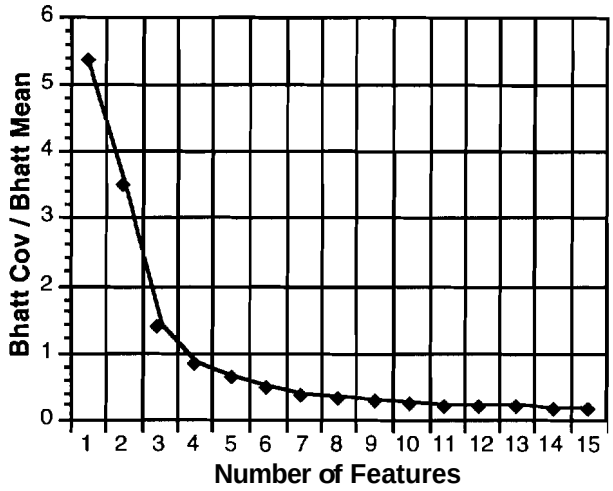


Fig. 2.17. Ratio of Bhattacharyya distance covariance component over the mean component.

The previous results show how the predominance of the mean or covariance Bhattacharyya distance components relates directly with first or second order statistics relevance in terms of classification accuracy. In the present work both components will be computed analytically and used to calculated upper bounds that will be functions of the number of dimensions. These bounds will be calculated for the case where the mean difference plays a predominant role and for the case where the covariance difference became predominant. Then the limits of the number of dimensions increment will be taken enabling one to understand the behavior of high dimensional data under such circumstances. That is the reason for dividing all the calculations into two cases: covariance predominance and mean predominance.

2.3.1 Case 1: Covariance difference as the dominant role in statistical class separability

Assume a two class problem where without loss of generality the first and second order statistics are:

$$\Sigma_1 = \begin{bmatrix} \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \sigma^2 \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} \alpha_1\sigma^2 & & 0 \\ & \ddots & \\ 0 & & \alpha_d\sigma^2 \end{bmatrix} \tag{2.18}$$

$$(M_2 - M_1) = \begin{bmatrix} \varepsilon_1 & \cdots & \varepsilon_k & \hat{\varepsilon}_{k+1} & \cdots & \hat{\varepsilon}_d \end{bmatrix}^T \tag{2.19}$$

Observe that every two covariance matrices can be simultaneously diagonalized to obtain the previous covariance matrices form [18]. That will enable us to have less complicated calculations without losing generality.

Under the conditions that:

(a) $\alpha_i \in (a, , a,)$, where $\alpha_{min} > 0$, and at least there exist an α_i such that $\alpha_i \neq 1$.

(b) $\varepsilon_{max} = \max_{\forall i \in (k+1,d)} \left(\left| \hat{\varepsilon}_i \right| \right)$ be such that $\varepsilon_{\max} \approx 0$.

(c) k = f(d) such that $\lim_{d \to \infty} \frac{k}{d} = 0$, (as an example $\forall \lambda > 0, d = k^{(1+\lambda)}$)

(d) $\varepsilon_i^2 \in (E_{\min}, E_{\max}), \forall i \in (1,k)$ and $E_{max} < \infty$ (to see the validity of this last assumption, see Appendix B).

Then as ***d*** increases the covariance contribution will dominate the Bhattacharyya distance.

Proof:

The means contribution to the Bhattacharyya distance can be written as (see Appendix A)

$$\mu_M = \frac{1}{8} \sum_{i=1}^{k} \frac{\varepsilon_i^2}{\overline{\sigma}_i^2} + \frac{1}{8} \sum_{i=k+1}^{d} \frac{\hat{\varepsilon}_i^2}{\overline{\sigma}_i^2} \leq \frac{1}{8} \sum_{i=1}^{k} \frac{\varepsilon_i^2}{\overline{\sigma}_i^2} + \frac{1}{8} \sum_{i=k+1}^{d} \frac{\varepsilon_{\max}^2}{\overline{\sigma}_i^2} \leq \hat{\mu}_M \tag{2.20}$$

Observe that α_{min} minimizes $(1+\alpha_i), \forall i$. Then

$$\hat{\mu}_M = \frac{k}{4\sigma^2(1+\alpha_{min})} \left(\frac{1}{k} \sum_{i=1}^{k} \varepsilon_i^2 \right) + \frac{d-k}{4\sigma^2(1+\alpha_{min})} \varepsilon_{max}^2 \tag{2.21}$$

Note that

$$\frac{1}{k} \sum_{i=1}^{k} \varepsilon_i^2 \leq \frac{1}{k} \sum_{i=1}^{k} E_{\max} = E_{\max} \tag{2.22}$$

with the consequence that

$$\hat{\mu}_M \leq \mu_{M\max} - \frac{k}{4\sigma^2(1+\alpha_{\min})} E_{\max} + \frac{d-k}{4\sigma^2(1+\alpha_{\min})} \varepsilon_{\max}^2 \tag{2.23}$$

The covariances contribution to the Bhattacharyya distance can be written as (see Appendix A):

$$\mu_C = \frac{1}{2} \sum_{i=1}^{d} ln \left(\frac{\sigma^2 + \alpha_i \sigma^2}{2\sigma^2 \sqrt{\alpha_i}} \right) = \frac{1}{2} \sum_{i=1}^{d} ln \left(\frac{1+\alpha_i}{2\sqrt{\alpha_i}} \right) \tag{2.24}$$

Let y be the argument that minimizes $\frac{1+\alpha_i}{2\sqrt{\alpha_i}}, \forall i$, subject to the constrain that $y \neq 1$. That argument must exist, based on the fact that $\alpha_i \in (\alpha_{min}, \alpha_{max})$, where $\alpha_{min} > 0$ and that $\exists i : \alpha_i \neq 1$. Then

$$\mu_C \geq \mu_{C min} = \frac{d}{2} ln \left(\frac{1+\gamma}{2\sqrt{\gamma}} \right) \tag{2.25}$$

Define a bound as

$$\Phi(\alpha_i, \varepsilon_i, d) = \frac{\mu_M}{\mu_C} \leq \frac{\mu_{M\max}}{\mu_{C\min}} = \Phi_{\max}(d) \tag{2.26}$$

where:

$$\Phi_{max}(d) = \frac{\frac{1}{4\sigma^2(1+\alpha_{min})}\left[kE_{max} + (d-k)\varepsilon_{max}^2\right]}{\frac{d}{2}ln\left(\frac{1+\gamma}{2\sqrt{\gamma}}\right)} \tag{2.27}$$

The quantity $\Phi_{max}(d)$ is an upper bound of $\Phi(\alpha_i, \varepsilon_i, d)$ and it can be rewritten as

$$\Phi_{max}(d) = \frac{\frac{k}{d}E_{max} + \frac{d-k}{d}\varepsilon_{max}^2}{2\sigma^2(1+\alpha_{min})ln\left(\frac{1+\gamma}{2\sqrt{\gamma}}\right)} \tag{2.28}$$

Finally taking the limit of d

$$\lim_{d\to\infty}\Phi_{\max}(d) = \frac{\varepsilon_{\max}^2}{2\sigma^2(1+\alpha_{\min})\ln\left(\frac{1+\gamma}{2\sqrt{\gamma}}\right)} \tag{2.29}$$

By the assumption that $\varepsilon_{max} \approx 0$, then $\lim_{d\to\infty}\Phi_{max}(d) \approx 0$. As a consequence

$$\lim_{d\to\infty}\Phi(\alpha_i, \varepsilon_i, d) \approx 0 \tag{2.30}$$

In conclusion, second order statistics and the hyper-ellipsoids shapes will play a more important role in discriminating among the classes than the means and the hyper-ellipsoids positions relative to one another.

Discussion

This proof only requires that $a_{,,} - \alpha_{min} > 0$ (differences in variances). It does not depend on how much this difference should be. The quantity $max|\varepsilon_i|$ can be as large as the physical devices permit. Also it only requires that $k = f(d) \ni lim_{d\to\infty}(k/d) = 0$, but it does not constrain the rate. In other terms, in low dimensional data the differences in covariance can be small and k = d and in terms of the mean such difference can be very large. In that case first order statistics will be more relevant in providing information than second order statistics in such low dimensional subspaces. But if as the dimension increases, the rate at which covariance information (even a small amount of information in low dimensional subspace) grows faster (nothing is said about how much faster) than the rate at which mean information grows (even large amounts of differences) then there will be a point where the total covariances information plays a more important role in discriminating among the classes than the means information.

2.3.2 Case 2: Mean differences as dominant in statistical class separability

Assume a two class problem, where without loss of generality, the first and second order statistics are:

$$\Sigma_1 = \begin{bmatrix} \sigma^2 & & & 0 \\ & \ddots & & \\ & & \sigma^2 & \\ & & & \ddots \\ 0 & & & \sigma^2 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} \alpha_1\sigma^2 & & & 0 \\ & \ddots & & \\ & & \alpha_k\sigma^2 & \\ & & & \hat{\alpha}_{k+1}\sigma^2 \\ 0 & & & & \ddots \\ & & & & & \hat{\alpha}_d\sigma^2 \end{bmatrix} \tag{2.31}$$

$$(M_2 - M_1) = [\varepsilon_1 \quad \cdots \quad \varepsilon_d]^T \tag{2.32}$$

Under the assumptions that:

(a) $a_i \in (a_{min}, a_{max})$ where $0 < a_{min} < a_{max} < \infty, \forall i \in (1,k)$.

(b) $\hat{\alpha}_i \in (1-\delta, 1+\delta), \forall i \in (k+1,d)$ where $\delta \approx 0$.

(c) $\varepsilon_i^2 \geq E_{\min} > 0, \forall i \in (1,d)$.

(d) $\lim_{d\to\infty}(k/d) = 0$, (as an example $\forall \lambda > 0, d = k^{(1+\lambda)}$).

As *d* increases, the means differences will dominate the Bhattacharyya distance.

Proof:

The means contribution to the Bhattacharyya distance can be written as (see Appendix A)

$$\mu_M = \frac{1}{4}\sum_{i=1}^{k}\frac{\varepsilon_i^2}{(1+\alpha_i)\sigma^2} + \sum_{i=k+1}^{d}\frac{\varepsilon_i^2}{(1+\hat{\alpha}_i)\sigma^2} \tag{2.33}$$

at the sarne time it can be written as:

$$\mu_M = \frac{k}{4\sigma^2}\left[\frac{1}{k}\sum_{i=1}^{k}\frac{\varepsilon_i^2}{(1+\alpha_i)}\right] + \frac{d-k}{4\sigma^2}\left[\frac{1}{d-k}\sum_{i=k+1}^{d}\frac{\varepsilon_i^2}{(1+\hat{\alpha}_i)}\right] \tag{2.34}$$

Note that the maximum of $(1+\hat{\alpha}_i) = (2+6)$ and that the maximum of $(1+\alpha_i) = (1+a_{max})$.

As a consequence

$$\mu_M \geq \mu_M = \frac{k}{4\sigma^2(1+\alpha_{max})}\left(\frac{1}{k}\sum_{i=1}^{k}\varepsilon_i^2\right) + \frac{d-k}{40^2(2+6)}\left(\frac{1}{d-k}\sum_{i=k+1}^{d}\varepsilon_i^2\right) \tag{2.31}$$

Observe that

$$E_{\min} \leq \frac{1}{m}\sum_{i=1}^{m}\varepsilon_i^2, \forall m \tag{2.36}$$

This implies that:

$$\hat{\mu}_M \geq \mu_{M\,min} = \frac{E_{min}}{4\sigma^2}\left(\frac{k}{1+\alpha_{max}} + \frac{d-k}{2+\delta}\right) \tag{2.37}$$

The covariance's contribution to the Bhattacharyya distance can be written as (see Appendix: A):

$$\mu_C = \frac{1}{2}\sum_{i=1}^{k} ln\left(\frac{1+\alpha_i}{2\sqrt{\alpha_i}}\right) + \frac{1}{2}\sum_{i=k+1}^{d} ln\left(\frac{1+\hat{\alpha}_i}{2\sqrt{\hat{\alpha}_i}}\right) \tag{2.38}$$

Let $\boldsymbol{a}$ be the argument that maximizes $(1+\alpha_i)/\left(2\sqrt{\alpha_i}\right), \forall i \in (1,k)$. Let $\hat{\alpha}$ be the argument that maximize $(1+\hat{\alpha}_i)/\left(2\sqrt{\hat{\alpha}_i}\right), \forall i \in (k+1,d)$, where $\hat{\alpha} \in (1-\delta, 1+\delta)$. Then

$$\mu_C \le \mu_{C\,max} = \frac{k}{2} ln\left(\frac{1+\alpha}{2\sqrt{\alpha}}\right) + \frac{d-k}{2} ln\left(\frac{1+\hat{\alpha}}{2\sqrt{\hat{\alpha}}}\right) \tag{2.39}$$

Define a bound

$$P(\alpha_i, \varepsilon_i, d) = \frac{\mu_C}{\mu_M} \le \frac{\mu_{C\max}}{\mu_{M\min}} = P_{\max}(d) \tag{2.40}$$

Substituting equations, tlie upper bound $P_{\max}(d)$ will be calculated as:

$$P_{max}(d) = \frac{\frac{k}{2} ln\left(\frac{1+\alpha}{2\sqrt{\alpha}}\right) + \frac{d-k}{2} ln\left(\frac{1+\hat{\alpha}}{2\sqrt{\hat{\alpha}}}\right)}{\frac{E_{min}}{4\sigma^2}\left(\frac{k}{1+\alpha_{max}} + \frac{d-k}{2+\delta}\right)} = \frac{\frac{2\sigma^2}{E_{min}}\left[\frac{k}{d-k} ln\left(\frac{1+\alpha}{2\sqrt{\alpha}}\right) + ln\left(\frac{1+\hat{\alpha}}{2\sqrt{\hat{\alpha}}}\right)\right]}{\left(\frac{\frac{k}{d-k}}{1+\alpha_{max}} + \frac{1}{2+\delta}\right)} \tag{2.41}$$

Taking the limit as $\boldsymbol{d}$ tends to infinity:

$$\lim_{d\to\infty} P_{max}(d) = \frac{2\sigma^2(2+\delta)}{E_{min}} ln\left(\frac{1+\hat{\alpha}}{2\sqrt{\hat{\alpha}}}\right) \tag{2.42}$$

Observe that because $\boldsymbol{6} = 0$ then $\hat{\alpha} \approx 1$ *and* $\lim_{d+-} P_{max}(d) \approx 0$. As a consequence

$$\lim_{d\to\infty} \sim (\alpha\varepsilon_i, \mathcal{J}) \approx 0 \tag{2.43}$$

In conclusion then, for the conditions specified in this case, first order statistics and the hyper-ellipsoids positions relative to one another will play a more important role than second order statistics and the hyperellipsoid shape.

Discussion

This proof only requires that $\varepsilon_i^2 \ge E_{\min} > 0, \forall i \in (1,d)$. It does not require a limitation on how large Emin should be. $a_{\prime\prime}$ could be as large as the physical devises will allow. Also it requires that $lim_{d\to\infty}(k/d) = 0$, but it does not constraiin how the limit should approach zero. Even if in low dimensional data, where $\boldsymbol{k \approx d}$, the covariance difference is very large and dominates over the means, if as the dimensionality increases, the rate at which means differences (even small differences) grows faster than the covariance one, then there will be a point where the total mean differences will provide more information for classes discrimination than covariances differences.

2.4 High Dimensional Characteristics Implications for Supervised Classification

Based on the characteristics of high dimensional data that the volume of hypercubes have a tendency to concentrates in the corners, and in a hyperellipsoid in an outside shell, it is apparent that high dimensional space is mostly empty, and multivariate data is usually in a lower dimensional structure. As a consequence it is possible to reduce the dimensionality without losing significant information and separability. Due to the difficulties of density estimation in nonparametric approaches, a parametric version of data analysis algorithms maybe expected to provide better performance where only limited numbers of labeled samples are available to provide the needed a priori information.

The increased number of labeled samples required for supervised classification as the dimensionality increases presents a problem to current feature extraction algorithms where computation is done at full dimensionality, e.g. Principal Components, Discriminant Analysis and Decision Boundary Feature Extraction [10]. A new method is required that, instead of doing the computation at full dimensionality, it is done in a lower dimensional subspace. Performing the computation in a lower dimensional subspace that is a result of a linear projection from the original high dimensional space will make the assumption of normality better grounded in reality, giving a better parameter estimation, and better classification accuracy.

A preprocessing method of high dimensional data based on such characteristics has been developed based on a technique called Projection Pursuit. The preprocessing method is called Parametric Projection Pursuit [19] [20].

Parametric Projection Pursuit reduces the dimensionality of the data maintaining as much information as possible by optimizing a Projection Index that is a measure of separability. The projection index that is used is the minimum Bhattacharyya distance among the classes, taking in consideration first and second order characteristics. The calculation is performed in the lower dimensional subspace where the data is to be projectedl. Such preprocessing is used before a feature extraction algorithm and classification process as shown in Figure 2.18.

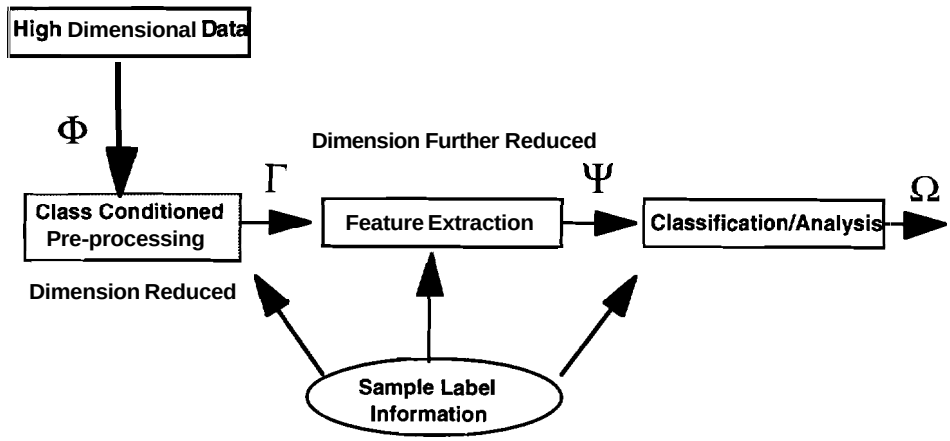


Fig. 2.18. Classification of high dimensional data including preprocessing of high dimensional data.

In Figure 2.18 'the different feature spaces have been named with Greek letters in order to avoid confusion. Φ is the original high dimensional space. Γ is the subspace resulting from a class-conditional linear projection from Φ using a preprocessing algorithm, e.g. Parametric Projection Pursuit. Ψ is the result of a feature extraction method. Ψ could be projected directly from Φ or, if preprocessing is used, it is projected from Γ. Finally Ω is a one dimensional space that is a result of classification of data from Ψ space. Note that the three procedures, preprocessing, feature extraction and classification use labeled samples as a priori information.

2.5 Conclusion

In this section we will consider some implications of what has been discussed for supervised classification. In terms of parameter estimation, a large number of samples are required to make a given estimation in multispectral data to adequate precision. In a nonparametric approach, the number of samples required to satisfactorily estimate the density is even greater. Both kinds of estimations confront the problem of high dimensional space characteristics. As a consequence, it is desirable to project the data to a lower dimensional space where high-dimensional geometric characteristics and the Hughes phenomena are reduced. Commonly used techniques such as Principal Components, Discriminant Analysis, and Decision Boundary Feature Extraction have the disadvantage of requiring computations at full climensionality in which the required number of labeled samples is very large. The procedures use estimated statistics that are not necessarily accurate. Another problem is the

assumption of normality. Nothing guarantees that at full dimensionality, that model fits well.

It has been shown that high dimensional spaces are mostly empty, indicating that the data structures involve exist primarily in a subspace. The problem is which subspace it is to be found in is situation-specific. Thus the goal is to reduce the dimensionality of the data to the right subspace without losing separability information. The approach is to make the computations in a lower dimensional space, i.e. in Γ instead of Φ, where the projected data produce a maximally separable structure and which, in turn, avoids the problem of dimensionality in the face of the limited number of training samples. Further, a linear projection to a lower dimensional subspace will make the assumption of normality in the Γ subspace more suitable than in the original Φ. In such a lower dimensional subspace any method used for feature extraction could be used before a final classification of data, even those that have the assumption of normality.

In remote sensing data analysis the best projection would certainly be the one that separates data into different meaningful clusters that are exhaustive, separable, and of information value [2, pp. 340]. A measure of separability among different statistical classes is thus needed. Based on what has been studied, it should take into consideration First order and second order statistics. Methods used in low dimensional subspaces to see which one could predominate, e.g. histograms or any other density estimation procedure, will not necessarily work in high dimensional data as section 2.3 shows.

3. PROJECTION PURSUIT, DIMENSIONAL REDUCTION AND FEATURE EXTRACTION

3.1 Introduction

In the last chapter it was shown why it is desirable to reduce the dimensionality of the multispectral data in a preprocessing step. As indicated in Figure 2.18 this preprocessing should be before the use of a feature extraction algorithm in order to make the analysis and the estimation of parameters more effective. This is due to the limited number of training samples, the Hughes Phenomenon and the geometrical and statistical properties of data in high dimensional space. It was shown that care should be taken with the assumption of normality and that the preprocessing method should avoid doing the computation in the high dimensional space. Instead, the computations should be done in a lower dimensional space to produce better parameter estimation.

Dimensional reduction is a process of projecting the data from an original space to a lower-dimensional subspace having more effective features. In statistical pattern recognition effective features are those most capable of preserving class separability [18, pp. 441]. It is well known that class separability among distributions is preserved in any nonsingular linear transformation. What is required is a transformation in which full separability among distributions is preserved as much as possible in a lower dimensional subspace. That transformation must reduce the dimensionality by searching for the subspace that preserves class separability as much as possible. Implied in the previous statement is the requirement for optimizing with respect to a measure of class separability. This measure of class separability should consider both first and second order statistics.

3.2 Feature Extraction Algorithm Overview

In order to understand what characteristics a preprocessing algorithm should have (second block in Figure 2.18) we studied the properties of high dimensional data (first block). In the present section a survey of commonly founcl dimensionality

reduction algorithms will be presented. These procedures have traditionally been used as feature extraction methods in relatively low dimensional data. One objective here is to study their properties and see if they fulfill the requirement that preprocessing must have in high dimensional data. Another is to see how a feature extraction method should relate to a preprocessing block.

3.2.1 Principal Components

This method assumes that the distribution takes the form of a single hyperellipsoid such that its shape and dimensionality can be determined by the mean vector and covariance matrix of the distribution [8, pp. 206]. This can be done by observing the eigenvalues of the positive definite covariance matrix, *C,* of the total multispectral data set. Writing Σ in its spectral representation we have $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_d]$ and:

$$\Sigma = A\Lambda A^T \tag{3.1}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \ddots \\ 0 & & \lambda_d \end{bmatrix} \tag{3.2}$$

The $\mathbf{a}_i$'s are the eigenvectors corresponding to the eigenvalues hi. The eigenvalues are ordered as: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$. The method comprises a linear transformation of the original data X into a new space Y, where $\mathbf{Y}_i = \mathbf{a}_i^T(\mathbf{X}-\mu_X)$. The $\mathbf{a}_i$ are then selected to reduce the dimensionality by choosing a d' < d such that:

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > p_{\%} \tag{3.3}$$

where $p_{\%}$ is some arbitrarily selected proportion of the total cumulative eigenvalue sum.

A problem with this method is that it treats the data as if it is a single distribution. Our goal is to divide this data into different distributions that represent different statistical classes, thus our requirement is to base this division upon class separability,

a factor that this method ignores. As a consequence this method could merge different classes necessarily harming classification accuracy. Though the computation of Σ is at full dimensionality this may not be a limitation in this case, since all data, not just the training samples may be used.

3.2.2 Feature subset selection

Some authors have proposed algorithms by which a subset of ,features can be chosen from the original set [2, pp. 164]. This requires a comparison between statistical distance measurements among the classes. The features that provide the largest statistical separability will be chosen. Among the measurements used for statistical separability are Divergence, Bhattacharyya, Jeffreys-Matusita, Cramer-Van Mises, Kierfer-Wolfowitz, Kolmogorov Variational, Kullback-Liebler Numbers, Mahalanobis, Samuels-Bachi, and Swain-Fu [21].

One type of feature subset selection, proposed by [22], uses an automatic band selection algorithm based on Markov chain theory. Applying this statistical theory and a quality criterion, the algorithm selects a near optimal set of bands to be used for classification purposes. The quality criterion is based on interclass dlistance or error rate estimation.

A problem with feature subset selection is that it considers a subset of all linear combinations. Consequently it can be optimum in that subset only. In order that a feature selection algorithm be optimal, the search for a subset of features has to be exhaustive [23]. The number of combinations of bands increases exponentially as the dimensionality increases and, as a result, an exhaustive search demands a very large number of computations.

3.2.3 Discriminant Analysis

In terms of classification using the Bayes classifier, Bayes error becomes the class separability criterion to measure feature effectiveness [18, pp. 441]. The major problem with this criterion is that a closed mathematical expression is available for only a few special cases. Even when it exists, the calculation of Bayes error demands numerical integration [18, pp. 87-90]. That is why other, simpler criteria had been used based on a mathematically closed form. One of those criteria used is Canonical Analysis [1, pp. 216]. In this method a series of vectors $\mathbf{a_i}$'s are calculated so they will maximize a criterion function called the Fisher ratio. Such a function is:

$$I(\boldsymbol{a}) = \frac{\boldsymbol{a}^T \Sigma_b \boldsymbol{a}}{\boldsymbol{a}^T \Sigma_w \boldsymbol{a}} \tag{3.4}$$

where

$$\Sigma_w = \sum_{i=1}^{M} P_i \Sigma_i \tag{3.5}$$

$$\Sigma_b = E\{(M_i - M_0)(M_i - M_0)^T\} = \sum_{i=1}^{M} P_i (M_i - M_0)(M_i - M_0)^T \tag{3.6}$$

$$M_0 = \sum_{i=1}^{M} P_i M_i \tag{3.7}$$

Equation (3.5) is the average within class covariance matrix. Equation (3.6) represents the between class covariance matrix and (3.7) is the overall mean.

One of the problems with this method is that if the difference in the class mean vectors is small the features chosen will not be reliable. If one mean vector is very different from the others, its class will eclipse the others in the computation of the between class covariance matrix. As a consequence, the feature extraction process will be ineffective. Another problem with this method is that for a case of M classes a maximum of M-1 features can be extracted, limiting the final dimensionality independently of class separation.

Fisher Ratio Discriminant Analysis Modifications

Some modifications have been performed on the Fisher ratio in order to obtain a variation of the Discriminant Analysis Canonical procedure. Two of those modifications are Orthonormal Discriminant Vectors (ODV) and Multidimensional Data Mappings.

Orthonormal Discriminant Vector

This method [24] uses the Fisher ratio criterion and sequentially extracts the features optimizing the criterion under the constraint of orthonormality, i.e. $a_i^T a_j = \delta_{ij}$. Where δ_{ij} is the Kronecker delta. Contrary to Canonical Analysis that, for an M class problem, can only calculate up to M-1 features, ODV can calculate .as much as d-1 features where d is the number of dimensions in the original feature space. A single modification of ODV based on a modified plus e-take away f algorithm was developed [25]. This modified ODV has a mechanism to remove the superfluous features automatically. It has been proved theoretically that this method performs better than Discriminant Analysis in terms of the Fisher criterion [26].

Parametric and Nonparametric Multidimensional Data Mappings.

This method [27] uses a modification of the previous criterion function that is an extension of Malina's class distance. Such criterion for two classes is:

$$I(a)=\frac{(1-\beta)a^T Va+\beta\left|a^T W^{(-)}a\right|}{a^T Wa} \tag{3.8}$$

Where β is a supplied scalar, **V** is a between-class scatter matrix (corresponding to Σ_B in the parametric case), W is a class independent scatter matrices (corresponding to Σ_W in parametric case), $\mathbf{W}^{(-)}$ is the difference between within class scatter matrices (which is the difference of covariance matrices in the parametric case). The authors use it to map high dimensional data from $\mathbf{R}^n$ to $\mathbf{R}^2$ or $\mathbf{R}^3$ for a two class case. It has the advantage of being flexible enough, in terms of the parameters, to obtain known projections and produce new ones, parametric and nonparametric. The disadvantage is that it has been derived for the two class case, and projected to $\mathbf{R}^2$ or $\mathbf{R}^3$. Even if it would be generalized, it shares the same disadvantage as Canonical Analysis. Another problem is the estimation of some control parameters. The authors think that many parameter must be tested to obtain appropriate ones.

Compared with Principal Component Analysis, these Discriminant Analysis methods have the advantage that class separability in terms of the Fisher criterion is explicitly used in the calculation. The major disadvantage is that parameters must be estimated at full dimensionality, where they are not necessarily accurate. As a consequence the vectors $\mathbf{a}_i$'s are not necessarily suitable for clusters separation.

3.2.4 Decision Boundary feature extraction algorithm

Lee and Landgrebe [10] proposed an algorithm based on decision boundaries that predicts the number of features necessary to achieve the same classification accuracy as in the original space. This algorithms has the advantage that it finds the necessary feature vectors.

Let **X** be an observation in the d-dimensional space. Under a Bayes decision rule with the hypothesis Hi, i = 1, 2, the decision will be made according to the rule:

$$\boldsymbol{X}\in\omega_1, \text{ if } h(\boldsymbol{X})<t, \text{ otherwise } \boldsymbol{X}\in\omega_2 \tag{3.9}$$

where:

$$h(\boldsymbol{X})=-ln\left(\frac{P(\boldsymbol{X}\mid\omega_1)}{P(\boldsymbol{X}\mid\omega_2)}\right) \tag{3.10}$$

$$t=ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right) \tag{3.11}$$

Let $\mathbf{X}^*$ be the projected vector of X in a subspace W. That subspace should have the characteristic that for any observation X:

$$(h(\boldsymbol{X})-t)\left(h(\boldsymbol{X}^*)-t\right)>0 \tag{3.12}$$

The physical meaning of the above equation is that the classification result of X* is the same as X. The proposed algorithm finds the minimum dimension of a subspace such that this inequality holds for the given observations and finds the features that produce such a projection.

This algorithm has been applied successfully. Its only problem is that it demands a high number of training samples for high dimensional space. This occurs because it compute:; the class statistical parameters at full dimensionality. The authors suggested, for a further development, an algorithm that will pre-process the data in order to reduce the dimensionality before using this algorithm [11, pp. 206-209].

3.2.5 Significant Weighted Supervised feature extraction

Kiyasu and Fujimura [28] discuss an algorithm based on a significance weighting approach. The algorithm first reduces the data using Principal Components Analysis. Then, it weights the classes in such a way that one feature can be used to separate a particular pair of classes without considering other pairs. Finally it will check if that feature separates all the other pairs of classes. If it does not work for a specific set of pairs, the process will be repeated for that particular group of classes.

There are several problems with this method. First, it assumes that one feature is enough to separate two classes. Second, one has to order the classes, which requires that some criteria be developed. Third, each time a new feature for any pair of classes is found, one must check whether it separates the other pairs sufficiently. Therefore the separabiliity of every pair of classes must be checked more than once. As the number of classes increase the computations rise exponentially.

3.2.6 Discriminative Feature Extraction

Bienn and Katagiri [29] tried to minimize the classification error using a discriminative learning theory. Under the assumption that classification is done pursuing the minimum Bayes risk, this method tries to estimate the features optimizing an index that directly minimize the classification error. It estimates the feature extraction parameter as well as the classification parameters at the same time optimizing a function of the global loss that is an index of misclassifications. Such optimizatiion is performed by a gradient search algorithm and an iterative approach.

The problem with this method is that such algorithm must estimate other parameters outside the feature extraction and classification ones. Because it is an iterative approach it has to performs a lot of classifications and feature extraction

estimatioris which are time consuming. It does the computation at full dimensionality, leaving the problem of having small number of label samples unsolved.

All the techniques discussed above have some advantages and some disadvantages. Among the disadvantages the most significant are (1) that the computations are performed at full dimensionality and (2) that the number of computatiions is quite high. The first disadvantage is related to the problems of high dimensional space and its estimations of parameters or densities. The second is related to computational efficiency.

We next discuss a technique named Projection Pursuit which has the advantage of making the computations in a lower dimensional subspace where an "interesting" projection will occur. It is flexible enough to allow the analyst to define what "interesting" means, making it useful for a variety of different purposes. We will use it to develop an algorithm to preprocess the data before engaging in final feature extraction and classification processes (see Figure 2.18).

3.3 Projection Pursuit

3.3.1 Definition

Projection Pursuit has been defined as [8, pp. 208-2121 "... the numerical optimization of a criterion in search of the most interesting low-dimensional linear projection of a high dimensional data cloud." In the original idea Projection Pursuit is used to select potentially interesting projections by the local optimization over projection directions of some index of interestingness. This introduces the challenge of how to characterize "interestingness" in a numerical fashion. Projection Pursuit automatically picks an "interesting" lower dimensional projection from high dimensional data by maximizing or minimizing a function called the projection index. This technique is able to bypass many of the problems of high dimensionality by making the computations in a lower dimensional subspace.

The idea of a projection index other than variance was discussed in the late sixties and early seventies. The first successful implementation was done by Friedman and Tukey [30]. The idea had been extended to projection pursuit regression [31] [32], and projection pursuit density estimation [33] [9]. Huber worked on the connection between projection pursuit and some other fields like computer tomography, time series, arid finite sample implementations [34].

For a mathematical interpretation, define the following vectors and functions:

X is the initial multivariate data set (dxN). In multispectral data, we refer to N elements consisting of d bands. A geometrical representation will imply that it is a set containing N data points in a d-dimensional space.

Y is the resulting dimensionally reduced projected data (mxN). A is the parametric orthonormal matrix (dxm) where $Y = \mathbf{A}^T\mathbf{X}$. Projection Pursuit is the method that computes A optimizing the projection index $I(\mathbf{A}^T\mathbf{X})$. Sometimes the projection index is written in the form $I(\mathbf{A})$ or $I(\mathbf{a})$ in cases having a parametric vector instead of a matrix.

3.3.2 Projection Pursuit and engineering applications

This technique has been applied in different areas of engineering. In the area of robotics it has been used in order to improve a robot's navigating system [35]. In that work the authors estimate the direction and configuration in the two dimensional path of the robot from the one dimensional data with the goal that the area of uncertainty of location has a Gaussian distribution with a small variance when projected to one dimension.

In the area of neural networks it has been applied in numerous occasions. It has been demonstrated that there exists a connection between the BCM learning procedure and Projection Pursuit [36] [37]. A projection index was developed as an objective function which is the expected value of the loss function of the neurons. Its minimization projects the data far from a Gaussian distribution. The projection index is

$$I(\boldsymbol{a}) = E\left[L_a(X)\right] \qquad \textbf{(3.13)}$$

where

$$L_a(X) = -\mu\left\{\frac{1}{3}\left(a^T X\right)^3 - \frac{1}{4}E\left[\left(a^T X\right)^2\right]\left(a^T X\right)^2\right\} \qquad (3.14)$$

The variable μ represents the learning rate. Jones [38] developed a Projection Pursuit learning network by approximating the target function $f(\mathbf{X})$ by the neural network output $o(\mathbf{X})$, where:

$$\mathbf{f}(X) \approx o(X) = \sum_{i=1}^{q} I_i\left(a_i^T X\right) \qquad \textbf{(3.15)}$$

and each projection index is defined as

$$I_i\left(a_i^T X\right) = \lambda_i \frac{e^{\alpha_i\left(a_i^T X\right)+\beta_i}}{1+e^{\alpha_i\left(a_i^T X\right)+\beta_i}} \qquad (3.16)$$

The $\mathbf{a}_i$'s are chosen to best approximate (3.15)

In terms of remote sensing data, Nason used the technique with multispectral images to project data to a 3-dimensional space corresponding to red, green, and

blue. That projection produces a scene on the screen that allows for a more exact human interpretation [39].

3.3.3 Projection index

The choice of the projection index is the most critical aspect of this technique. What "interesting" means depends on what function or projection index one uses. In remote sensing data analysis "interesting" would certainly be a projection which separates data into different meaningful clusters which are exhaustive, separable, and of information value [2, pp. 340].

Many nonparametric projection indices have been proposed with the purpose of maintaining the distance among the clusters. The Friedman-Tukey index is the "result of constructing a kernel density estimate from the projected data point and then summing its values at those data points" [40]. Let Y = $\mathbf{a}^T\mathbf{X}$, where a is a vector, then:

$$I(\boldsymbol{a}^T\boldsymbol{X}) = d(\boldsymbol{a}) = \int \hat{f}(Y)dF_N(Y) \tag{3.17}$$

where $\hat{f}(Y)$ is the kernel estimate and F_N is the empirical distribution of the projected data. Jones and Sibson show that maximization of this index emphasizes a large departure from a parabolic density function form rather than specific instances of clustering.

Other nonparametric indices were proposed because of their special properties. Among these are the Standardized Fisher (3.18) and the negative Shannon entropy (3.19) [8, pp. 210]:

$$\sigma^2(Y)\int \frac{f'(Y)^2}{f(Y)}dY \tag{3.18}$$

$$\int f(Y)log(Y)dY \tag{3.19}$$

After the data have been spherized both indices have the property that each is minimized at the normal density with the same mean and standard deviation. It is well known in Information Theory that entropy is maximized by the Normal distribution [41]. Maximizing the negative entropy index will thus give the least normal projection. This type of linear projection would be expected to produce a multimodal density with the consequence of maximizing the separation among clusters.

Peter Hall [42] discussed two other indices for density estimators and regression. The first one, named Friedman's, index is:

$$I(\boldsymbol{a}) = \int_{-1}^{1}\left(f_{\boldsymbol{a}}(u) - \frac{1}{2}\right)^2 du \tag{3.20}$$

where $U_a = 2\Phi(Y) - 1$ and $Y = \mathbf{a}^T\mathbf{X}$. Note that Y is normal if U is uniform. As a consequence the maximization of I(**a**) is a departure from normality. 'The otlier index

proposed by Hall is the L^2 distance between the density of $Y = \mathbf{a}^T\mathbf{X}$ arid the standard normal density $\phi(y)$:

$$I(\boldsymbol{a}) \equiv \int_{-\infty}^{\infty} \left(f_Y(y) - \phi(y)\right)^2 dy \qquad (3.21)$$

Optirnizing the indices implies a recalculation and a numerical integration of them, which becomes difficult as the number of dimensions in Y increases. To overcome this, it has been proposed to estimate the indices by a series of polynomial estimations from the data. Huber suggested the use of a Moment index that is an approximation to Shannon entropy [34]. The index is based on the third and fourth sample moments of the projected data and was computed by Jones and Sibson [40]. Friedman and Hall used a series of orthonormal polynomials. The Friedman's index used a normalize Legendre polynomial sequence estimation. Hall's index used a Hermite polynomial series. In all of these, the series must be truncated to a number that needs to be estimated.

The indices just discussed have five main disadvantages. The first is that the data must be centered at zero and spherized in order to spread the data equally in all directions. That action causes an enhanced contribution from noisy variables. The second disadvantage is that these indices are suitable only for nonparametric approaches which wastes a priori information. Consequently, these indices do not allow sufficient flexibility to the analyst in order to define what interesting means on a case-by-case basis. The third disadvantage is that the techniques requires a lot of data in order to estimate the Moment index, the polynomial series elements, or the number of elements of the truncated series of orthogonal polynomials. The fourth disadvantage is that classes are not defined, and as a result statistical distance is not explicitly delimited. The fifth is that it is not clear how to estimate the final number of features to preserve as much information as required.

3.4 Parametric Projection Pursuit

3.4.1 A parametric approach

Taking into consideration the disadvantages of the nonparametric projection indices discussed above, a parametric approach will be proposed in the present work. The analyst will use labeled samples in order to define classes explicitly. In addition, a convenient statistical distance among the classes plus some constraints on matrix **A** will give sufficient flexibility for the development of a projection index that will imply a convenient definition of "interesting", as shown in Figure 3.1.

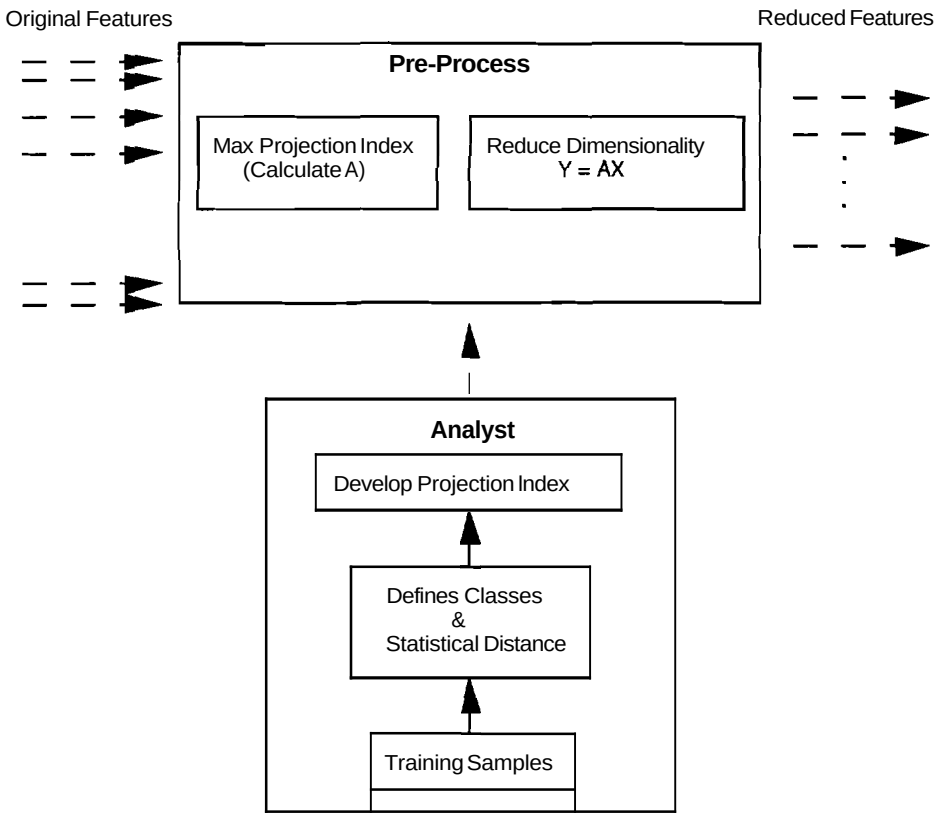


Fig. 3.1. A possible projection pursuit scheme for the remote sensing circumstance.

Discriminant Analysis and Parametric Projection Pursuit are similar processes in terms of optimizing a criterion function $I(\mathbf{a}^T\mathbf{X})$ analytically or numerically. The main difference with Discriminant Analysis is the order of the process as shown in Figures 3.2 and 3.3.

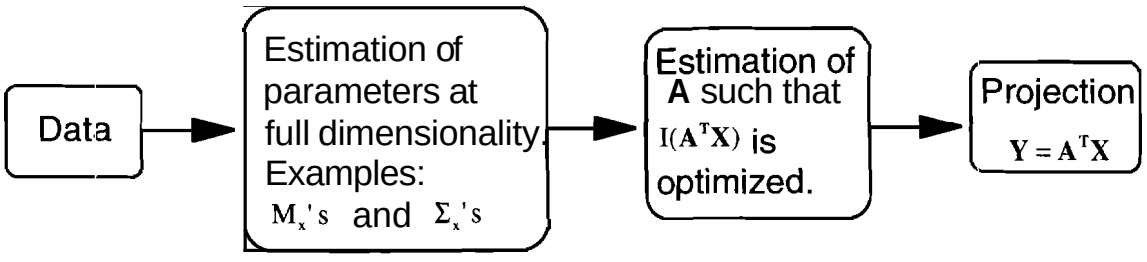


Fig. 3.2. Discriminant Analysis process order

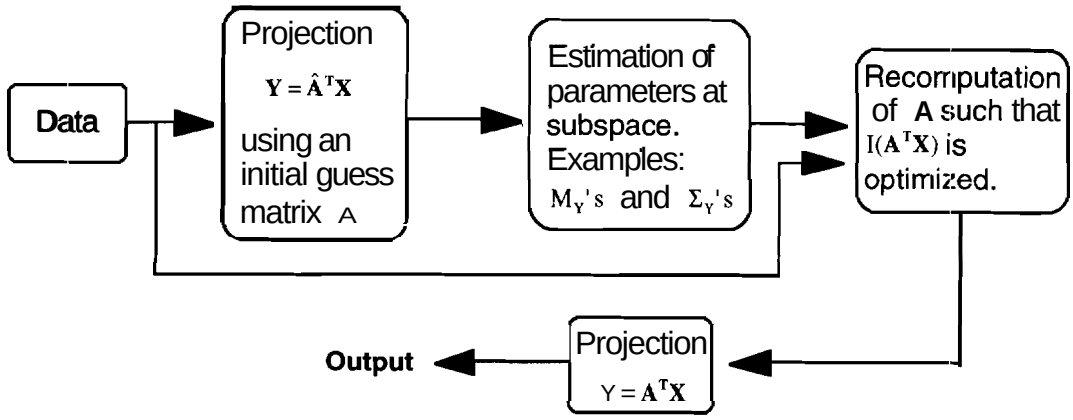


Fig. 3.3. Projection Pursuit process order.

Observe that Projection Pursuit starts with an a priori matrix $\hat{A}$, then the parameters in a low dimensional space are estimated and matrix A is recomputed by optimizing the projection index $I(A^T X)$. Because the optimization is performed in a low dimensional subspace, a numerical method is needed. Note that the parameters in Projection Pursuit are functions of the parametric matrix A. Discriminant Analysis is the opposite, A is a function of the parameters. The computations at a lower dimensional space enables this method to better handle the problem of small numbers of samples, the Hughes phenomena, high dimensional geometrical and statistical properties, and the assurription of normality as previously mentioned.

3.4.2 Parametric projection indices

Bo [43] proposed the use of a parametric index for the two class problem, defined as:

$$I(A) = \frac{B(A)}{W(A)} \tag{3.22}$$

where

$$B(A) = \left(A^T M_1 - A^T M_2\right)^T \left(A^T M_1 - A^T M_2\right) \tag{3.23}$$

$$W(A) = trace\left(A \Sigma_1 A^T + A \Sigma_2 A^T\right) \tag{3.24}$$

This index tries to maximize the difference in the means and redluce the scatter within the same class. It has the advantage of having a closed solution and a procedure of estimating the final number of features. But it has the disadvantage of not being related, directly or as a bound, with classification accuracy. Also it must make the computation at full dimensionality, reducing the method to a discriminant analysis method with a projection index different from the Fisher criterion. The computation at full dimensionality entails the problem already discussed of estimating the parameters

with a small number of training samples producing a lack of accuracy in terms of the estimated features.

With the objective of enhanced classification accuracy we proposed the use of Bhattacharyya distance among two classes because of its relationship with classification accuracy and it uses of first and second order statistics as discussed in chapter 1 [18, pp. 99-109]. Such an index for the two class case is:

$$I(A^T X) = \frac{1}{8}(M_{2Y} - M_{1Y})^T \left(\frac{\Sigma_{1Y} + \Sigma_{2Y}}{2} \right)^{-1} (M_{2Y} - M_{1Y}) + \frac{1}{2} ln \left(\frac{\left| \frac{\Sigma_{1Y} + \Sigma_{2Y}}{2} \right|}{\sqrt{|\Sigma_{1Y}||\Sigma_{2Y}|}} \right) \quad (3.25)$$

In the case of more than two classes the minimum Bhattacharyya distance among the classes could be used:

$$I(A^T X) = \min_{i \in C} \left\{ \frac{1}{8}(M_{2Y}^i - M_{iY}^i)^T \left(\frac{\Sigma_{iY}^i + \Sigma_{2Y}^i}{2} \right)^{-1} (M_{2Y}^i - M_{1Y}^i) + \frac{1}{2} ln \left[\frac{\left| \frac{\Sigma_{1Y}^i + \Sigma_{2Y}^i}{2} \right|}{\sqrt{|\Sigma_{1Y}^i||\Sigma_{2Y}^i|}} \right] \right\} (3.26)$$

C is the number of combinations of group of two classes. Assuming there are L classes then:

$$C = \frac{L!}{2!(L-2)!} \quad (3.27)$$

From ground truth information the analyst can define the classes and estimate the mean and covariance of each. As an example, consider two sets of training samples in 2-dimensional space. The first appears in Figure 3.4. Both data sets are samples from normal distributions. The parameters of the data are:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \text{ and } \Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

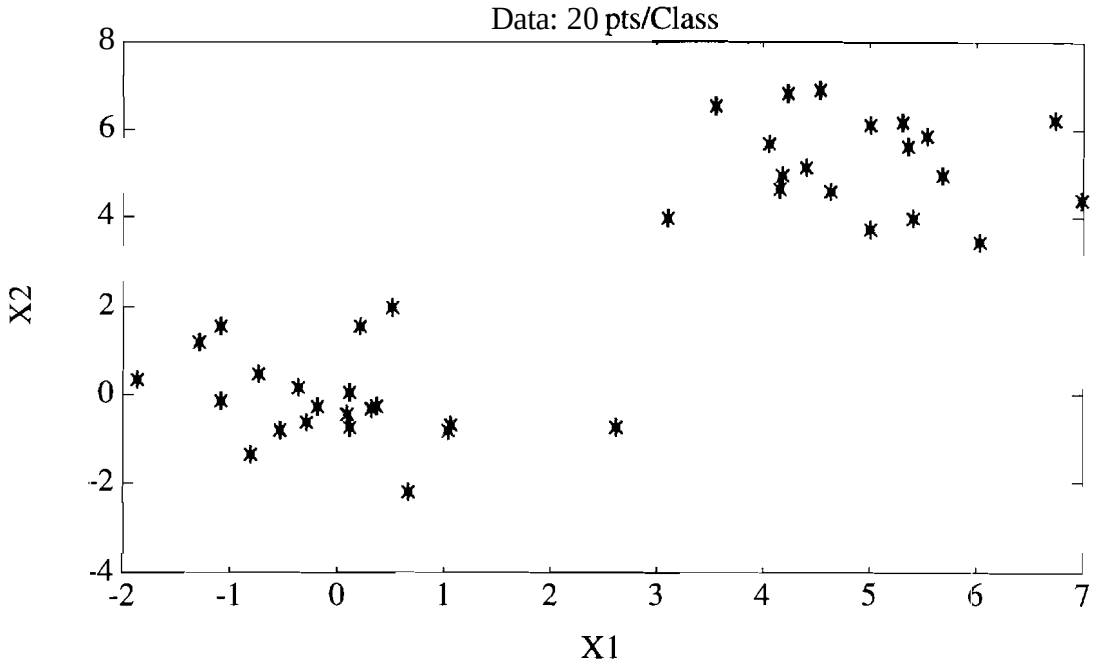


Fig. 3.4. Example two dimensional normally distributed data.

Parametric Projection Pursuit calculates the angle at which the vector $a = [\cos(\theta) \quad \sin(\theta)]$ maximizes the projection index of the projected data (I-dimension). From the projected training samples the means and variances in one dimension can be estimated. The negative of the Bhattacharyya distance was used as a statistical distance and as the projection index. Therefore we want in this case to minimize the index (equivalent to maximize Bhattacharyya distance). Figure 3.5. shows the plot of the negative Bhattacharyya distance versus angle.

After computing 'the vector a_{max} that maximizes Bhattacharyya distance (minimize negative Bhattacharyya distance) we projected the data to a one dimensional space. Figure 3.6 shows the density functions of the projected data.

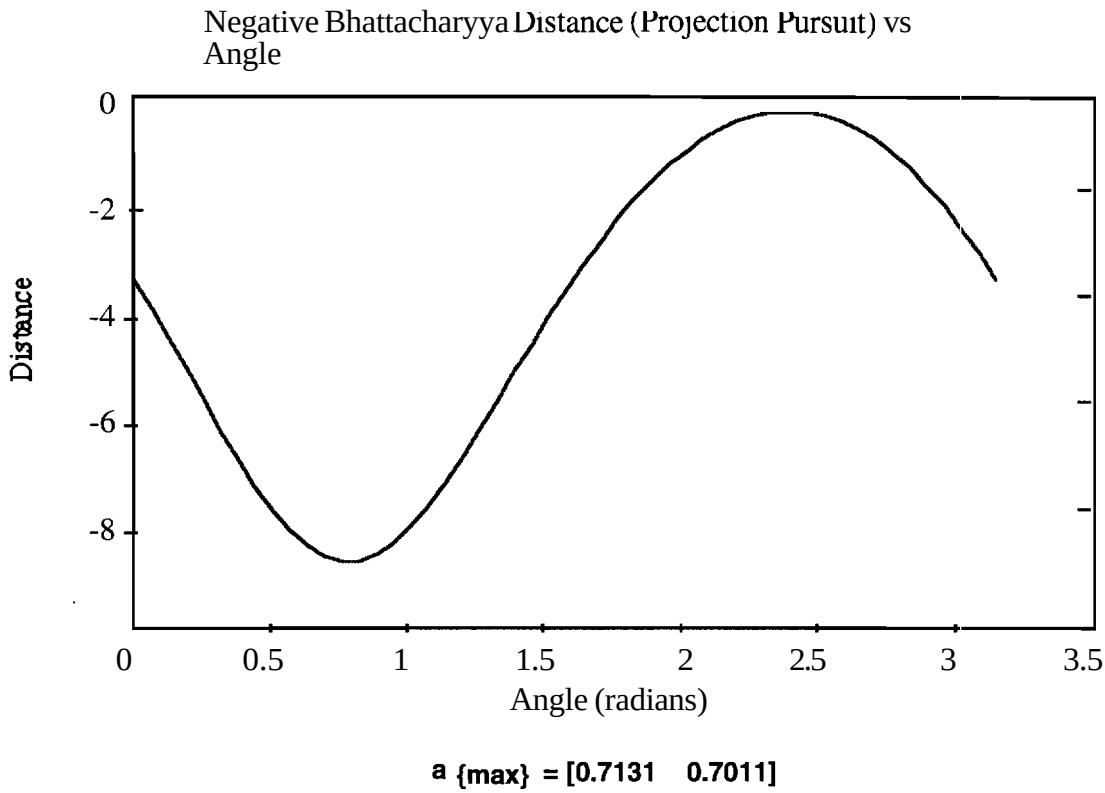


Fig. 3.5. Negative of Bhattacharyya distance versus angle.

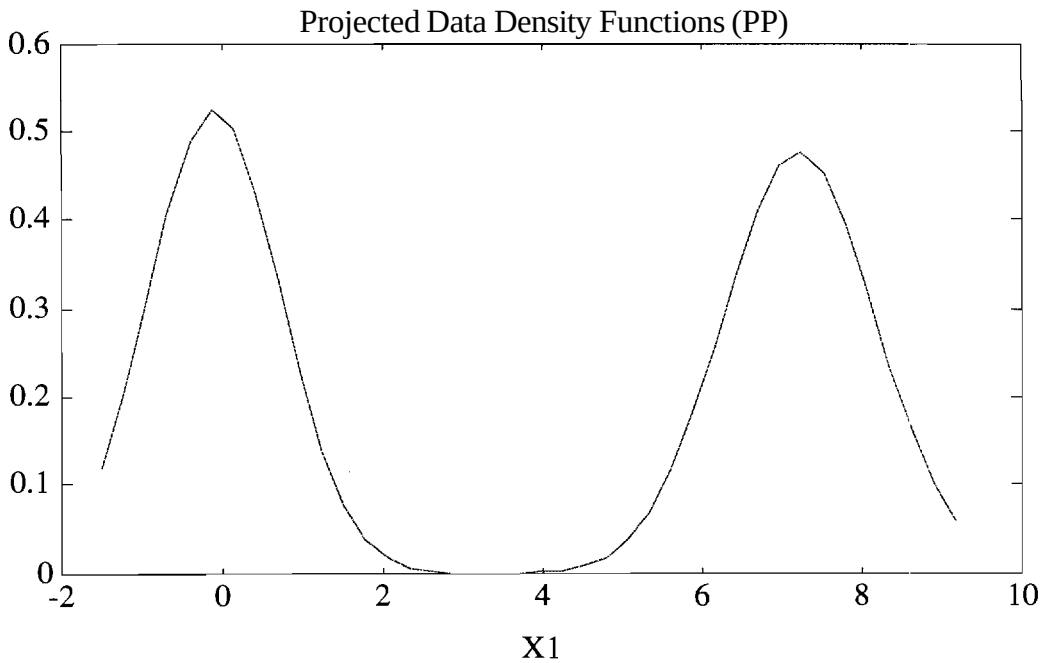


Fig. **3.6.** Densities of the projected data.

In the second set of data we have two normal classes with parameters:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 1.4 \\ 1.4 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 5 & 0 \\ 0 & 1/2 \end{bmatrix} \text{ and, } \ \Sigma_2 = \begin{bmatrix} 1/2 & -2 \\ 1 & 5 \end{bmatrix}$$

As can be seen these two classes are more difficult to separate.

Figure 3.7 shows the data in a 2-dimensional space, Figure 3.8 the negative Bhattacharyya distance, and Figure 3.9 the density functions of the projected data at a{max}.

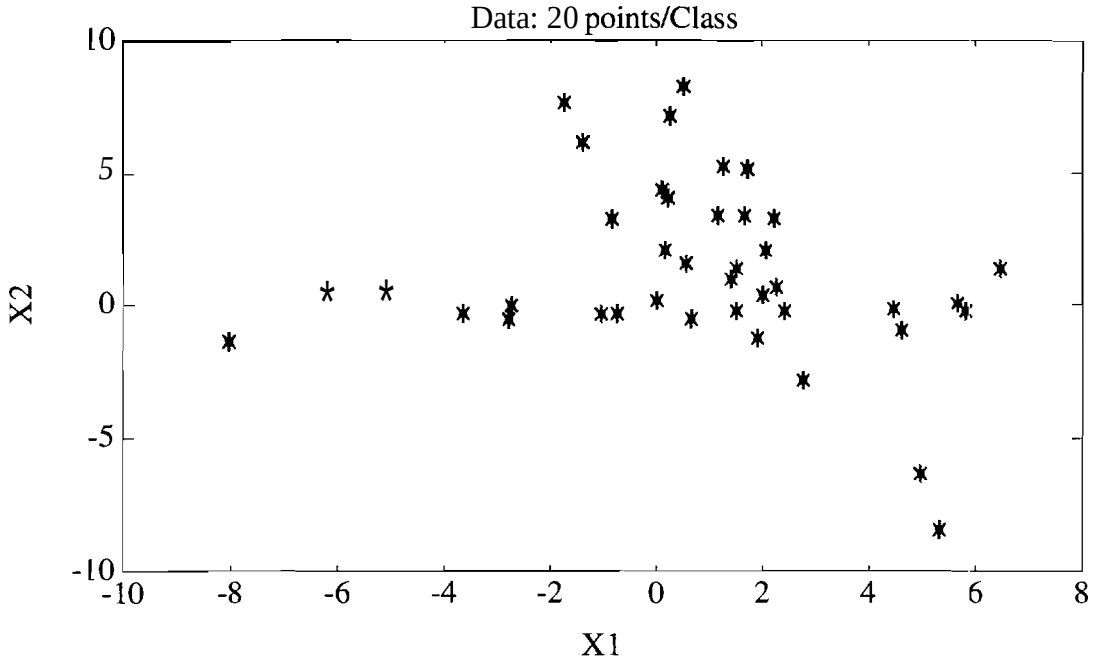


Fig. 3.7. Example two dimensional normally distributed data.

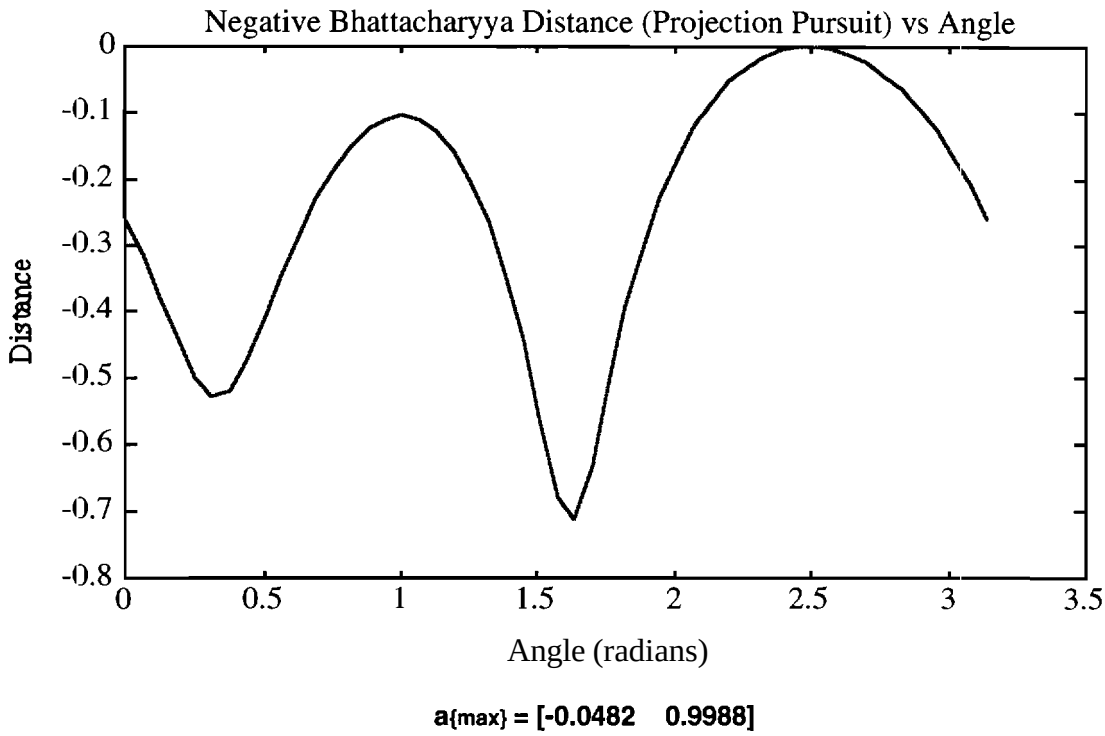


Fig. 3.8. Negative of Bhattacharyya distance versus angle.

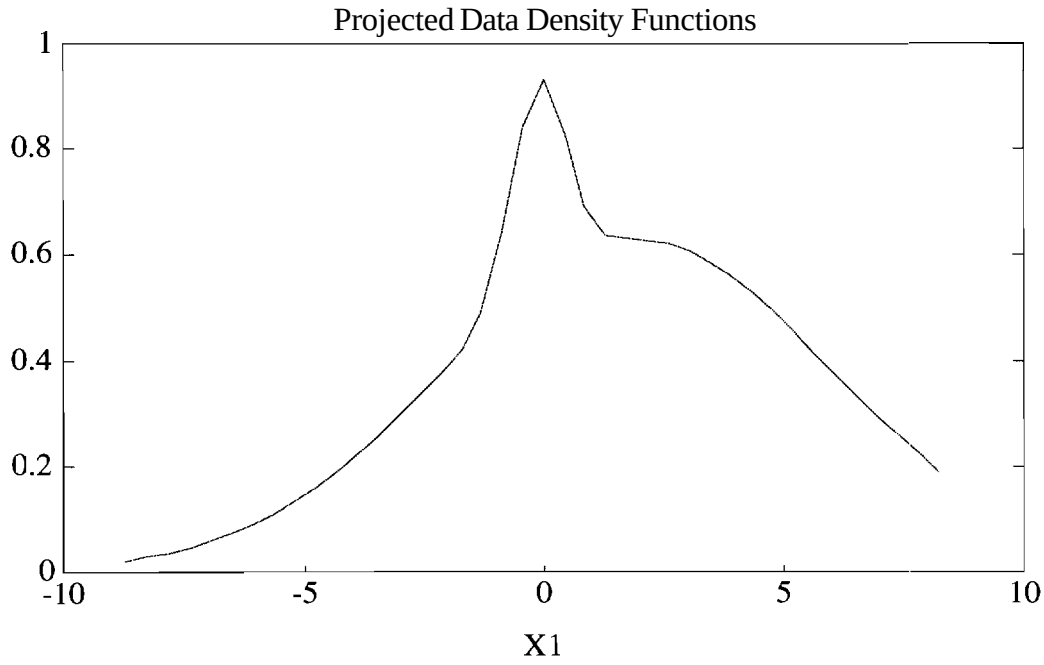


Fig. 3.9. Densities of the projected data.

Figure 3.8 shows an in- porta ant detail. The optimization process can arrive at a local optimum instead of at a global one.

The computation of the parametric matrix A can lead to some problems. It must be guaranteed that the columns of A are linearly independent. Additionally there are obstacles such as the arrival at a local optimum and the computation time. Such difficulties increase when the number of dimensions is large in the original space Φ, as in the case of AVIRIS data with 220 bands. Reducing the dimensionality directly from 220 to, for example, 20 and avoiding such problems in the process of optimization of the projection index could be difficult. In order to overcome to a great extent such obstacles, a set of constraints on the matrix A will be proposed. Henceforth, when Projection Pursuit is mentioned, it will refer to the parametric approach.

3.5 Projecting Adjacent Groups of Features: Parallel and Sequential Projection Pursuit

3.5.1 Proposed constraints on A

In this section the special constraints imposed on the A matrix will be explained. The objective of these limitations is to divide the bands in the space Φ into a partition of groups of adjacent features in order to project each group to one dimension. For a definition of the constraints, A can be rewritten as: $A = [\mathbf{A}_1\ \mathbf{A}_2\ \cdots\ \mathbf{A}_{M-1}\ AM]$, were $\mathbf{A}_i$ is the i^{th} column of A. Every column of A will be filled with zeroes, except at a group of adjacent positions, i.e., $\mathbf{A}_i = [0\ \cdots\ 0\ \mathbf{a}_i\ 0\ \cdots\ 0]^T$ where $\mathbf{a}_i$ is defined as: $a_i = \left[a_{1i}\ \ a_{2i}\ \ \ldots\ \ a_{n_i i}\right]^T$. Observe that the column $\mathbf{A}_i$ will combine $\mathbf{n}_i$ adjacent bands. In order to have a partition of groups of adjacent bands the columns must be orthogonal, and no two $\mathbf{A}_i$'s may have nonzeroes at the same locations. In other terms, for all i, j such that for $i \neq j$ $\mathbf{A}_i^T \cdot \mathbf{A}_j = 0$.

The physical interpretation of the constraints are shown in Figure 3.10 and Figure 3.11. Every group of $\mathbf{n}_i$ adjacent bands will be linearly combined to produce one feature. No two groups will have the same feature. The spectral response of every element of the multispectral data is projected to a lower dimensional subspace preserving the order of the features of the spectral response for the purpose of human analysis. These projections correspond in Figure 2.18 to a mapping from the original space Φ to the subspace Γ.

Some of the advantages that the constraints provide to the optimization process are:

It (1) is fast, (2) preserves the order of the features in the class spectral response, (3) is flexible in terms of the number of adjacent bands to be combined, (4) takes into consideration the ground truth information and the interest of the analyst, (5) the A columns are orthogonal, allowing the algorithm to avoid linear dependencies among $\mathbf{A}_i$'s, (6) will make easier the process to construct an initial guess matrix $\hat{A}$

Still there is an issue to be solved: how is the optimization of the projection index to be implemented in such a scheme of linear combination of features? There are two approaches: (1) in every group of adjacent features the projeclion function is optimizecl locally and independently of each other, producing one feature, (2) The linear combinations of adjacent bands are calculated in a way that optimizes the global projection index in the projected subspace where the data set **Y** is localized. These approaches will be called Parallel Parametric Projection Pursuit and Sequential Paranietric Projection Pursuit.

3.5.2 Parallel Parametric Projection Pursuit

In this approach each group of adjacent bands is linearly projected to obtain one feature. In each projection a vector $\mathbf{a}_i$ is calculated for the i^{th} group of adjacent bands in order to optimize the projection index in the projected vector. That projection creates a new feature in the projected subspace Γ. The projections in every group are independent of each other. Figure 3.10 shows a physical interpretation of the scheme of projection in the spectral response of an element. There must be the same number of optimizations as the number of groups of adjacent bands.

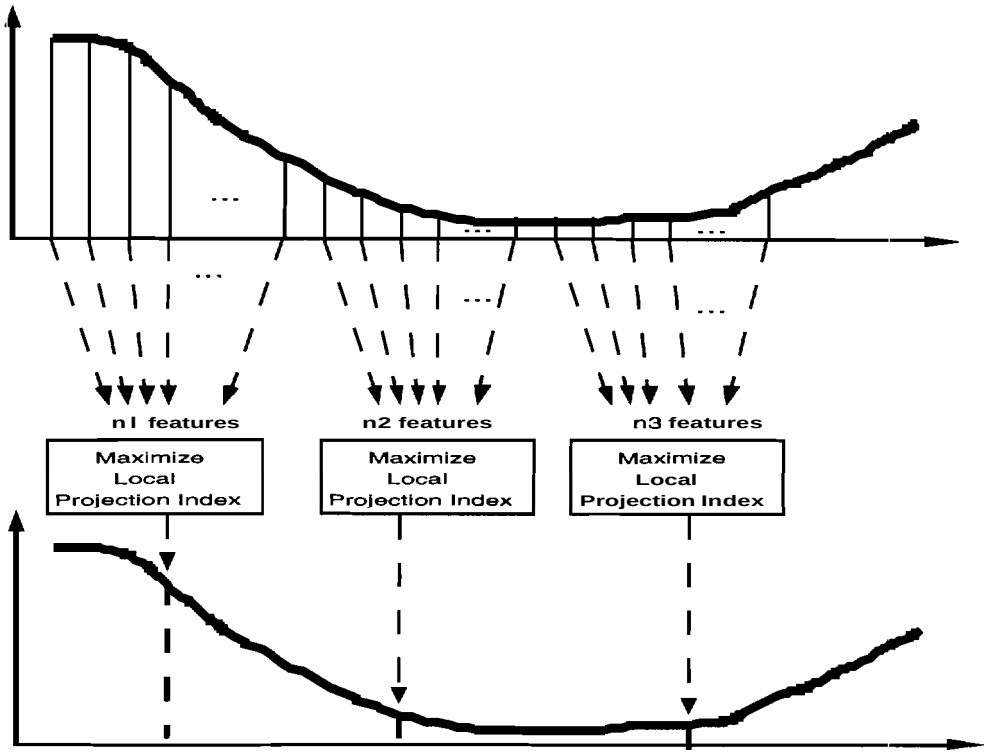


Fig. 3.10. Parallel Parametric Projection Pursuit.

The advantage of such approach is that it is fast, because every group of adjacent bands is projected in parallel and independently of one another. At the same time, this is a disadvantage because there is a lack of relation between such groups of adjacent bands. As a consequence there is a lack of control in the optimization of the projection index in the whole subspace Γ.

3.5.3 Sequential Parametric Projection Pursuit

The problem of lack of relation between groups of adjacent bands is solved by a new algorithm that will project the groups of neighboring bands optimizing the global projectiori index in the projected subspace Γ. For a physical interpretation of this algorithm see Figure 3.11, where the projection of a spectral response of an element is presented. This algorithm can be time consuming. A way of overcoming this problem is to develop an iterative procedure for this approach. Such an iterative approach will follow these steps:

(1) An initial guess for every $\mathbf{a}_i$ for every group of adjacent bands is stored.

(2) Maintaining the rest of the $\mathbf{a}_i$'s constant, compute $\mathbf{a}_1$ (the vector that projects the first group of adjacent bands) to maximize the global minimum Bhattacharyya distance.

(3) Keep repeating the procedure for the i^{th} group where $\mathbf{a}_i$ is calculated optimizing again the global Bhattacharyya distance while maintaining the $\mathbf{a}_j$'s constant, where $i \neq j$.

(4) Once the last i^{th} group of adjacent band is projected kept repeating the process from step 2 (compute all the $\mathbf{a}_j$'s sequentially) until the maximization stops increasing significantly.

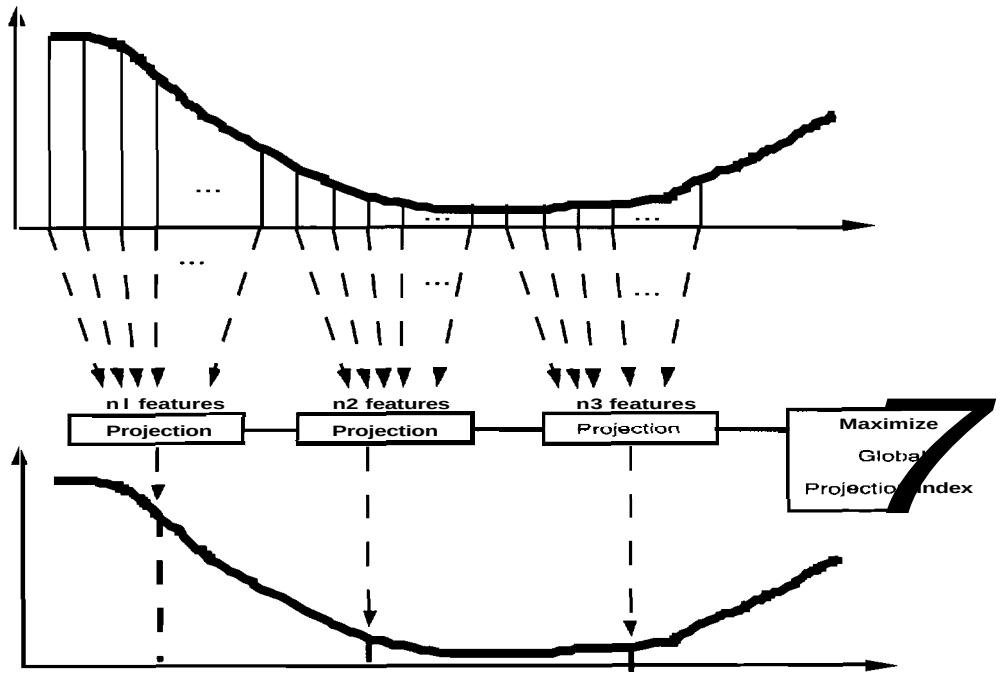


Fig. 3.11. Sequential Parametric Projection Pursuit.

3.5.4 Optimization

Projection Pursuit based procedures require a numerical optimization of the multidimensional function $I(\mathbf{A})$, also written as $I(\mathbf{A}^T\mathbf{X})$. Different classes of methods have been developed to optimize multidimensional functions. Among them are Downhill Simplex, Direction Set, Conjugate Gradient, Variable Metric and Simulated Annealing. The analyst can use the method that is thought more appropriated to the type of data and projection index used. In the present research the Downhill Simplex method has been used. This method requires almost no special assumptions about the projection index to be optimized. It could be extremely slow and at the same time robust. This method has been suggested for the case when the optimization is only an incidental part of the overall problem [44]. We believed that is the case because of the many oplimizations that need to be done in the Parallel approach and the iterative version of Sequential Projection Pursuit.

3.6 Experiments

3.6.1 Comparing methods

A series of three experiments were developed with the objective of comparing preprocessing methods, i.e. Parallel and Sequential Projection Pursuit approaches, with the direct use of a Feature Extraction method. The experiments also will enable us to observe how sensitive Projection Pursuit methods are to initial guess of matrix A and different projection indices.

The multispectral data used in these experiments is a segment of AVIRIS data taken of NW Indiana's Indian Pine test site. From the original 220 spectral channels 200 were used, discarding the water absorption bands. This data was obtained in June 1992. By that time most of the crops in the agricultural portion of the test site had not reached their maximum ground cover. In such circumstances the classification is a challenging problem, because the energy measured in the data came not only from the crops; but also from variations in the soil type, soil moisture, and previous crop residues. In the present experiment four classes were defined: corn, corn-notill, soybean-min, soybean-notill. The total number of training samples is 179 (less than the number of bands used). Thus, the algorithms were tested against the problem of a severe lirnitation of samples. Table 1 shows the number of training samples and test samples for each class.

Table 3.1
Classes, number of training and test samples.

Classes	Training Samples	Test Samples
Corn	22	234
Corn-notill	52	620
Soybean-min	61	1910
Soybean-notill	44	737
Total	**179**	**3501**

The multispectral data was reduced in dimensionality to 20 dimensions by three methods: direct use of Discriminant Analysis as a feature extraction method to project from 100 to 20 dimensions. Parallel Projection Pursuit and Sequential Projection Pursuit as preprocessing methods to project from a 200 to a 20 dimensional space.

Using Discriminant Analysis, the data was reduced from 100 bands (one in every two bands from the original 200) to 20 (from Φ space to Ψ subspace). From the original number of bands 100 were used because of the limited nurnber of training samples (179). Parallel Projection Pursuit and Sequential Projection Pursuit (iterative approach) were applied to the data to reduce the dimensionality from 200 to 20 dimensional subspace (from Φ to Γ) optimizing a projection index. In both approaches the number of adjacent bands combined in each group was held constant:: 10 bands linearly combined to produce a new feature. After the dimensionality of the data was reduced to 20 by both approaches, Discriminant Analysis, Decision Boundary and Feature Selection where used as feature extraction algorithms in order to project from Γ to the Ψ subspace. The feature selection method used was minimum Bhattacharyya distance as a measure of statistical distance among the classes.

Four types of classifiers were used: ML, ML with 2% threshold, a spectral-spatial classifier named ECHO [45] [46] and ECHO with 2% threshold. In the second and the fourth, a threshold was applied to the standard classifiers such that if the classes were truly normal 2% of the least likely points would be thresholded. These 2% provide one indication of how well the tales of the data fit the normal model. All of these classifiers performed a projection from Ψ to the resulted space Ω.

In the first experiment the projection index used was the minimum Bhattacharyya distance among the classes. The initial guess for matrix $\hat{\mathbf{A}}$ is one that averages every group of adjacent bands, i.e. $\hat{\mathbf{a}}_i = [1 \quad 1 \quad \cdots \quad 1]^T$. This experiment will test Parallel and Sequential Projection Pursuit against direct use of Feature Extraction methods, i.e. Discriminant Analysis, to project data from Φ space to Ψ subspace. In the second experiment the same projection index is used, while a different initial guess for matrix $\hat{\mathbf{A}}$ was used. This experiment will test how well Parallel and Sequential Projection Pursuit deal with the problem of global optimization and how sensitive they are to a variation in $\hat{\mathbf{A}}$. The third experiment uses a different projection index, the Fisher criterion, and will test it against the use of minimum Bhattacharyya distance. All the tests are in terms of test field classification accuracy.

3.6.2 Experiment 1

The minimum Bhattacharyya distance among the classes was calculated in 20 dimensional space for the three data sets corresponding to the three rnethods used to project the data to a subspace of Φ. The result is shown in Figure 3.12.

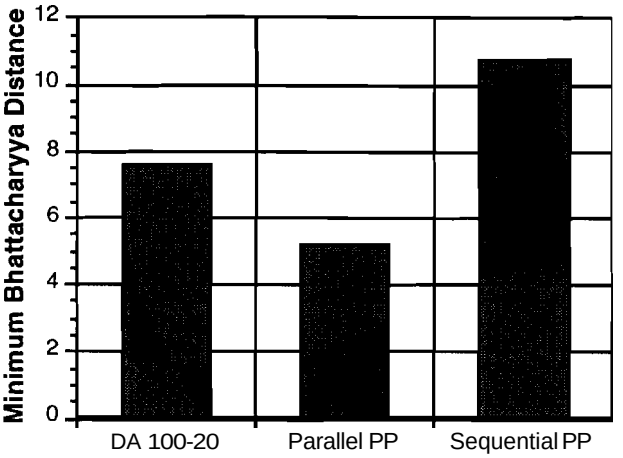


Fig. 3.12. Minimum Bhattacharyya Distance among the classes.

As can be observed Sequential Projection Pursuit preserved more information in terms of minimum Bhattacharyya distance than Discriminant Analysis From 100 bands (DA 100-20) and Parallel Projection Pursuit. The result is based on the fact that Discriminant Analysis makes the computation at high dimensionality (100 dimensions of the original Φ space) with a small number of label samples (179 samples) where the Hughes Phenomena takes place. Another element to take into consideration is that Discriminant Analysis calculates the features maximizing another index than Bhattacharyya distance, named the Fisher criterion.

Sequential Projection Pursuit makes the computation and directly maximizes the projection index to a 20 dimensional space Γ. Parallel Projection Pursuit maximizes the minimum Bhattacharyya index at each one of the 20 features independently of each other. As a consequence, there is a lack of control over the distance among the classes in the total projected subspace. The subsequent subsections will show the results of projecting the data from the Γ subspace to Ψ with different feature extraction or selection methods in order to compare them with direct projection from Φ space to Ψ using Discriminant Analysis (DA 100-20).

Discriminant Analysis

This feature extraction method was used to project data from the Γ subspace to Ψ after the Projection Pursuit based methods were applied. It will provide the most direct cornparison against direct projection from Φ to Ψ (DA 100-20) because the same feature extraction procedure was used either at the Φ space and at the Γ subspace.

After Discriminant Analysis was applied to both data sets where Parametric Projection Pursuit (Parallel and Sequential approaches) was used they were classified and the test fields classification accuracy results can be seen in Figures 3.13, 3.14, 3.15 and 3.16. The classification accuracy results on the test fields for standard Maximum Likelihood classifier can be seen in Figure 3.13.

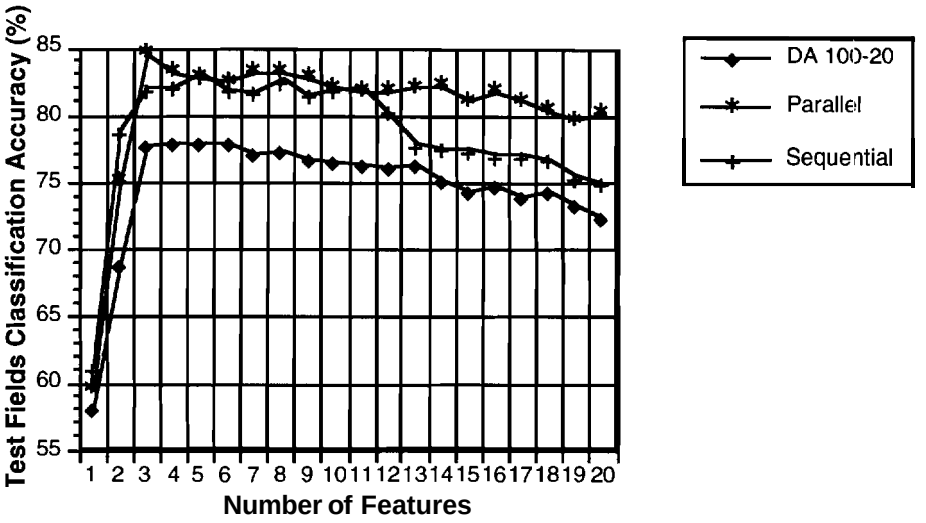


Fig. 3.13. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after difference methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

As can be seen, the classification accuracy in the data from the two Projection Pursuit based approaches are much better than using direct Discriminant Analysis (DA 100-20). The reason is that both approaches made the computation at a small dimensional space. This allows the approaches to deal better with the Hughes Phenomena and high dimensional space characteristics, preserving more information. This enables Discriminant Analysis to make the computation at fewer dimensions with the same number of labeled samples, computing more accurate features. Because we have a small number of classes (4) the optimum number of features using Discriminant Analysis is 3. It is possible that such a small number of classes enables the Parallel approach to reach the maximum, in terms of classification accuracy, because this procedure optimizes each group of adjacent bands locally. Also the global minimum Bhattacharyya distance for Parallel Projection Pursuit was large enough, more than 5, to maintain the classes well separated for classification purposes. On the basis of the fact that the optimization in each feature is independent

of each other, the results can not be guaranteed for most experiments, especially for the cases where the number of classes is large.

The same steps were followed again but this time using Maximum Likelihood with a 2% threshold. The results are shown in Figure 3.14.

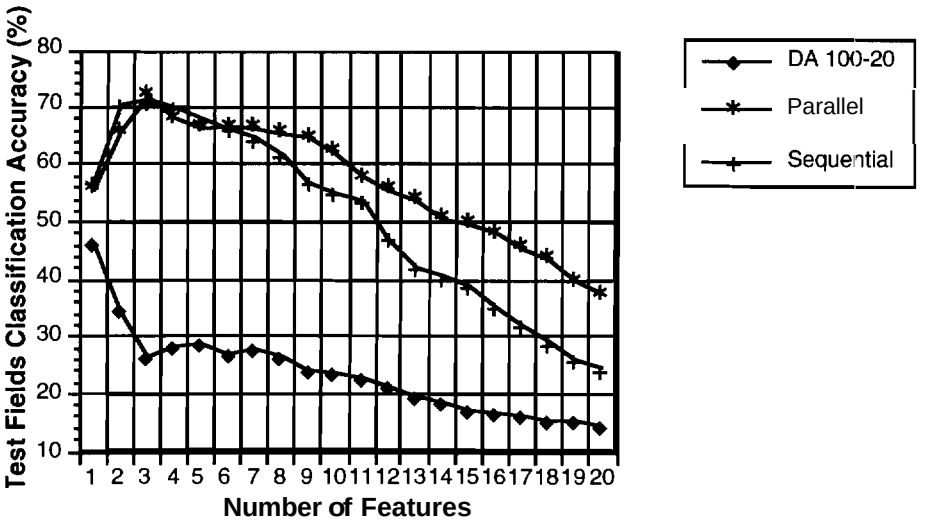


Fig. 3.14. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

Note that both approaches of Projection Pursuit performed significantly better compared with Discriminant Analysis used directly from 100 dimensions, with a difference as much as 50%. It is significant that such a difference happens at the use of the best three features. It is known that Discriminant Analysis computes a number of features equal to the number of classes minus one, in this case three. In the Discriminant Analysis algorithm the rest of the features are selected randomly. The optimum classification accuracy was expected to be at three for ML{threshold} in all cases. Such a maximum point was reached only with the use of Projection Pursuit based algorithms. In the direct Discriminant Analysis (DA 100-20) that is not the case, because it is thresholding most of the data. Because of the Hughes Phenomena and other high dimensional characteristics, Discriminant Analysis is not computing accurate features as a result of making the computation at 100 dimensions with a small number of samples. This is shown in the fact that classification accuracy immediately starts to decrease. Projection Pursuit based algorithms, on the other hand, increase as expected until they reach a maximum at three best features. The

reason is that the assumption of normality holds better when the computations are done at the lower dimensional space, Γ.

Figure 3.15 and 3.16 show the results for the ECHO classifier and ECHO with 2% thresholds. The results are similar to those with the ML classifiers and support our previous discussion. The only difference is that for the ECHO classifier, Parallel Projection Pursuit performs even better than the Sequential approach.

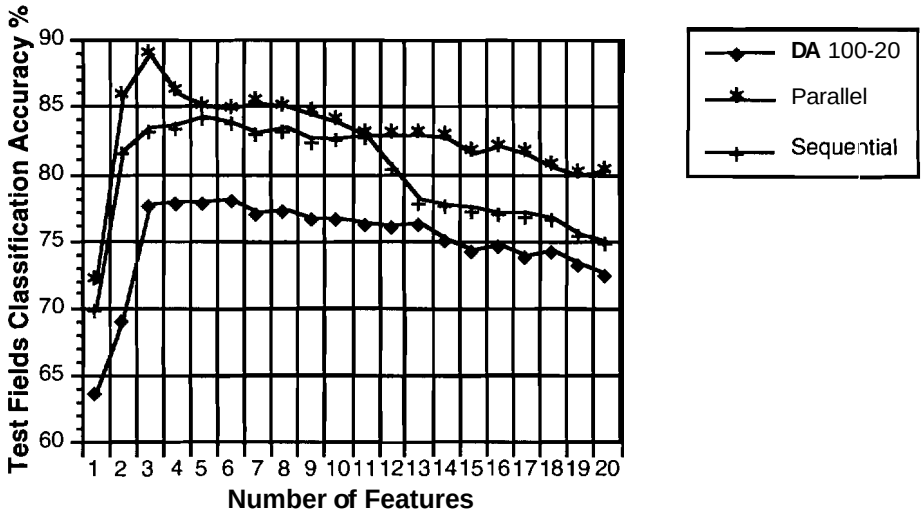


Fig. 3.15. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

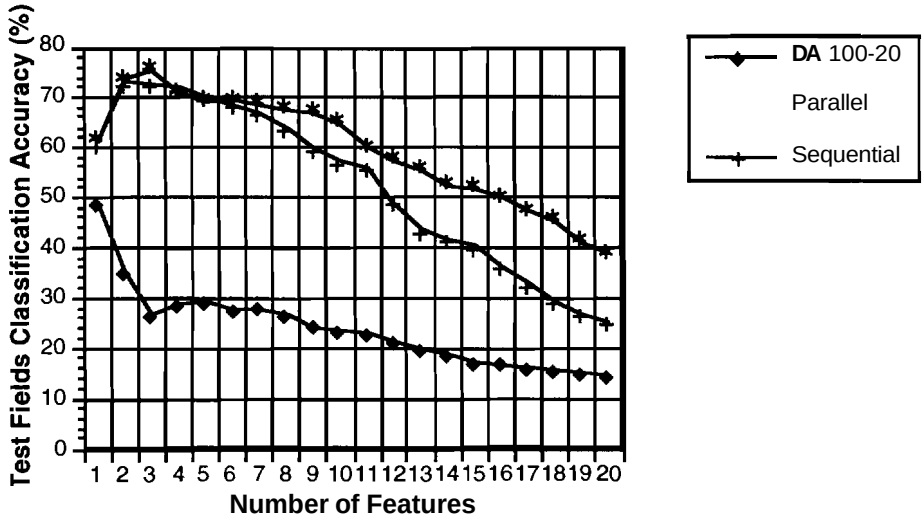


Fig. 3.16. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

Decision Boundary

This feature extraction algorithm was used to project data from Γ to Ψ after the use of Projection Pursuit based algorithms and compare its results with direct use of Discriminant Analysis in high dimensional space. The Decision Boundary method could not be used at 200 bands to project the data from Φ to Ψ, because it required at least 201 samples per class. The difference between DA 100-20 and Decision Boundary at 20 dimensions is low. The results in the ML and ECHO classifier cases can be explained by the fact that Decision Boundary demands more samples than Discriminant Analysis. Still the classification with thresholds shows that Projection Pursuit based preprocessing approach have a better grounded assumption of normality.

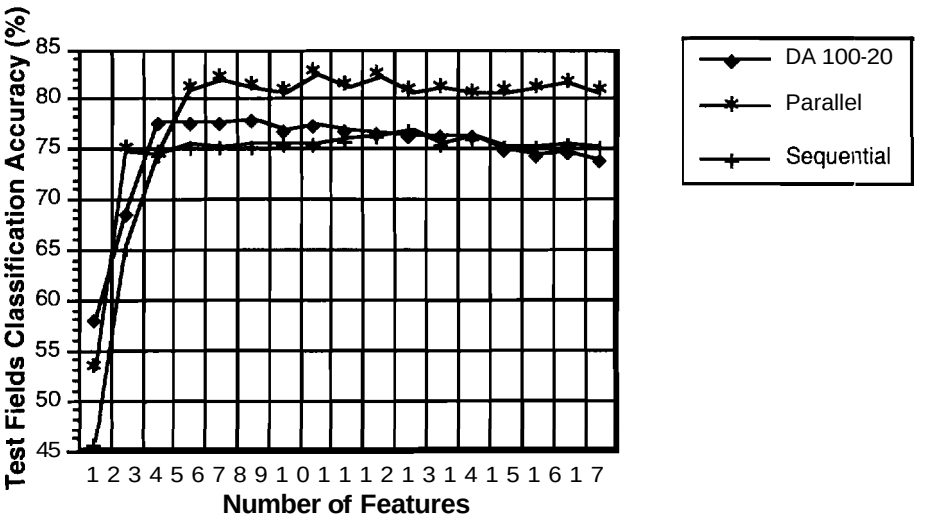


Fig. 3.17. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

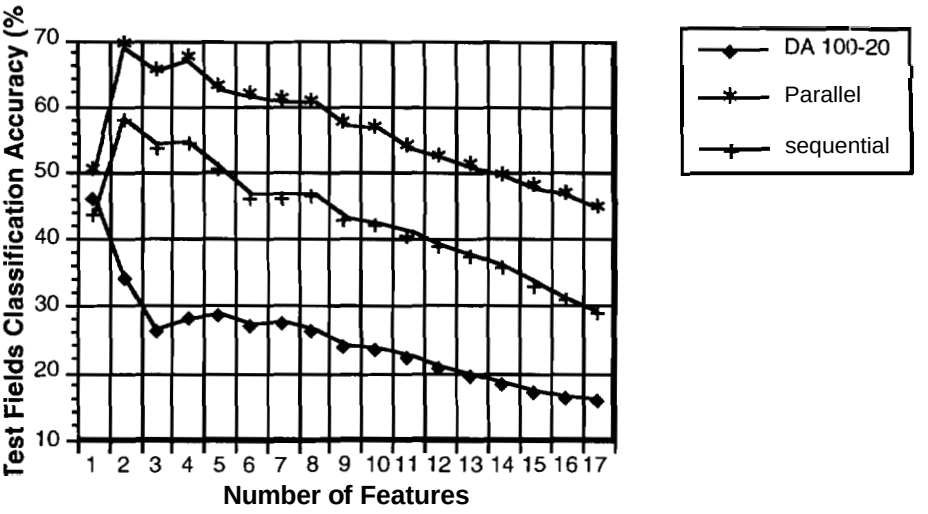


Fig. 3.18. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

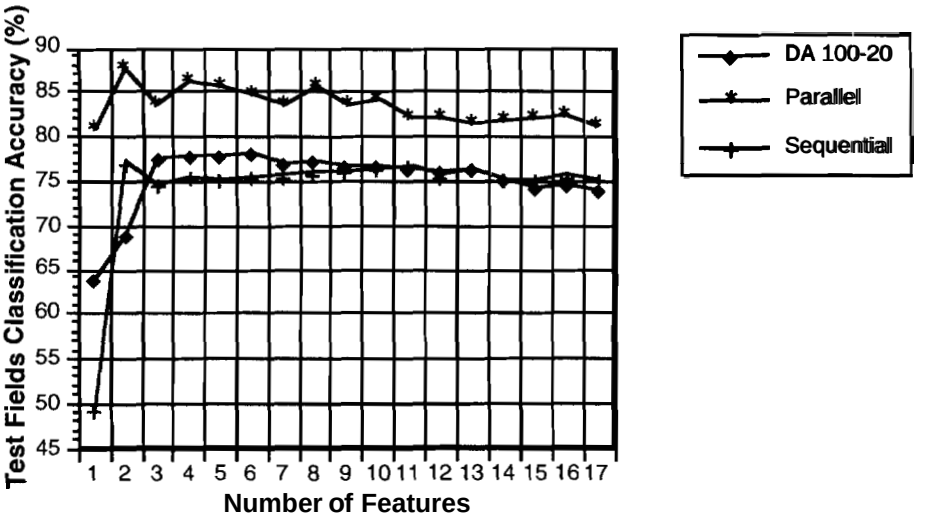


Fig. 3.19. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

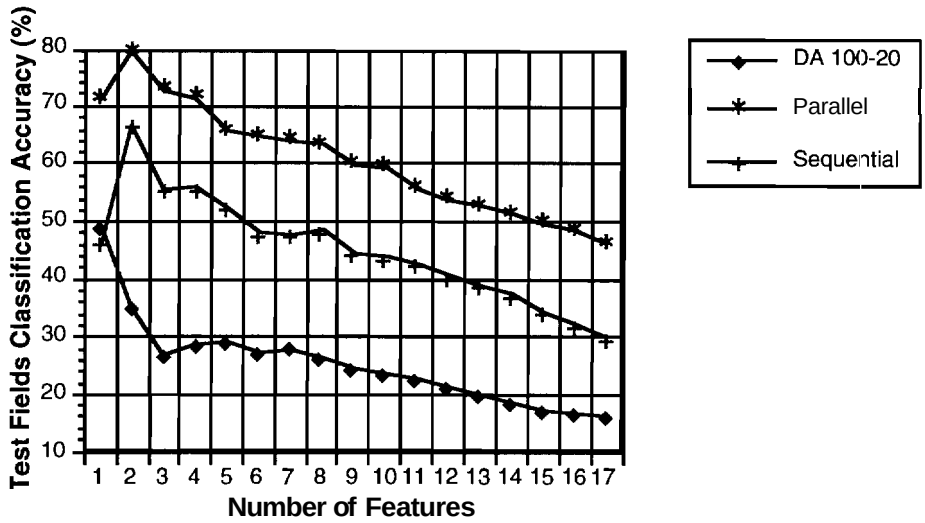


Fig. 3.20. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

Feature Selection

Feature selection could not be used in the 200 dimensional space Φ to project the data lo the Ψ subspace. That is because the number of calculations for feature selection in high dimensional space will be extremely high $200!/((20!)(180!)) \approx 1027$. Feature selection was applied, as previously done with Discriminant Analysis and Decision Boundary, after the use of Projection Pursuit based algorithm:;. The results in terms of classification accuracy, were compared with direct application of Discriminant Analysis (DA 100-20). In all the experiments the classification accuracy and the assumption of normality were better with feature selection than with direct use of Discriminant Analysis. Note that in the first to fourth features Sequential Projection Pursuit performs better than in the rest. The reason is that feature selection is more related to Sequential Projection Pursuit. That occurs because the Sequential approach directly maximizes the same global statistical distance used in feature selection.

The results for ML and ECHO classifiers confirm what had been said previously, that Projection Pursuit based algorithms handle Hughes phenomena, normality assumptions and geometrical and statistical properties of high dimensional space better than direct use of Discriminant Analysis (DA 100-20) in high dimensional data.

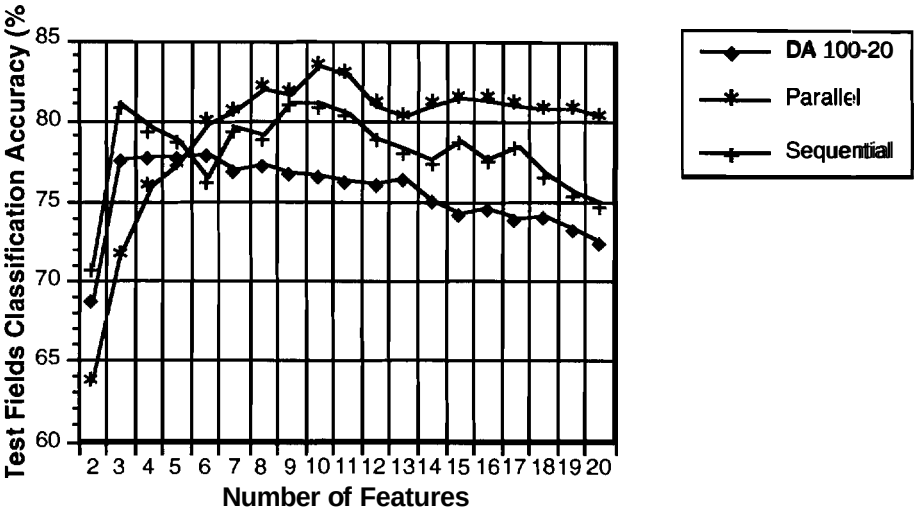


Fig. 3.21. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

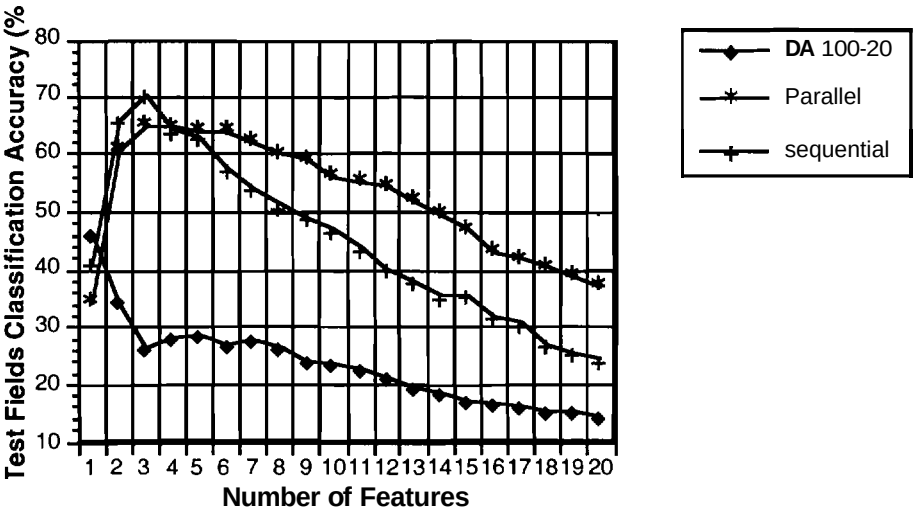


Fig. 3.22. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

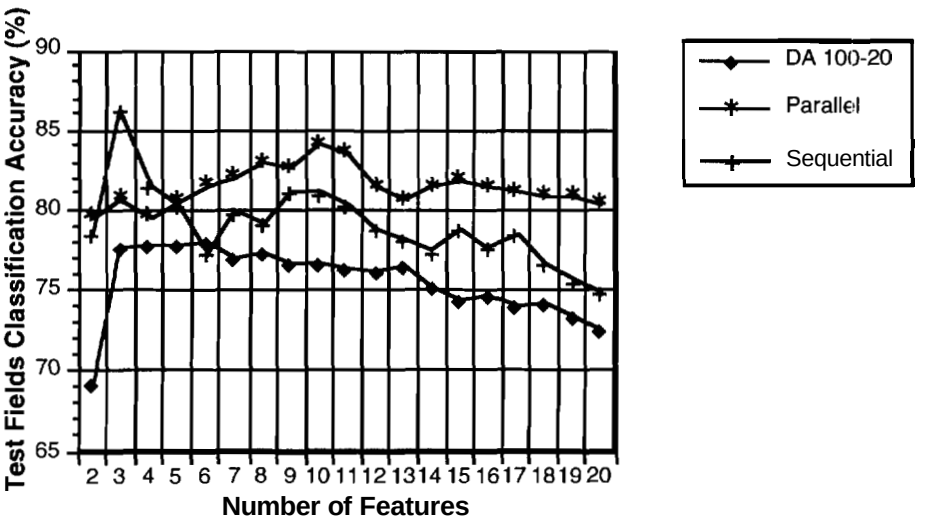


Fig. 3.23. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

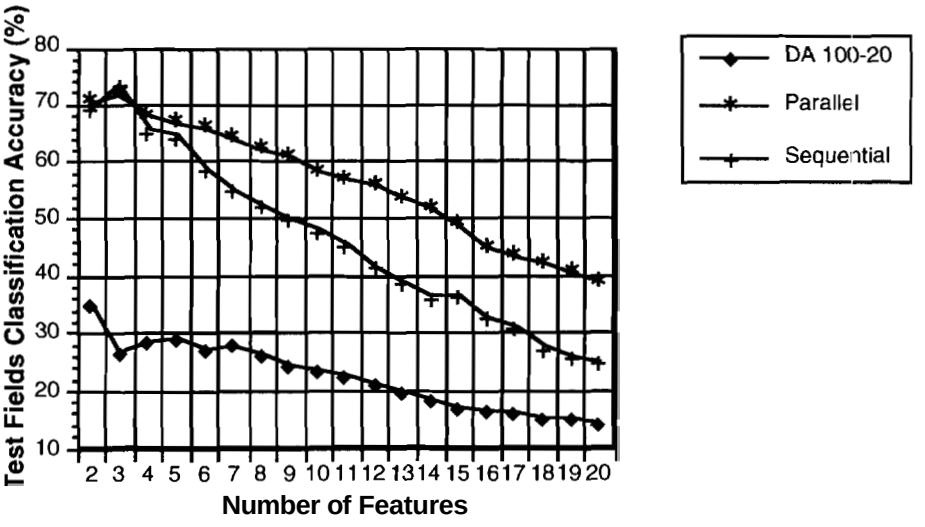


Fig. 3.24. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

3.6.3 Experiment 2

In this experiment the same projection index, i.e. minimum Bhattacharyya distance and a different initial guesses for matrix A were used in order to test how sensitive the Projection Pursuit procedures were to this parameter.

After the data was projected to the 20 dimensional subspaces, by the different methods, the minimum Bhattacharyya distance among the classes was calculated. The results can be seen in Figure 3.25. The figure shows how Sequential Projection

Pursuit's amount of statistical distance increases with respect to experiment 1. At the same time Parallel Projection Pursuit's index decreases significantly with respect to the same experiment.

As mentioned before this shows how the lack of overall control in the optimization process affects the performance of Parallel Projection Pursuit. The subsequent subsections will show the result of projecting the data from Γ subspace to Ψ by different feature extraction or selection methods in order to compare them with direct projections from Φ space to Ψ subspace using Discriminant Analysis (DA 100-120).

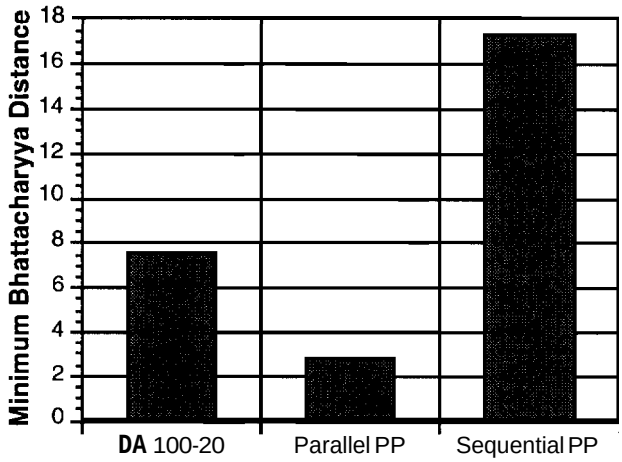


Fig. 3.25. **Minimum Bhattacharyya distance among the classes.**

Discriminant Analysis

Here Discriminant Analysis was used as a feature extraction method to project the data from Γ space to Ψ subspace to compare its results with direct use of Discriminant Analysis from Φ to Ψ and with the previous experiments.

Figures 3.26 and 3.27 show the ML classification results. Note how the Parallel approach performs more poorly than even DA 100-20. That is because of the small separation among the classes in the Γ subspace. This experiment shows that Parallel Projection Pursuit depends more on the initial guess matrix $\hat{\mathbf{A}}$ variation than Sequential Projection Pursuit. The spatial-spectral ECHO classifier has similar results shown in Figures 3.28 and 3.29. Sequential Parametric Projection Pursuit with its direct control over the overall optimization shows a better performance in terms of maintaining classes separation in the process of reducing the dimensionality and is more robust than the Parallel approach to the initial guess of matrix A. Because in the Parallel approach the optimization is done in each feature independent of each other, it is not guaranteed what could happens in the global projection index.

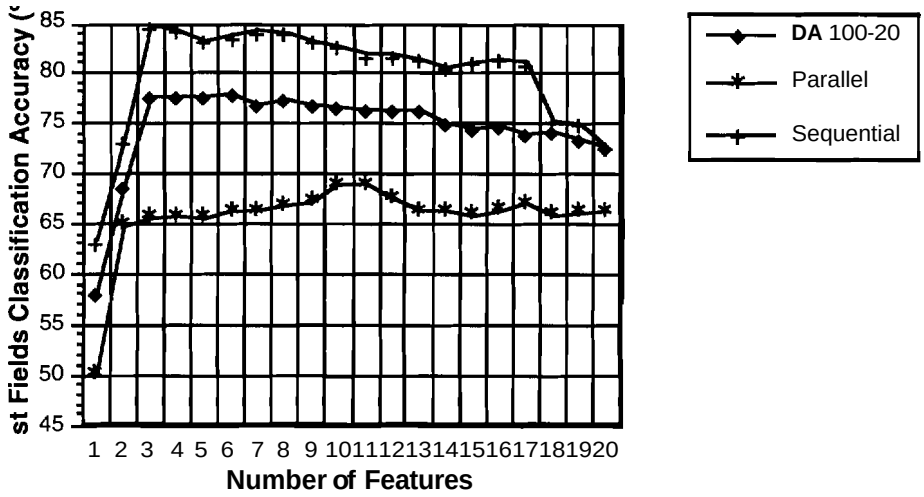


Fig. 3.26. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

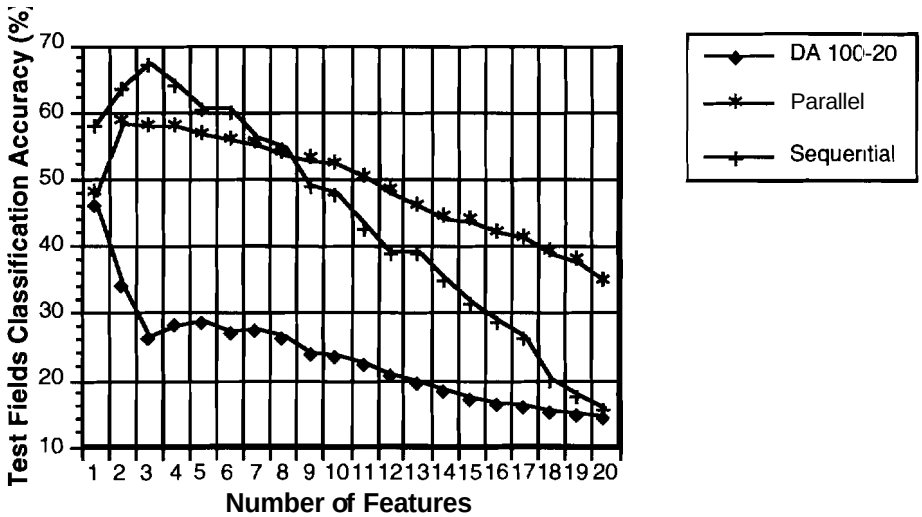


Fig. 3.27. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

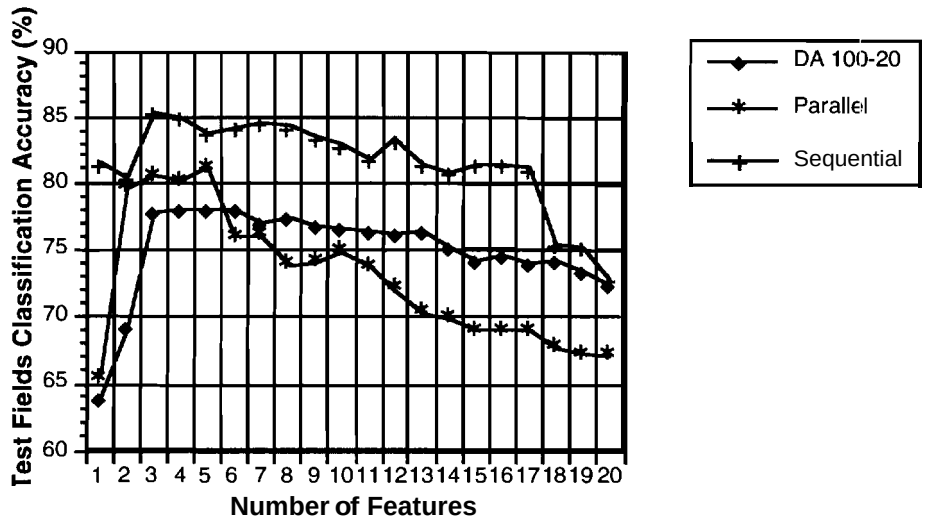


Fig. 3.28. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

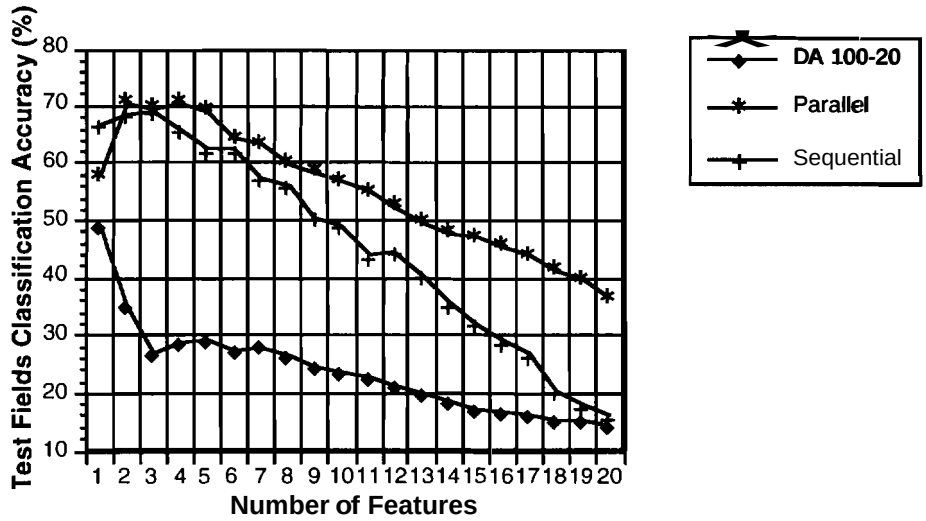


Fig. 3.29. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

Decision Boundary

The results of the use of Decision Boundary as a feature extraction method show as in experiment 1 that this depends on a large number of labeled samples. This method is probably more sensitive to that number than to the separation of classes at high dimensional space in order to estimate accurate features.

The results show that Discriminant Analysis is less sensitive to the number of labeled data than Decision Boundary Feature Extraction in terms of classification

accuracy, In some circumstances Decision Boundary can estimate such inappropriate features so as to even diminish the assumption of normality, as shown in Figure 3.30 and Figure 3.31.

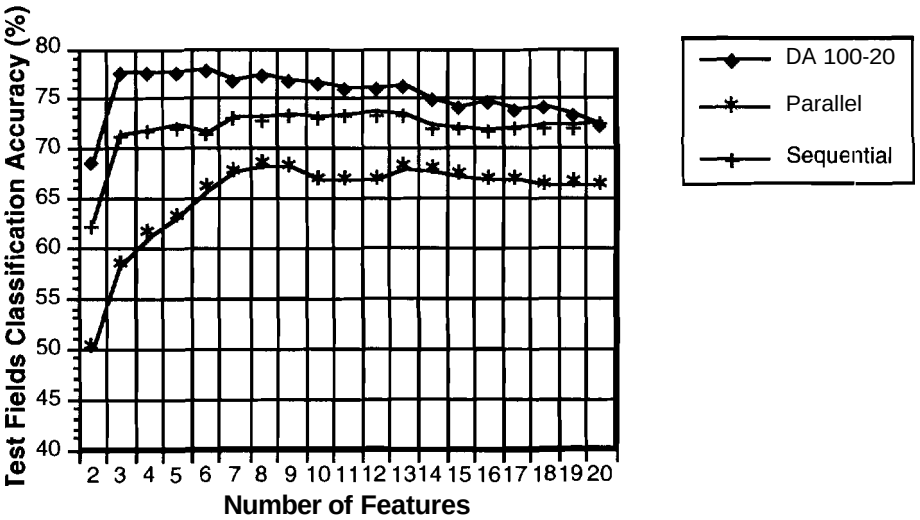


Fig. 3.30. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

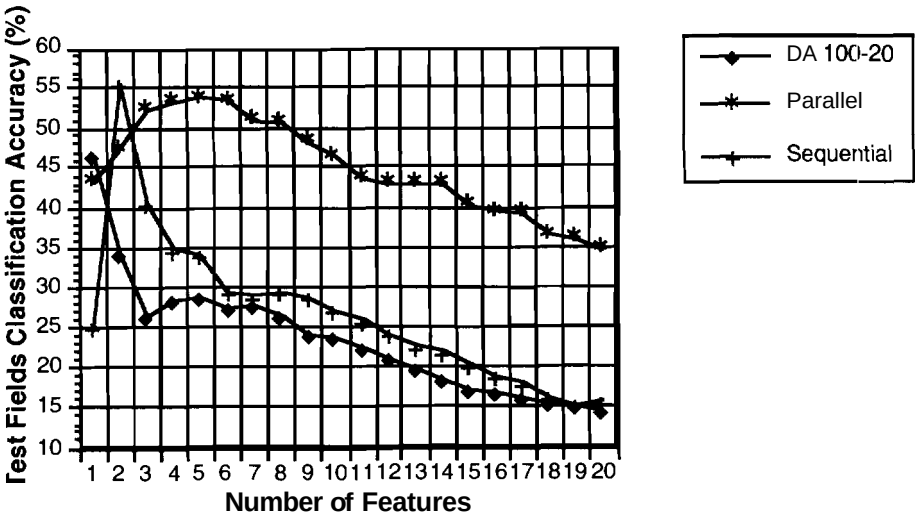


Fig. 3.31. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

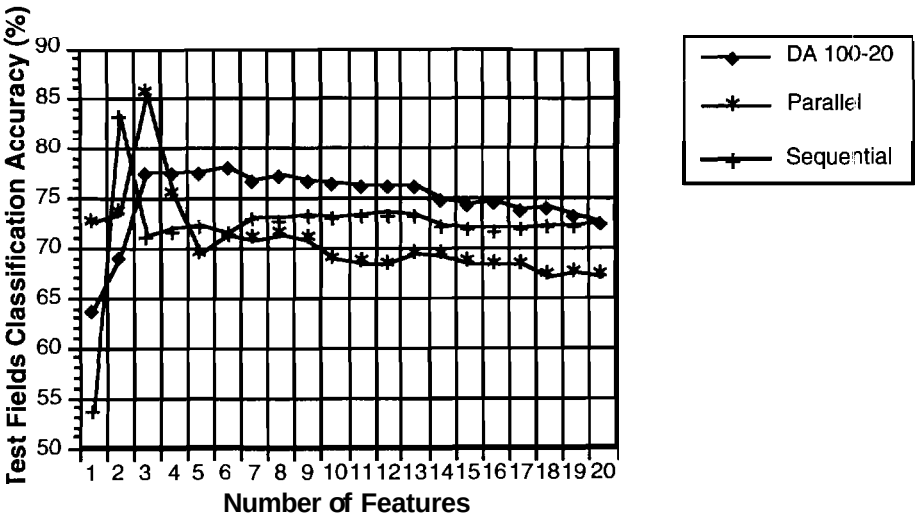


Fig. 3.32. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

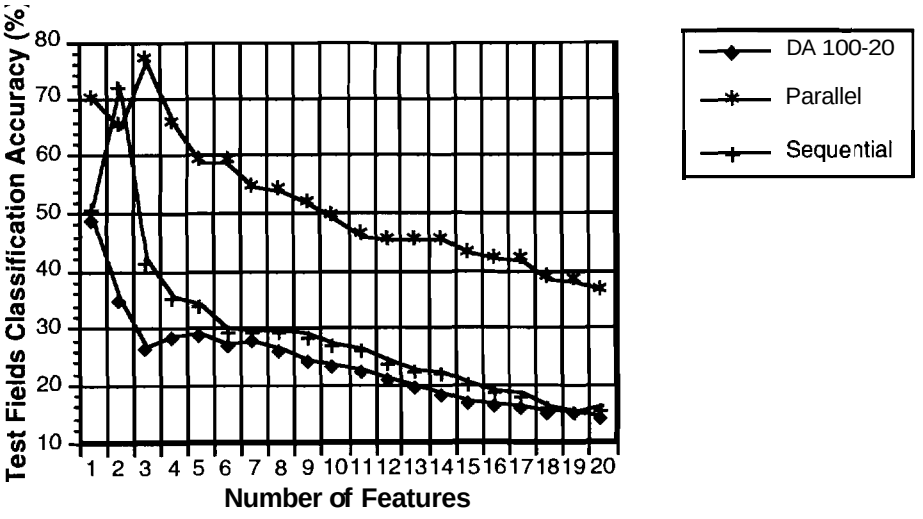


Fig. 3.33. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

Feature Selection

The results of this subsection show that Sequential Projection Pursuit, which has the largest measure of minimum Bhattacharyya distance performs better than direct Discriminant Analysis and Parallel Projection Pursuit. The Parallel approach had the poorest performance due to the small measure of projection index. Feature selection, as statecl before, seems to be directly related to the global minimum Bhattacharyya distance. The results shown in Figures 3.34, 3.35, 3.36, and 3.37 are not surprising

since the feature selection algorithm applied uses the minimum Bhattacharyya distance as it measure of class separability.

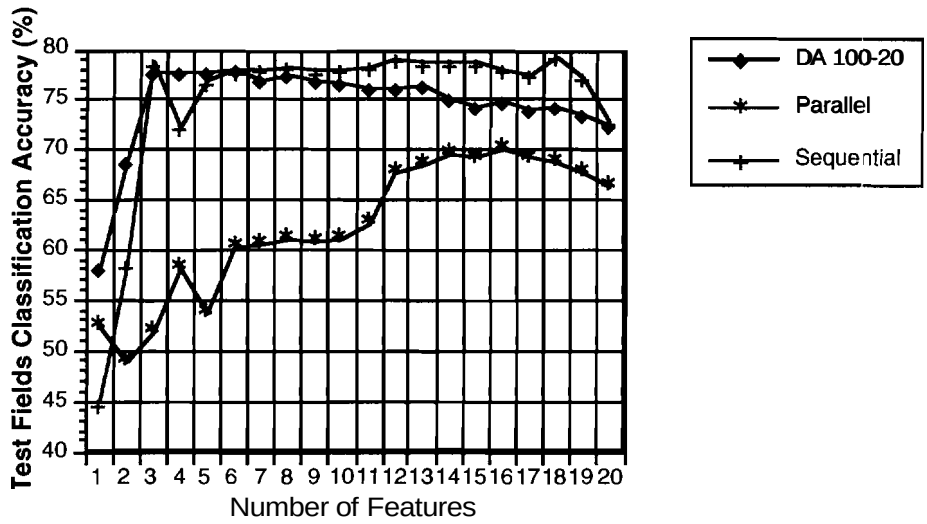


Fig. **3.34.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ML Classifier.

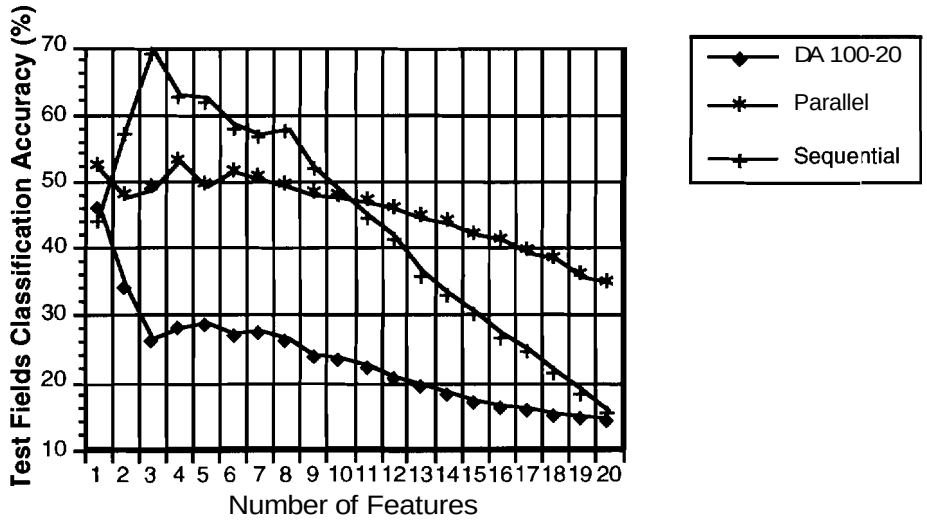


Fig. **3.35.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ML with threshold.

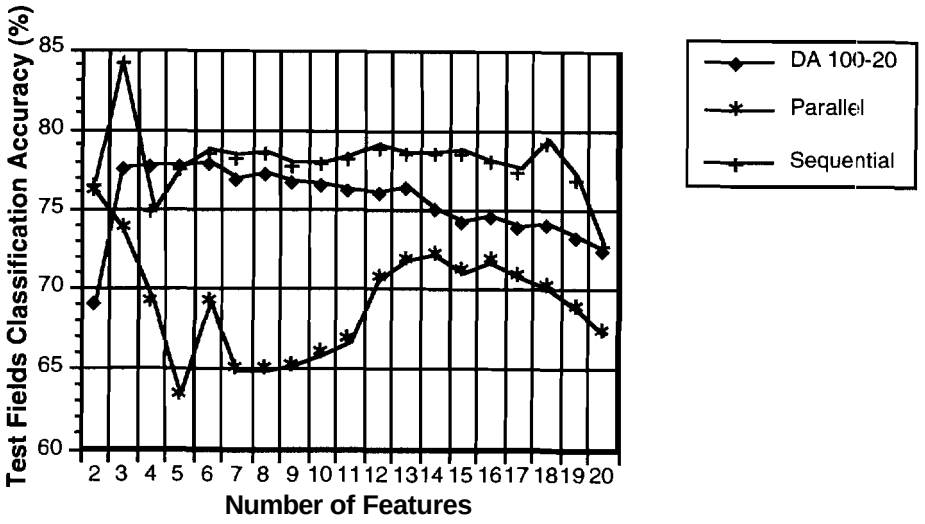


Fig. **3.36.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO Classifier.

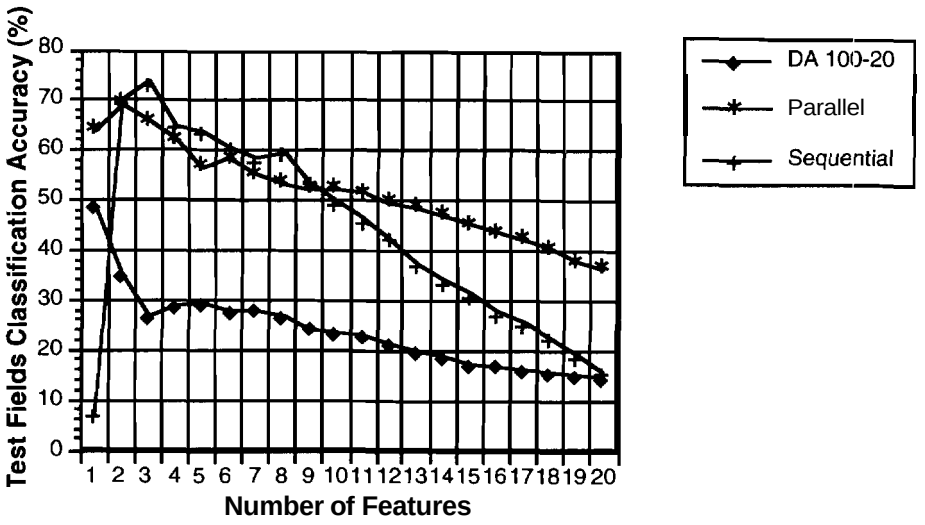


Fig. **3.37.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (Parallel and Sequential) for ECHO with threshold.

3.6.4 Experiment 3: Fisher ratio criterion as a projection index

The purpose of this experiment is to test another possible projection index and compare it with minimum Bhattacharyya distance. The proposed projection index is the Fisher criterion as defined previously. A minor modification has been done to provide a, matrix Projection Pursuit form. Accordingly the index is

$$I(A) = trace\left[\left(A^T \Sigma_B A\right)^{-1}\left(A^T \Sigma_W A\right)\right] \qquad (3.28)$$

Note that **A** is not a square matrix. As a consequence the projection index can not be reduced to $\mathrm{trace}\left[\Sigma_B^{-1}\Sigma_W\right]$ which has a closed analytic solution [18, pp. 445-4551. Sequential Projection Pursuit was used with the Fisher criterion as its projection index to project the data from Φ space to Γ subspace. Only Sequential Projection Pursuit was usecl because of the lack of global control of the Parallel approach, as shown in the previous results. Different feature extraction and selection methods will be used to project the data from Γ to Y, i.e. Discriminant Analysis, Decision Boundary Feature Extraction and feature selection. The last one uses the minimum Bhattacharyya distance as a measure of class separability.

Discriminant Analysis

In this subsection Discriminant Analysis was used as a feature extraction method after the use of Sequential Projection Pursuit and compares it with direct use of Discriminant Analysis at full dimensionality (DA 100-20). The results are poorer than direct use of Discriminant Analysis and than Projection Pursuit Based algorithms using minimum Bhattacharyya distance as a projection index. This is due to some inherent problems in the Fisher criterion index. One is that if the difference in the mean vectors is small, the features estimations will not be reliable. Another problem is that the Fisher criterion index estimates the parameters for the entire labeled data set and is not class specific. Finally it is not directly related with probability of error as Bhattacharyya distance is. Note that most of the data are thresholded on ML-2% and ECHO-2%. These suggest doubt that normality assumptions hold.

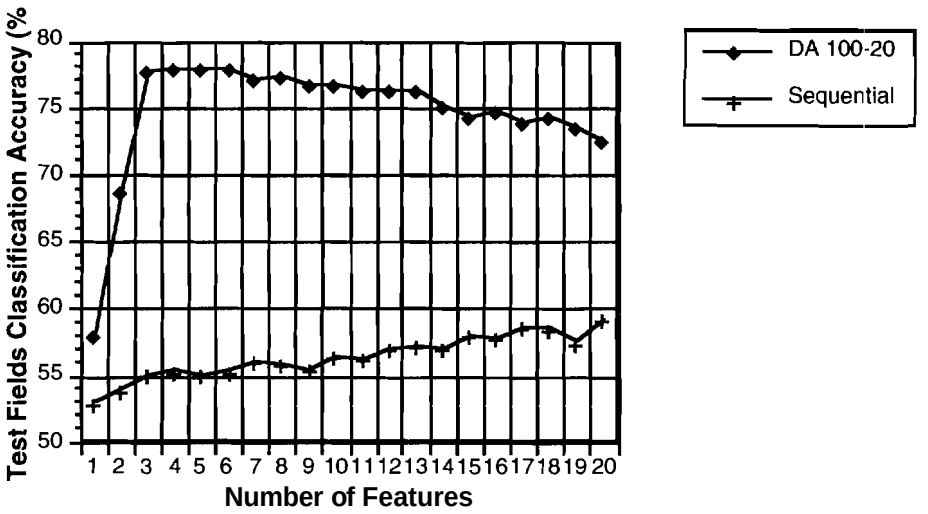


Fig. 3.38. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after a method based on Projection Pursuit (Sequential) for **ML** Classifier.

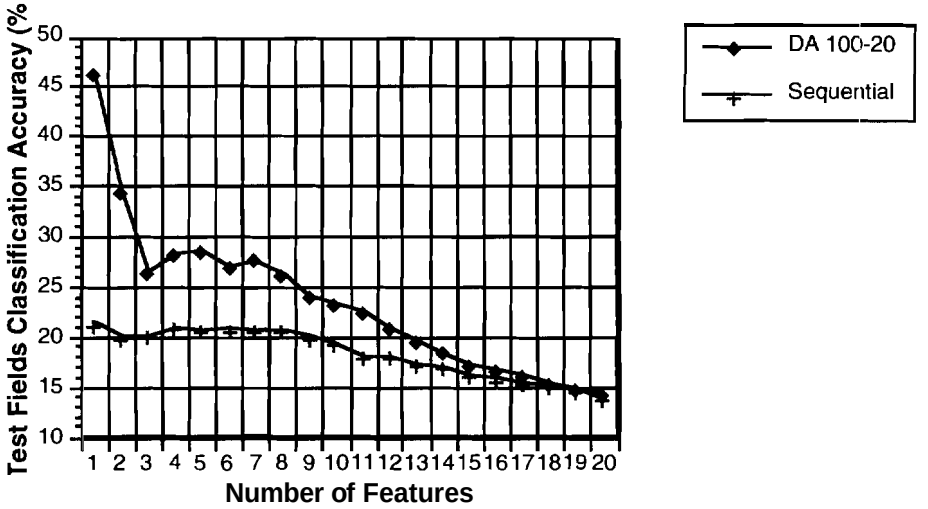


Fig. 3.39. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after a method based on Projection Pursuit (Sequential) for **ML** with threshold.

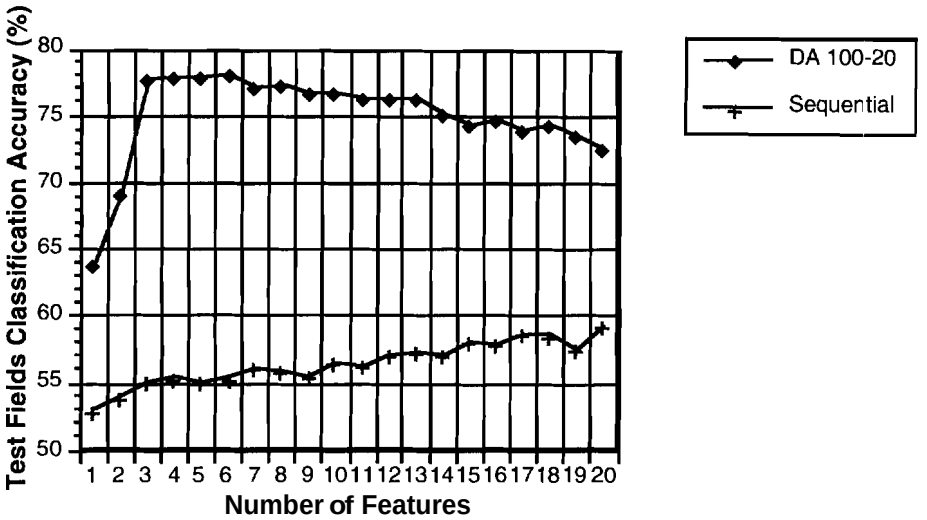


Fig. 3.40. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after a method based on Projection Pursuit (Sequential) for ECHO Classifier.

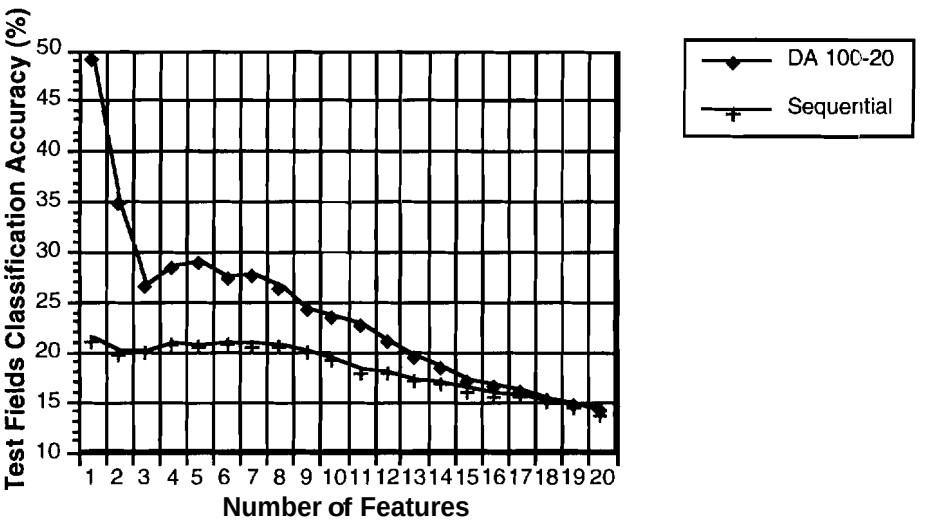


Fig. 3.41. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after a method based on Projection Pursuit (Sequential) for ECHO with threshold.

Decision Boundary

The results with Decision Boundary are similar than with the use of Discriminant Analysis as a feature extraction method. Direct use of Discriminant Analysis (DA 100-20) produces better results because of the problems mentioned of Fisher criterion, and the small number of labeled samples, a problem to which Decision Boundary is more sensitive than Discriminant Analysis.

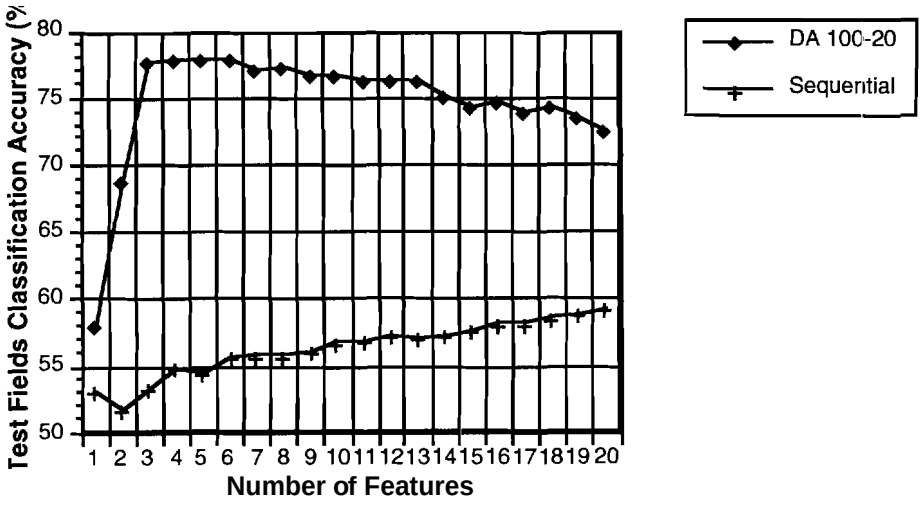


Fig. 3.42. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after a method based on Projection Pursuit (Sequential) for ML Classifier.

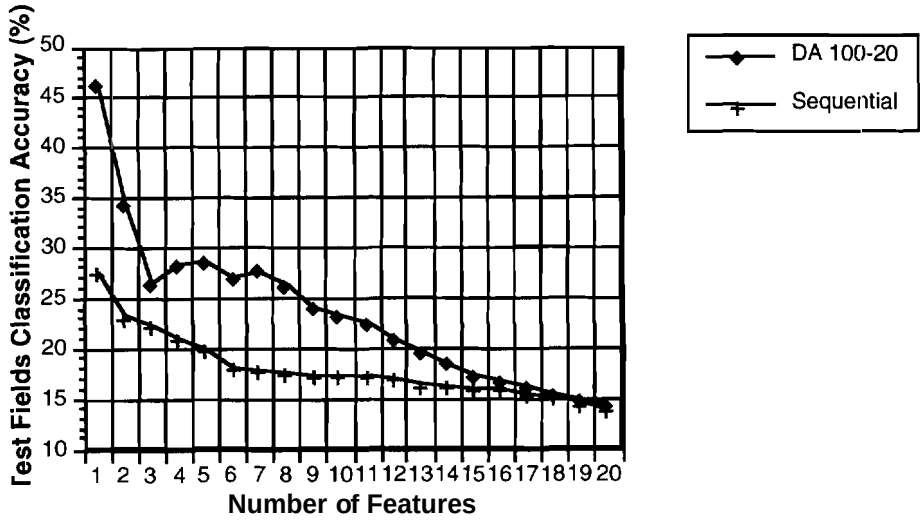


Fig. 3.43. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after a method based on Projection Pursuit (Sequential) for ML with threshold.

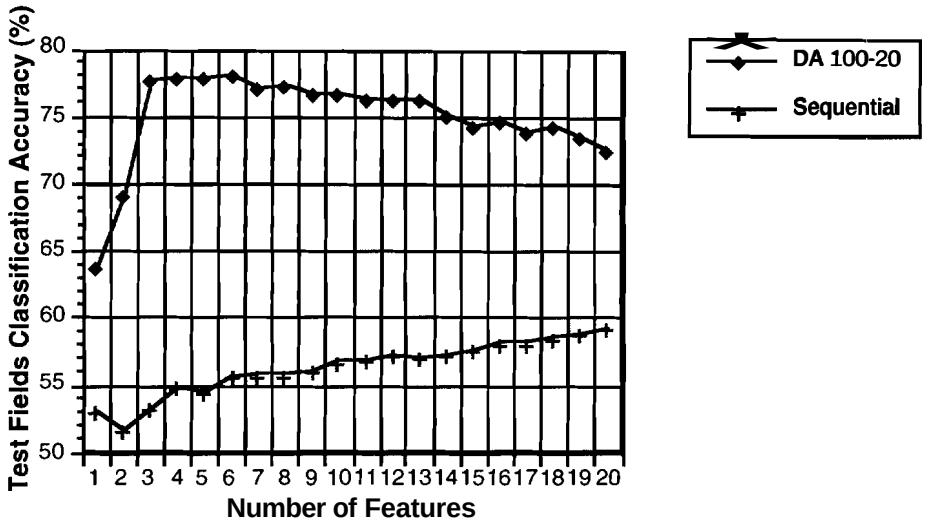


Fig. 3.44. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after a method based on Projection Pursuit (Sequential) for ECHO Classifier.

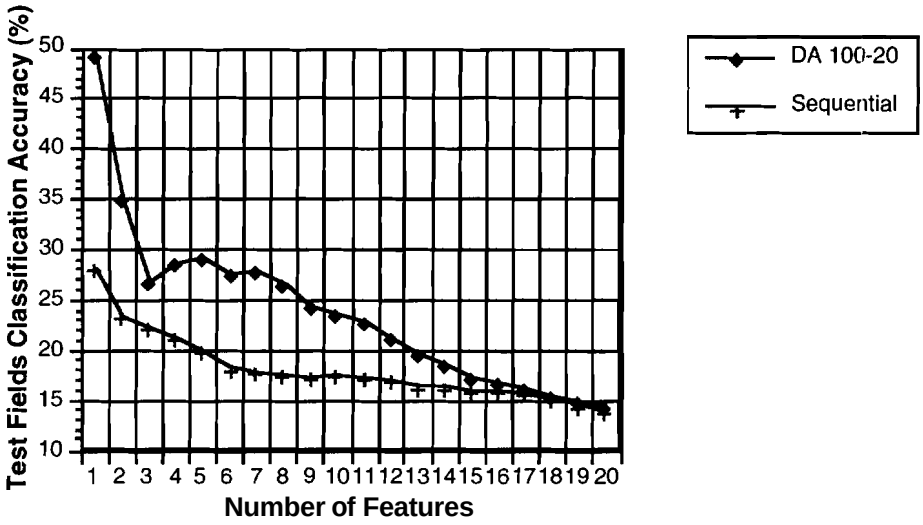


Fig. 3.45. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after a method based on Projection Pursuit (Sequential) for ECHO with threshold.

Feature Selection

The results with feature selection after using Sequential Projection Pursuit are much better than with Decision Boundary or Discriminant Analysis methods. In terms of ML classification, Discriminant Analysis at full dimensionality (DA 100-20) still performs better. With the ECHO classifier, Sequential Projection Pursuit performs better and reaches a maximum with the use of two features, then it compares with DA

100-20 until 16 features. The Sequential approach performs better with respect to ML-2% and ECHO-2%. The data is maintained together in clusters.

Even when the results of feature selection are better than with the use of Discriminant Analysis and Decision Boundary in this experiment, they are poorer than Feature Selection in experiment 1 and 2 where the projection index used is minimum Bhattacharyya distance.

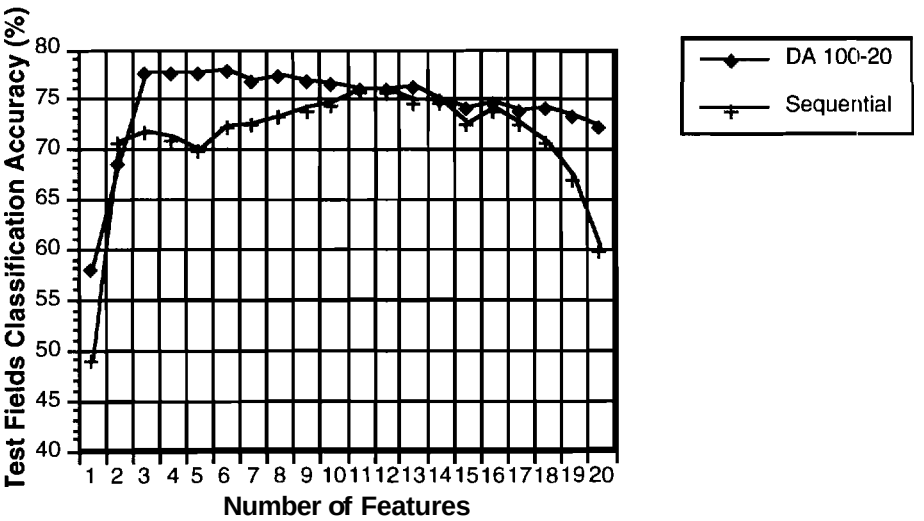


Fig. 3.46. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after a method based on Projection Pursuit (Sequential) for **ML** Classifier.

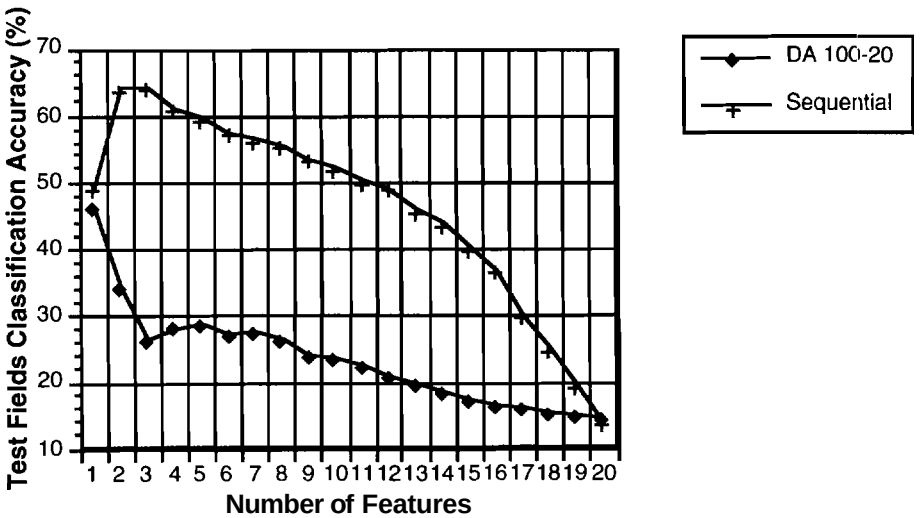


Fig. 3.47. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after a method based on Projection Pursuit (Sequential) for **ML** with threshold.

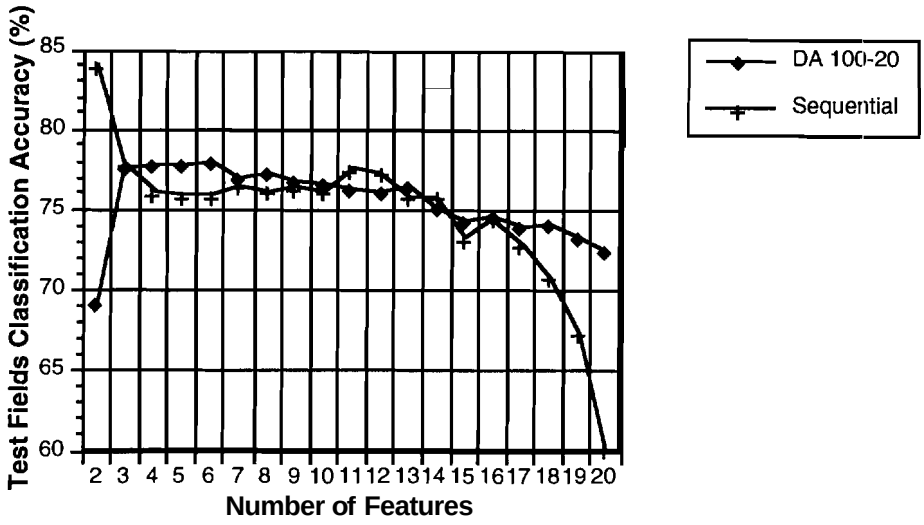


Fig. **3.48.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after a method based on Projection Pursuit (Sequential) for ECHO Classifier.

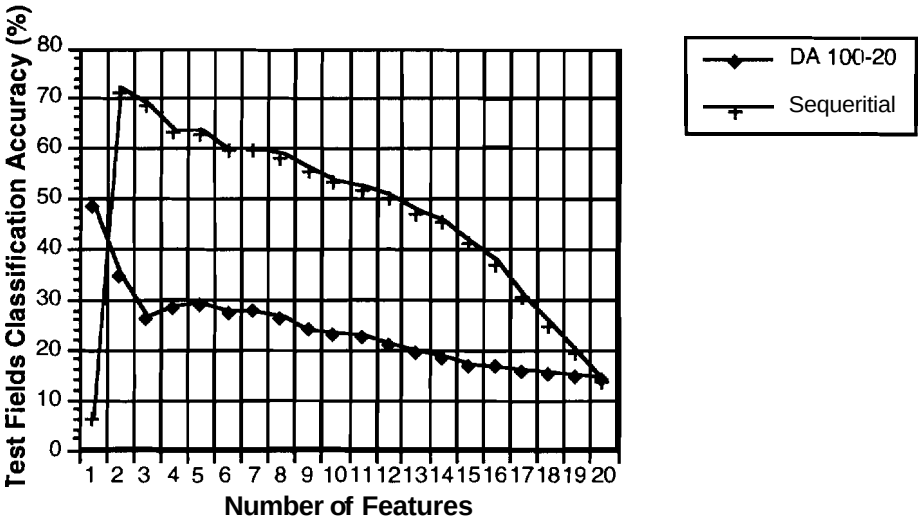


Fig. **3.49.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after a method based on Projection Pursuit (Sequential) for ECHO with threshold.

3.7 Conclusion

The increasing number of features in modern data sources augment the amount of information that should be extractable from multispectral data. At the same time, since there is usually a limit on the number of labeled samples, the effects of degrading factors such as the Hughes phenomena and other characteristics of high dimensional data are exacerbated as the number of dimensions increases. The challenge is to reduce the number of dimensions while avoiding the obstacles posed

by the above mentioned phenomenon, and while preserving maximum information and using a priori data.

A modified scheme of supervised classification had been proposed. Such modification is the result of an addition of a preprocessing algorithm with the purpose of reducing the dimensionality of the data, projecting it to a subspace where Feature Extraction or Selection is more suitable. Projection Pursuit had been the method used to develop the algorithms for accomplish such preprocessing. A parametric version was developed and used based on the use of a projection index that uses labeled samples as a priori information.

Parametric Projection Pursuit fulfills the criteria established in Chapter 1 for a preprocessing method. This procedure, performing the computations at a lower dimensional subspace, makes the assumption of normality better grounded in reality, providing better estimations of parameters and features. All of this enables the algorithm to better deal with the Hughes phenomena, maintaining the data in clusters and providing better classification accuracy.

Two approaches had been developed, Parallel and Sequential Parametric Projection Pursuit. The Parallel approach has the advantage of being faster, but it does not guaranteed that it will perform better in terms of the optimization of the overall projection index. The Sequential method had the disadvantage of being slow if it is directly implemented. Such disadvantage could be overcome to a great extent with an iterative version. The advantage that Sequential Projection Pursuit has to offer is a direct control of the projection index over the projected subspace.

The optimization of the global projection index allows more control and a better performance against the problem of local maxima and the sensitivity with the initial guess matrix $\hat{\mathbf{A}}$ than local optimization in the Parallel approach.

Two possible projection indices were tested, minimum Bhattacharyya distance among the classes and the Fisher criterion. Both use first and second order statistics. The experiments demonstrated that minimum Bhattacharyya distance performs better in terms of classification accuracy. This is due to some inherent properties of minimum Bhattacharyya distance and some problems with the Fisher function. Bhattacharyya distance is related with classification accuracy as a bound. Among some problems with the Fisher criterion there are two significant ones that could affect the calculations. The two are when the means of two classes are significantly close, and if one class mean is very different from the others. This index contains the parameter of the whole training set; meanwhile, minimum Bhattacharyya distance uses training samples

separately for all the classes. On the basis of these arguments and empirical results, minimum Bhattacharyya distance is preferred over the Fisher criterion.

4. GLOBAL OPTIMIZATION

4.1 Introduction

As discussed previously, Parametric Projection Pursuit based algorithms are sensitive in terms of arriving at a small local maximum instead to the global one. Experiments 1 and 2 of the previous chapter are examples of that problem. Figure 4.1 displays the values of the global minimum Bhattacharyya distance for the different methods used and in the different experiments, i.e. direct Discriminant Analysis (DA 100-20), Parallel Projection Pursuit at experiment 1 (PPP1) and 2 (PPP2), and Sequential Projection Pursuit at experiment 1 (SPP1) and 2 (SPP2). Some statements can be established as a consequence of the results. In the process of optimizing the projection index, in this case minimum Bhattacharyya distance, Parallel Projection Pursuit was too sensitive to the initial choice matrix. From figure 4.1 it can be observed that this scheme is not able to optimize the global projection index more than the direct application of Discriminant Analysis (DA 100-20). This is due to the fact that Parallel Projection Pursuit optimize local projection indices and as a result it has a lack of control in the overall projection index optimization. On the other hand, Sequential Projection Pursuit is more robust to the problem of small local maxima bringing about a larger optimization of the projection index . Still, an algorithm is needed to find an initial choice for matrix $\hat{\mathbf{A}}$ that enables it to arrive to an acceptable, though perhaps suboptimum solution.

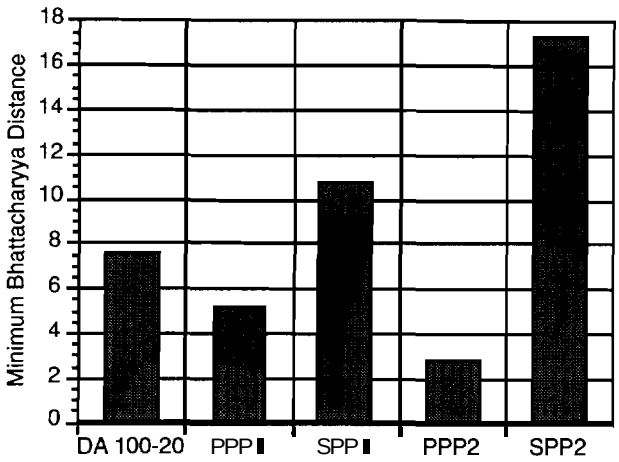


Fig. 4.1. Minimum Bhattacharyya distance produced by the different methods and different experiments.

In order to observe the importance of the initial choice for matrix $\hat{\mathbf{A}}$ and the problem of arriving at a poor local maxim, let's see an example. Project two class data from a two dimensional space to one. The statistical parameters of the data are:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 3 & -3/2 \\ 1 & 1/2 \end{bmatrix}, \text{ and } \Sigma_2 = \begin{bmatrix} 1/2 & 1 \\ -3/2 & 2 \end{bmatrix}$$

Figure 4.2 shows the Bhattacharyya distance as a function of the angle of projection of a normalized vector.

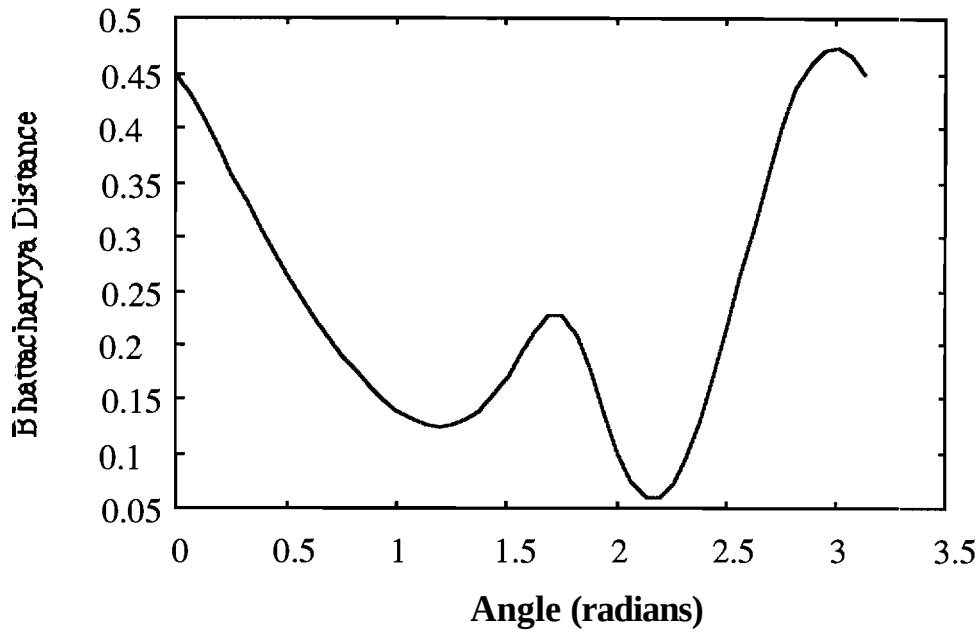


Fig. 4.2. Bhattacharyya distance for the two dimensional illustration

Note that there are two maxims. One is located at angle 1.73 radians and the other, which is global, at 3.00 radians. There is a difference of almost 250% in these two maxims. It is expected that the situation would worsen as the number of dimensions increases.

The purpose of the present chapter is to develop an algorithm that estimates A in order to overcome, as much as possible, the problem of small local maxima. In order to do that, the algorithm will estimate a set of variables in the A matrix: the initial choice vectors $\hat{\mathbf{a}}_i$ that linearly combines the adjacent bands, and the number of adjacent bands $\mathbf{n}_i, \forall i$ in every group.

In the non-parametric version of Projection Pursuit density approximation and regression the use of a two stage algorithm has been proposed in order to estimate the orientation with a better rate of convergence [47]. The first stage uses undersmoothed density estimators to estimate the orientation. The second stage uses those orientations for another estimation with a correct amount of smoothing.

An analogous idea will be developed here for Parametric Projection Pursuit.

4.2 Preprocessing Block Stages and the Initial Conditions

In order to avoid reaching a suboptimal local maximum instead of the desired global one, the preprocessing block in Figure 2.18 is divided into two stages as shown in Figure 4.3. The first one has the objective of estimating an initial choice of matrix A.

The estimation of this parametric matrix is based on the initial choice vectors $\hat{a}_i$'s and the number of adjacent bands n_i combined in each group in the partition of features shown in Figure 3.11. The second stage is the numerical optimization of the global projection index in order to estimate A, as explained in chapter 3. Tlhe focus of this chapter is in the development of an algorithm that accomplish the objectives of stage 1.

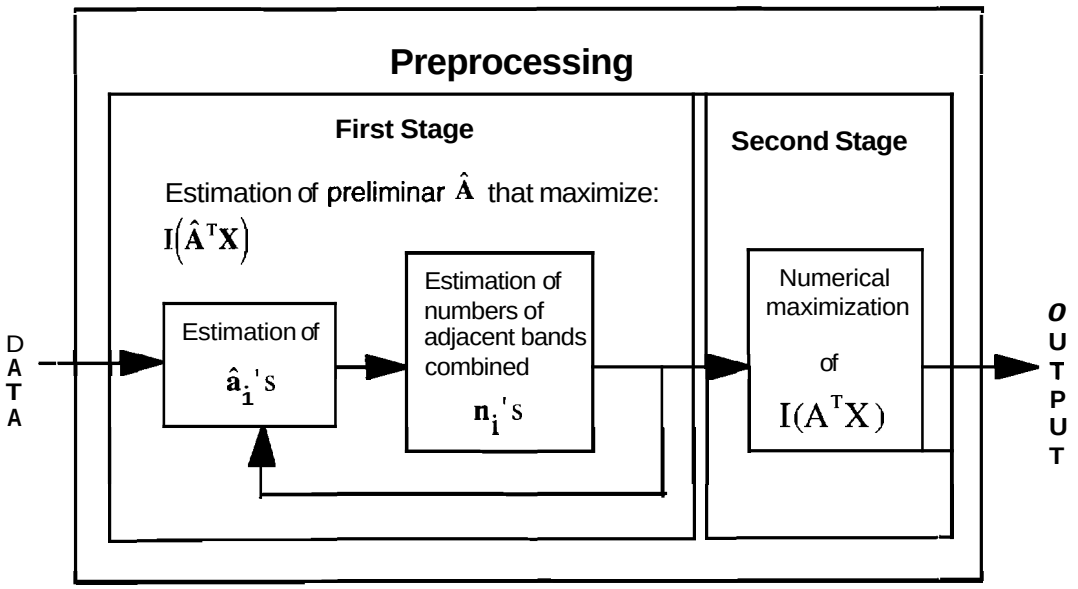


Fig. 4.3. Preprocessing block.

4.3 Estimation of the Initial Choice $\hat{a}_i$'s for Each Group of Adjacent Bands

Each group of adjacent bands will have a bank of estimated guesses $\hat{a}_1$'s. In this section we will assume that the values of n_i are given. The procedure to calculate them will be explained in section **4.4.** The matrix 6 will be constructed by choosing one estimated guess $\hat{a}_i$ from each bank. Among these guesses there are two that are very significant. The first one is based on the assumption that the mean difference is dominant in the Bhattacharyya distance. The mean difference portion of the Bhattacharyya distance is:

$$\mu_M = \frac{1}{8}\left(M_2 - M_1\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(M_2 - M_1\right) \qquad (4.1)$$

The other is based on the assumption that the covariance difference is the part that is dominant. The covariance difference portion of the Bhattacharyya distance is:

$$\mu_C = \frac{1}{2} ln \left(\frac{\left| \frac{\Sigma_1 + \Sigma_2}{2} \right|}{\sqrt{|\Sigma_1||\Sigma_2|}} \right) \tag{4.2}$$

The mean difference portion is maximized by the vector [18, pp. 455-4571:

$$a_{M\,max} = \left(M_2 - M_1 \right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} \tag{4.3}$$

In order lo compute the vector that maximizes the covariance difference element a previous matrix A must be computed. That matrix is defined as:

$$\Lambda = \Sigma_2^{-1} \Sigma_1 \tag{4.4}$$

The vector that maximizes μC, a_{Cmax} is the eigenvector of A that corresponds to the largest quantity of a function of its eigenvalue. That function is defined as:

$$\lambda_i + \frac{1}{\lambda_i} + 2 \tag{4.5}$$

These vectors and parameters are estimated to maximize the projection index in the one dimensional projected feature where each group of adjacent bands will be projected, The vectors must be estimated for every combination of two classes. Those estimates depend only on the groups of adjacent bands and are independent of the estimates of the other groups. Also in each bank a vector that averages all the features and vectors that select only one feature in that group of bands will be stored. Assuming there are K classes and n_i features in each group of adjacent bands, then the total number of initial choices $\hat{a}_i$'s in the i^{th} group of adjacent bands are:

$$Total = 2 \frac{K!}{2!(K-2)!} + n_i + 1 \tag{4.6}$$

The first element corresponds to twice the number of every conibination of two classes, corresponding to a_{Mmax} and a_{Cmax}. The second corresponds to choosing one feature from the n_i possible ones and the third to averaging.

The process of building the initial choice matrix $\hat{\mathbf{A}}$ from the estimated $\hat{\mathbf{a}}_i$ stored in each bank that belongs to each group of adjacent bands is similar to the iterative procedure of the numerical optimization of the Sequential Projection Pursuit algorithm. The procedure is as follows:

(1) Choose one $\hat{\mathbf{a}}_i$ from each bank for every group of adjacent bands. Every $\hat{\mathbf{a}}_i$ belongs to the proper place in the i^{th} column of $\hat{\mathbf{A}}$ that corresponds to the i^{th} group of adjacent bands.

(2) Maintaining the rest of the $\hat{\mathbf{a}}_i$'s constant, choose the $\hat{\mathbf{a}}_1$ from the first bank of samples that maximizes the global projection index.

(3) Repeat the procedure for each group such that the $\hat{\mathbf{a}}_i$ is chosen from the i^{th} bank of samples, meanwhile the $\hat{\mathbf{a}}_j$s for $i \neq j$ will be held constant.

(4) Once the last $\hat{\mathbf{a}}_i$ is chosen, repeat the process from step 2 until the maximization converges or stops to increase significantly.

Note thal the value of the n_i's could not be larger than the minimum number of samples per class. That will ensure a nonsingular matrix Σ_i for each class.

Observe that in the case of storing in each bank that belongs to each group of adjacent bands only vectors that select one feature in that particular group we would have a Projection Pursuit version of feature selection for high dimensional data.

Two experiments were developed with the purpose of showing the validity of this algorithm.

4.3.1 Experiment 1

This experiment has the objective of projecting two class data from a two dimensional space to one. The statistical parameters are:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

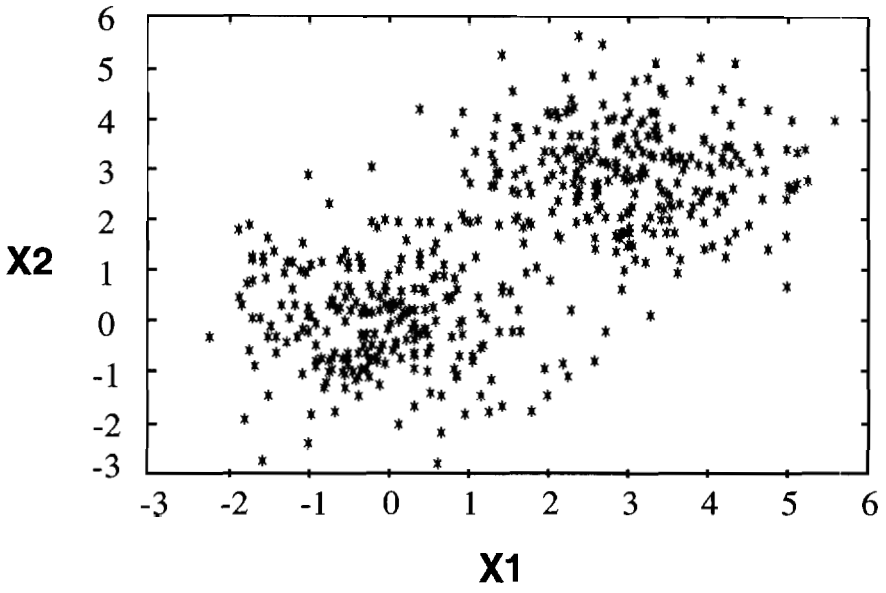


Fig. 4.4. Data set in two dimensional space.

From the parameters and Figure 4.4 it can be seen that the means' difference component is the only term that exists in the Bhattacharyya distance. Figure 4.5 shows the Bhattacharyya distance as a function of the angle of projection. The theoretical value at which the maxim is located is .78 radians. Because there are only two classes and a two dimensional space, only one bank of $\hat{\mathbf{a}}_i$ guesses is constructed. The total number of guesses in this bank is 2(1) + 2+ 1=5. Corresponding to $\mathbf{a}_{Mmax}$, $\mathbf{a}_{Cmax}$, averaging, and choosing one coordinate (X1 or X2).

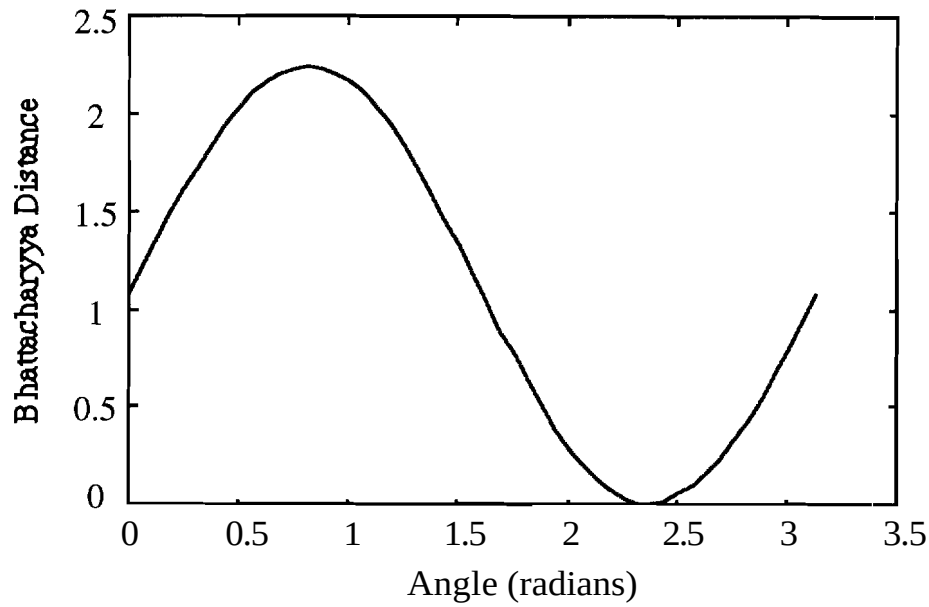


Fig. 4.5. Bhattacharyya distance.

As expected the algorithm chooses $\mathbf{a}_{Mmax}$ which corresponds to the assumption that means difference dominates. The vector $\mathbf{a}_{Mmax}$ is a normal vector with an angle of .78 radians, exactly where the theoretical maxima is.

4.3.2 Experiment 2

In the present experiment data which belongs to two statistical classes will be projected from a 2 dimensional space to one. The statistical parameter:; are:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mathbf{M}_2 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 3 & -3/2 \\ 1 & 1/2 \end{bmatrix}, \text{ and } \Sigma_2 = \begin{bmatrix} 1/2 & 1 \\ -3/2 & 2 \end{bmatrix}$$

In this particular case the Bhattacharyya distance has two components: means and covariance differences. Figure 4.6 shows the data in the two dimensional space. Figure 4.7 shows the Bhattacharyya distance as a function of the angle of projection. From there it could be seen that there is a possibility to arrive at a small local maximum (which is at 1.7272 radians) instead of at the global maxirnum (located at 3.00 radians).

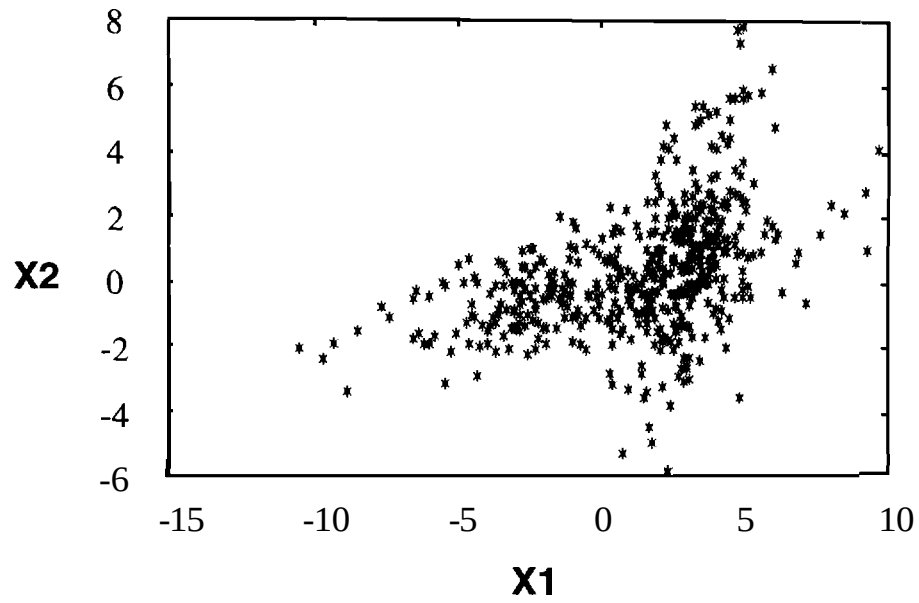


Fig. 4.6. Data set in two dimensional space.

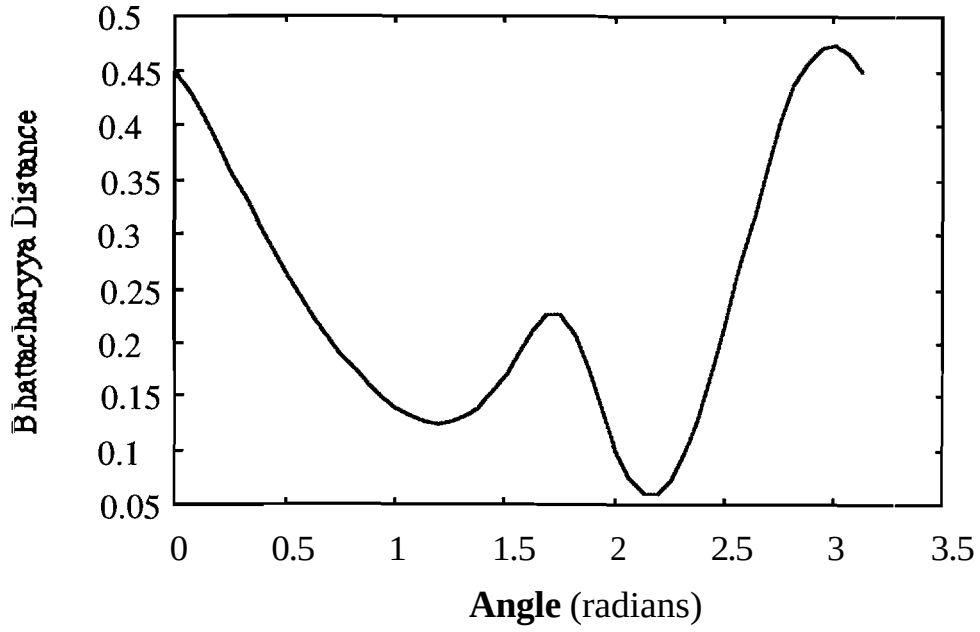


Fig. 4.7. Bhattacharyya distance.

From the five estimated guesses, the algorithm chooses a_{Mmax} which is located at .0997 radians (which is equivalent to π + .0997). Note that this guess is good enough to arrive to a global maxim with the use of a numerical optimization method. It is interesting that a_{Cmax} is located at .8909 radians. Still that guess should be enough for a numerical optimization method, but is closer to the local maxim than a_{Mmax}.

4.4 Estimation of the Number of Adjacent Bands n_i Combined in Each Group in the Partition of Features

The second block of stage one in Figure 4.3, which estimates the values of the n_i's, will be based on well-developed techniques of binary decision trees. Decision trees have been used in machine learning systems for some time [48]. Also they have been applied in pattern recognition and remote sensing image analysis. An example of their application is the design of decision tree classifiers where they have been used to partition the space in developing decision rules [49]. Some authors [50], [51] applied them in the design of hierarchical classifiers that decide at each node to which class a particular sample belongs.

The basic idea of decision trees is to break a particular complex problem into simpler ones that can be more easily solved. It is expected that solutions can be united and at least approximate the optimum global solution.

It has been demonstrated that an optimal decision tree is an N-P complete problem [52]. In terms of pattern classification four heuristic methods of Decision Tree Classifiers have been developed in order to overcome that problem: (a) top-down, (b) bottom-up, (c) hybrid and (d) tree growing-pruning. Top-down methods start to separate the samples into different groups until the final number of classes of information value is reached. Bottom-up methods have the opposite approach; starting with a group of classes, they groups classes until the root node is reached. In the hybrid approach the bottom-up procedure is used to aid the top-dlown approach. Finally in the tree growing-pruning approach the tree is allowed to grow to its maximum size and then the tree is pruned.

A binary tree algorithm will be used in this project to estimate the suboptimum number of adjacent bands that should be linearly combined in order to reduce the dimensionality. The heuristic approach used is a hybrid decision tree. In the following is explained how every heuristic approach just described can be applied in an algorithm to accomplish the objective of the second block in the first stage of Figure 4.3.

4.4.1 Top-down

This algorithm starts to collect the feature space Φ as a partition of groups of adjacent bands. Each group of adjacent bands will be projected to different features in the projected subspace Γ. As a consequence each group is equivalent to a dimension of the reduced feature subspace Γ. It is in that subspace where a final feature extraction algorithm will be applied before the classification occurs.

This algorithm begins projecting linearly the total number of features to one dimension. It estimates the projection $\hat{\mathbf{a}}_1$ that maximizes the minimum Bhattacharyya distance. At this point this algorithm integrates the previously described procedures in this chapler in section 4.3.

Starting from one group of adjacent bands, the algorithm breaks the group into a partition of two groups of adjacent bands (step 1 in Figure 4.8). Then it breaks each group indlependently of each other into two new partitions creating two sets of three dimensional space. The preliminaries optimum $\hat{\mathbf{a}}_i$'s will be calculated for each independent set. For every set of three dimensional space the increment of the global minimum Bhattacharyya distance is computed and named ΔB1 and ΔB2. Figure 4.8, step 2 shows this graphically. The algorithm chooses the largest increment in the Bhattacharyya distance (in Figure 4.8 the group with ΔB1, indicated by the dark circles). In the next step each group of adjacent bands, including the previously rejected groups (in this case the group with increment AB2 indicated by white circles), is divided independently into two groups of adjacent bands. This process creates three sels of four groups of adjacent bands corresponding to three sets of four dimensional spaces. Again the set that produces a larger increment in the global projection index is chosen (in this case a group with increment AB2 in step 3, Figure 4.8). The procedure is repeated successively in the following steps:

(a) Divide independently each group of adjacent bands into two new groups, creating new independent sets of groups of adjacent bands.

(b) For each set compute the global projection index and compute the increment in the projection index ΔB_i.

(c) Choose the set that produces the larger increment in the global projection index if the percentage increment is larger than a threshold τ_{T-D}. The percentage of increment is defined as:

$$\Delta BI_i = \frac{max\left(\Delta B_i\right)}{PI_{i-1}} \tag{4.7}$$

In the equation PI is the projection index value. The index i represents the current value, while i-1 represents the previous one. These steps are repeated until the increment in minimum Bhattacharyya distance is not larger than a threshold τ_{T-D} or until the algorithm reaches a maximum number of features established by the analyst or by the number of label samples.

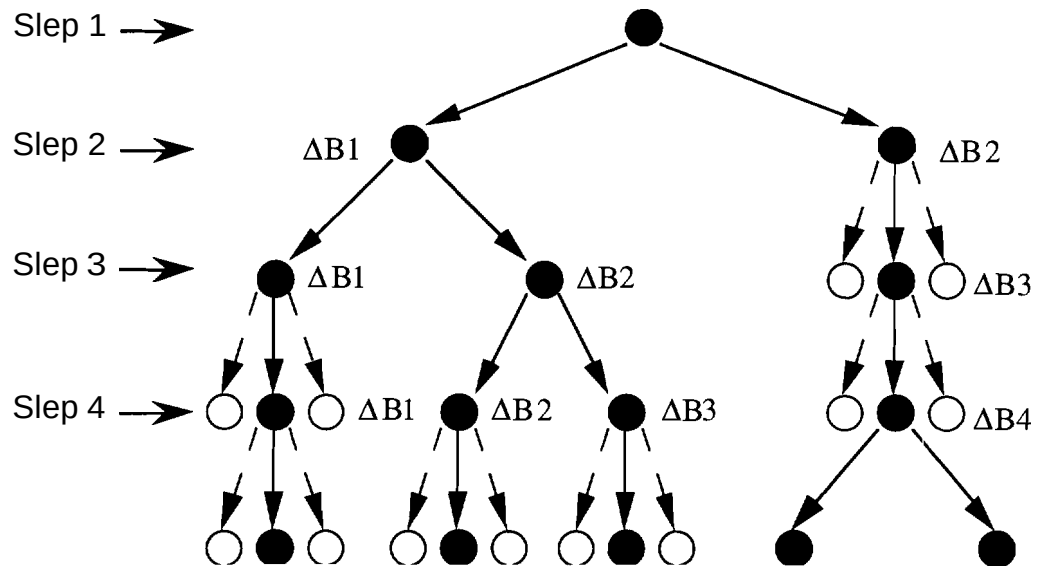


Fig. 4.8. Top-down algorithm.

In the case of an even number of adjacent features the group is divided in two equal numbers of groups. In the case of an odd number, i.e. (2N+1) either of two things could be done: (i) Choose randomly the combination of one group having N and the other N+I or (ii) Compute both possibilities as two independent sets and choose the one that produces the largest increment in the minimum Bhattacharyya distance as in step c.

The first procedure is faster. If all groups have an odd number of features, this algorithm is twice as faster as the second. The second procedure ensures choosing the optimum combination. Observe that at each step the algorithm increases by one, the number of groups of adjacent bands linearly combined in the partition. This implies that the dimensionally reduced space increases one dimension at each step. At step **k** it will create **k** independent sets of **k+I** groups of adjacent bands corresponding to **k+I** dimensional subspace Γ.

4.4.2 Bottom-up

This algorithm starts with a number of features in the dimensional projected subspace Γ, where each one corresponds to one group of adjacent bands in the partition of the high dimensional space Φ. The goal of this procedure is to reduce the number of dimensions of the lower dimensional subspace avoiding a significant reduction of the projection index.

Every two adjacent groups of adjacent bands are joined into one producing an independent set of groups of adjacent bands. For each set the preliminary optimum

$\hat{\mathbf{a}}_i$'s will be calculated. Like in top-down, here this algorithm integrates the procedure described in section 4.3. Then for each independent set the decrease in projection index ΔB_i is computed. It is important to note here that ΔB_i is an absolute value measure always positive in the equations. The algorithm chooses the set that produces the minimum reduction in the projection index if the percentage of decrease is smaller than a defined threshold τ_{D-T}. The percentage of decrease is defined as:

$$ABD_i = \frac{min\left(\Delta B_i\right)}{PI_{1-1}} \tag{4.8}$$

where PI is defined as in top-down procedure. The procedure can be repeated, creating new sets of dimensionally reduced spaces by combining adjacent groups of adjacent bands, including those previously rejected as shown in Figure 4.9.

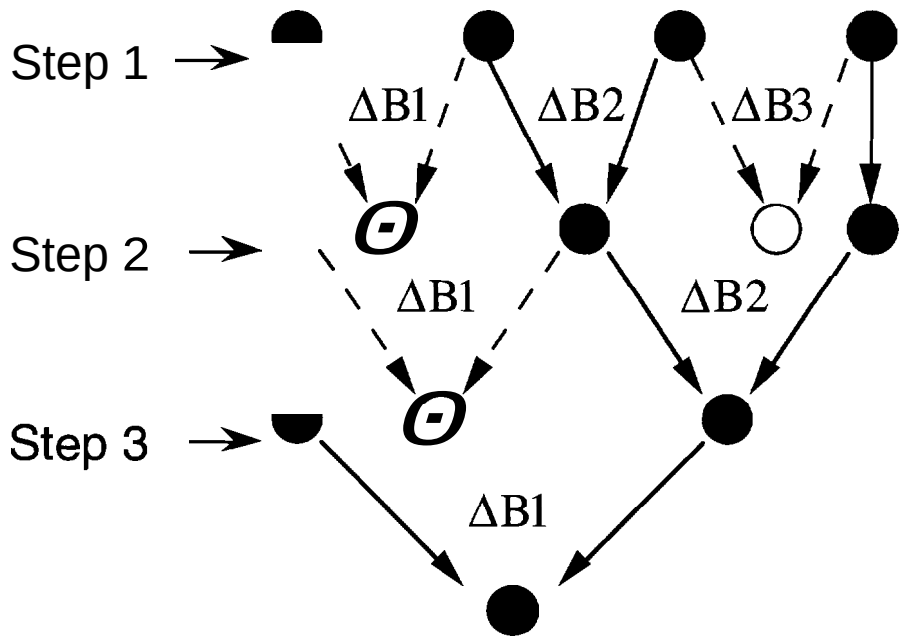


Fig. 4.9. Bottom-up algorithm.

At step **k** it will produce **k-1** independent sets each one with **k-1** groups of adjacent bands corresponding to subspaces of **k-1** dimensions.

4.4.3 Hybrids

There are two types of hybrids or combinations of these two groups:

Hybrid I

Starting with the top-down procedure the present algorithm allows the tree to grow until it reaches its maximum number of features. There are two ways to decide when the algorithm arrives at a maximum: the maximum number is supplied by the analyst taking into consideration the number of labeled samples and other factors, or until the percentage of growth of ABI is less than a threshold τ_{T-D}. Then apply the bottom-up procedure in order to reduce the number of features. This last step is allowed to reduce the dimensionality until it reaches a minimum number of features supplied by the analyst or until its percentage of reduction ABD is larger than the threshold τ_{D-T}.

Hybrid II

This procedure results by interchanging both algorithms: top-down and bottom-up. Starting with the top-down procedure increase the dimensions of the subspace by 1. Then use bottom-up to verify that it can reduce by one dimension without decreasing the projection index significantly. In order to avoid an infinite loop the relationship between the thresholds should be $\tau_{D-T} \leq \tau_{T-D}$. This algorithm should stop when both algorithms sequentially fail to meet the requirements with respect to the thresholds or when it arrives at a maximum or minimum number of features provided by the analyst or limited by the number of training samples. Hybrid I is significantly faster, however Hybrid II is more efficient especially when the number of labeled samples is quite small.

The top-down binary tree has some characteristics that resemble a greedy algorithm. A greedy algorithm has the attribute that, at each step, it makes the choice that looks better at the moment. It makes locally optimal choices with the hope that it will lead to a globally optimal solution [53]. The fundamental difference is that in the top-down algorithm every choice is not limited to the children of the chosen nodes. Every choice includes all nodes.

The bottom-up tree at the same time resembles some elemenits of a dynamic programrning algorithm, i.e. the binary parsed tree. The similarity is that it combines groups of adjacent channels with a minimum loss of projection index.

4.5 High Dimensional Projection Pursuit Feature Selection

Frorn now on we will call the Parametric Sequential Projection Pursuit algorithm just Projection Pursuit. It will use the methods in sections 4.3 and 4.4 of this chapter equivalent to stage 1 in Figure 4.3 in order to estimate A. Then it uses a numerical

optimization algorithm equivalent to stage 2 in Figure 4.3 to finally compute **A**. Projection Pursuit Feature Selection uses the method explained in sections 4.3 and 4.4 in the present chapter with a significant transformation. Every bank described in section 4.3 will only contain vectors that choose one feature in every group of adjacent bands. It follows the procedure described in that section to choose which vectors will maximize the global minimum Bhattacharyya distance. Through the feedback shown in Figure 4.3 it also estimates a suboptimum width of each group of adjacent bands. In this method there is no second stage, i.e., numerical optimization of the projection index. This algorithm has significant fewer computations in high dimensional data than a normal feature selection algorithm as described in chapter 3

4.6 Experiments

A series of experiments had been developed in order to test the! algorithm. The first experiment was designed to test the algorithm with a ten dimensional generated data. The first and second order statistics are known. This experiment will calculate two matrices A, one for Projection Pursuit and the other for Projection Pursuit Feature Selection with their Bhattacharyya distances and the final **A** for Projection Pursuit.

The second experiment uses real multispectral data from an AVIRIS frame. The objective is to use the first stage algorithm to calculate $\hat{\mathbf{A}}$ for Projection Pursuit and Projection Pursuit Feature Selection. Then it calculates **A** with a numerical analysis stage. It compares them with direct use of Discriminant Analysis at full dimensionality in the Φ space and verifies how this algorithm is enhanced by Projection Pursuit in terms of test field classification accuracy. This experiment represents the case of having a small number of classes and training samples.

The third experiment has the purpose of testing the algorithm against the case of having a relative larger number of classes, and training samples. Projection Pursuit was used to see how it enhances the performance of two known feature extraction schemas; Decision Boundary Feature Extraction and Discriminant Analysis, in terms of classification accuracy. Both of those algorithms were applied at full dimensionality and their fields classification accuracy results were compared with their application after Projection Pursuit was used.

4.6.1 Experiment 1

The purpose of this experiment is to test the first and second stage of preprocessing in generated data with known statistics. It will be a test of how well the first stage estimates the $\mathbf{n}_i$'s and the final dimensionality of the data for Projection

Pursuit and Projection Pursuit Feature Selection. The data for this experiment were generated using the following first and second order statistics:

$$\mathbf{M}_1 = [.5 \quad .5 \quad 2 \quad 2 \quad 2 \quad 1.7 \quad 1.7 \quad 1 \quad 1 \quad 1]^T$$
$$\mathbf{M}_2 = [-.5 \quad -.5 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad -1 \quad -1 \quad -1]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 20 & -17 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 32 & -32 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -17 & -32 & 113 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 41 & 15 & 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 99 & -14 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & -14 & 58 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}$$

The theoretical Bhattacharyya distance is 3.675 and the estimated Bhattacharyya distance is 3.823. It is important to note that the algorithm will use estimated parameters. In this case the estimated Bhattacharyya distance is the measure used to compare the others.

The original number of features is ten and the number of samples per class is 500. In this experiment the hybrid version used for Projection Pursuil: and Projection Pursuit Feature Selection is the hybrid II approach for the first stage. The thresholds to finish are the same $\tau_{T-D} = \tau_{D-T} = .005$. It is generated data where groups of adjacent channels influence each other. In the first two channels the mean difference is predominant. The covariance dominates in the third, fourth and fifth channel. The sixth and the seventh channel are a mixture of mean and covariance differences. The eight, nine and tenth have mean difference dominance.

Projection Pursuit

In this part of the experiment the Projection Pursuit algorithm was used. Table 4.1 shows the results in terms of number of features, the number of adjacent features combined in each group, which is the vector **n,** the Bhattacharyya distance for the matrix A (PP1) and the Bhattacharyya distance for **A,** after the numerical optimization (PP2). These two matrices were generated by the binary tree method in a first stage algorithm explained in section 4.3 and 4.4 of this chapter and by the numerical optimization method explained in chapter 3.

Table 4.1

Number of Features	n	Stage 1 (Binary Tree) PP1	Stage 2 (Numerical opt.) PP2
1	[10]	2.9182	2.9182
2	[5 5]	3.2901	3.2901
3	[2 3 5]	3.5306	3.5312
4	[2 3 3 2]	3.6523	3.6529
5	[2 3 1 2 2]	3.6875	3.6880
6	[2 2 1 1 2 2]	3.7179	3.7252
7	[2 1 1 1 1 2 2]	3.8104	3.8109

Observe the division in 4 bands. It almost fits the different groups of adjacent bands. It does not fit exactly because the parameter are being estimated and are not exactly as the theoretical used to generate the data. That could also be because the feature seven and eight are a mixture of mean and covariance difference. For groups where the mean difference is dominant, it almost did not break them. For groups where covariance difference is dominant, it divided until having groups of single features. That is expected because pure covariance difference domination should require more features to preserve information. Another important observation is that in the first stage calculation, the algorithm that computes $\hat{\mathbf{A}}$ in this case was almost enough to estimate the sub-optimum transformation. For thresholds of value .005, the algorithm stops at 7 features. The values of PP1 and PP2 were close; it almost did not need a numerical optirnization.

Projection Pursuit Feature Selection

This part of the experiment uses the Projection Pursuit Feature Selection algorithm. It does not use the numerical optimization of a second stage. The first stage only uses vectors of the form: [0 ... 0 1 0 ... 0] in the guessed estimation in each bank of adjacent bands. It requires a larger dimensionality in the projected subspace (9 vs. 7) than the previous experiment for the thresholds $\tau_{D-T} = \tau_{T-D} = .005$. 'The results are shown in table 4.2.

Table 4.2

Number of Features	n	Stage 1 (Binary Tree) PP1
1	[10]	0.5065
2	[5 5]	0.6741
3	[3 2 5]	1.1249
4	[3 2 3 2]	1.6022
5	[3 2 3 1 1]	3.3994
6	[2 1 2 3 1 1]	3.5366
7	[2 1 1 1 3 1 1]	3.6549
8	[1 1 1 1 1 3 1 1]	3.7629
9	[1 1 1 1 1 1 2 1 1]	3.8174

Note that the algorithm stop at a number of dimensions close to the number at full dimensionality. The values of PP1 in PPFS are less, than PP1 and PP2 values of PP found in table 4.1.

4.6.2 Experiment 2

The multispectral data used in these experiments is a segment of AVIRIS data taken of NW Indiana's Indian Pine test site. From the original 220 spectral channels 200 were used, discarding the atmospheric absorption bands. In the present experiment four classes were defined: corn, corn-notill, soybean-min, and soybean-notill. The total number of training samples is 179 (less than the number of bands used) and the total number of test samples is 3501. Table 4.3 shows the number of training and test samples for each class.

Table 4.3

Classes	Training Samples	Test Samples
Corn-notill	52	620
Soybean-notill	44	737
Soybean-min	61	1910
Corn	22	234
Total	179	3501

The multispectral data was reduced in dimensionality from 200 dimensions in Φ space to 20 dimensions by three methods: (1) using direct Discriminant Analysis as a feature extraction method to project from 100 to 20 dimensions (DA 100-20), (2) Sequential Projection Pursuit having only a numerical maximization stage (PP) , and (3) Projection Pursuit with a first stage that estimated matrix A (PP-Opt) and to 16 dimensional subspace Γ by one method: (4) Projection Pursuit Feature! Selection (PP-Opt-FS). DA 100-20, one of the few known feature extraction algorithms that can be used to extract high dimensional information without estimating singular matrices with such small number of label samples. Using Discriminant Analysis the data was reduced from 100 bands (one in every two bands from the original 200) to a 20 dimensional subspace Ψ. From the original number of bands, 100 were used because of the limited number of training samples (179). Iterative Sequential Projection Pursuit (PP) was applied to the data in order to reduce the dimensionality, maximizing the minimum Bhattacharyya distance among the classes. In this approach the number of adjacent bands combined in each group was 10 and the initial choice vector for maximization was chosen to be a vector that averages the adjacent bands on a group. This approach only has a numerical optimization method. It was used as a measure of improvement of performance of Projection Pursuit with a first stage named in this experiment Projection Pursuit optimized (PP-Opt). Projection Pursuit Feature Selection (PPFS) and the optimum version of Sequential Projection Pursuit (PP-Opt) were used as described in sections 4.4 and 4.5. Both use the hybrid II heuristical approach to construct the a priori matrix A with thresholds τ_{T-D} and τ_{D-T} equal to .005.

In the Projection Pursuit based algorithms, after the dimensionality of the data was reduced, Discriminant Analysis, Decision Boundary and feature selection were used as feature extraction algorithms in order to project the data from Γ to Ψ. The feature selection method used minimum Bhattacharyya distance as a measure of statistical distance among the classes.

Four types of classifiers were used. The first one is ML classifier, the second is ML with 2% threshold. The third is a spectral-spatial classifier named ECHO [45] [46] and the fourth is ECHO with 2% threshold. In the second and the fourth a threshold was applied to the standard classifiers whereby in case of normal distribution of the data 2% of the least likely points will be thresholded. These 2% provide one indication of how well the data fit the normal model and are maintained in clusters that represent statistical classes. All of these classifiers performed a projection from Ψ to the resulted

space $2. All of these schemes of preprocessing, feature extraction, and data analysis are summarized in Table **4.4.**

Table **4.4**

Case	Preprocessing $\Phi \rightarrow \Gamma$	Feature Extraction $\Gamma \rightarrow \Psi$	Classifier $\Psi \rightarrow \Omega$
1	N/A	Direct use of Discriminant Analysis $\Phi \rightarrow \Psi$ (DA 100-20)	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%
2	Projection Pursuit with only numerical optimization (PP)	(a) Discriminant Analysis (b) Decision Boundary (c) Feature Selection	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%
3	Projection Pursuit with First and Second Stage (PP-Opt)	(a) Discriminant Analysis (b) Decision Boundary (c) Feature Selection	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%
4	Projection Pursuit Feature Selection (PP-Opt-FS)	(a) Discriminant Analysis (b) Decision Boundary (c) Feature Selection	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%

Projection Pursuit

Table 4.5 shows the results of the partition of groups of adjacent bands. It starts at ten because one class has only 22 labeled samples (corn). That will imply that the estimated covariance matrices, which are needed to estimate the $\hat{a}_i$'s cannot be larger than 22 - 1 That will ensure a nonsingular estimation of the covariance matrix. The program subtracts two to the minimum number of labeled samples per class instead of one, which will make the maximum number of adjacent features in a group being 20. At the sarne time it stops at 20 because the algorithm is defined to stop at the minimum nurnber of labeled samples per class - 2.

Table 4.5

Number of Features	**n**
10	[20 20 20 20 20 20 20 20 20 20]
11	[20 10 10 20 20 20 20 20 20 20 20]
12	[20 10 10 20 20 10 10 20 20 20 20 20]
13	[20 10 10 10 10 20 10 10 20 20 20 20 20]
14	[20 10 10 10 10 20 10 10 10 10 20 20 20 20]
15	[20 10 10 10 10 20 10 10 10 10 20 20 20 10 10]
16	[20 10 5 5 10 10 20 10 10 10 10 20 20 20 10 10]
17	[20 10 5 5 10 10 20 5 5 10 10 10 20 20 20 10 10]
18	[20 10 5 5 10 10 20 5 5 10 10 5 5 20 20 20 10 10]
19	[20 10 5 5 10 10 20 5 5 10 10 5 5 20 10 10 20 10 10]
20	[20 10 5 5 10 10 20 5 5 10 10 5 5 20 5 5 10 20 10 10]

Table 4.6 shows the values of the projection index for $\hat{\mathbf{A}}$ for each partition of group of adjacent bands. Only the last partition and it estimated $\hat{\mathbf{a}}_i$'s will be given to a numerical optimization method.

Table 4.6

Number of Features	PP1 - Minimum Bhattacharyya Distance
10	5.7136
11	6.6216
12	7.2698
13	7.5288
14	8.3720
15	8.7819
16	9.3800
17	9.8638
18	10.3147
19	10.8491
20	11.2186

Projection Pursuit Feature Selection

The $\hat{\mathbf{A}}$ here was generated using Projection Pursuit Feature Selection algorithm. Unlike the Projection Pursuit optimum, it starts to build the Γ space from one dimension because it does not need to compute any feature based on the first and second order statistics. Table 4.7 and 4.8 show the results as the Γ space was built for different partition of groups of adjacent bands.

Table 4.7

Number of Features	n
1	[200]
2	[100 100]
3	[50 50 100]
4	[25 25 50 100]
5	[25 13 12 50 100]
6	[12 13 13 12 50 100]
7	[12 13 13 12 25 25 100]
8	[12 13 13 12 25 25 50 50]
9	[12 13 13 12 13 12 25 50 50]
10	[12 13 13 12 13 12 25 25 25 50]
11	[12 13 13 12 6 7 12 25 25 25 50]
12	[12 7 6 13 12 6 7 12 25 25 25 50]
13	[12 7 6 13 12 6 7 12 25 12 13 25 50]
14	[12 7 6 13 6 6 6 7 12 25 12 13 25 50]
13	[12 7 6 19 6 6 7 12 25 12 13 25 50]
14	[12 7 6 9 10 6 6 7 12 25 12 13 25 50]
15	[12 7 6 9 10 6 6 3 4 12 25 12 13 25 50]
16	[6 6 7 6 9 10 6 6 3 4 12 25 12 13 25 50]
15	[6 6 7 6 9 10 6 6 3 4 12 25 25 25 50]
16	[6 6 7 6 9 10 6 6 3 4 12 12 13 25 25 50]

Table 4.8

Number of Features	PP1 - Minimum Bhattacharyya Distance
1	0.1790
2	0.3689
3	1.2999
4	2.5719
5	3.0469
6	3.3786
7	3.7681
8	4.4081
9	4.9991
10	5.6360
11	5.9841
12	6.5579
13	6.9356
14	7.2868
13	6.9356
14	7.3654
15	7.8199
16	8.2205
15	7.8551
16	8.3080

The dimensionality of the projected subspace was not able to grow after 16 features because it could not grow more than 5% (thresholds values are .005). Note that the case of 13, 14, 15, and 16 features were repeated because of the loop created in the hybrid II algorithm, given the interchange between top-down and bottom-up algorithms. Projection Pursuit optimum after the estimation of A, uses a numerical optimization method in order to accomplish the second stage of Figure 4.3. It increases the minimum Bhattacharyya distance from 11.2186 in the first stage to 18.30.

The minimum Bhattacharyya distance among the classes was calculated for the three data sets at a 16 dimensional space for PP-Opt-FS, and in a 20 dimensional space for DA 100-20, PP, and PP-Opt. The results are shown in Table 4.9.

Table 4.9
Minimum Bhattacharyya Distance among the classes

	DA 100-20	PP-Opt-FS	PP	PP-Opt
Min. Bhatt. Dist.	7.53	8.33	10.73	18.30

Observe that the Projection Pursuit based algorithms preserved more information in terms of minimum Bhattacharyya distance than direct use of Discriminant Analysis at Φ space. The result is based on the fact that Discriminant Analysis makes the computation at full dimensionality (100 dimensions) with a small number of labeled samples (179 samples). Meanwhile the Projection Pursuit based algorithms make the computation and directly maximize the projection index in the 16 or 20 final dimensional space. Another factor is that Discriminant Analysis calculates the features maximizing another index than Bhattacharyya distance, i.e., Fisher criterion. Observe that Projection Pursuit Feature Selection compares favorably with Discriminant Analysis. Also Projection Pursuit optimization using the first stage loop before the numerical optimization (PP-Opt), as described in section 4.4, has the best performance. It has an improvement of around 83% over Projection Pursuit which only has a numerical optimization stage (PP). It avoids, better than the others, the problem of reaching a small local maximum.

The subsequent subsections will show the results of projecting the preprocessed data from the Γ subspace to Ψ with different feature extraction or selection methods in

order to compare them with direct projection from Φ space to Ψ using Discriminant Analysis (DA 100-20). The comparison will be in terms of test fielcls classification accuracy. Because of the small number of training samples, their classification results are not that relevant.

Feature Extraction Methods

Discriminant Analysis

This feature extraction method was used to project data from the Γ subspace to Ψ after the Projection Pursuit based methods were applied. It will provide the most direct comparison against direct projection from Φ to Ψ (DA 100-20) because the same feature extraction procedure was used either at Φ space and at Γ subspace.

After Discriminant Analysis was applied to data sets preprocessed by Projection Pursuit based algorithms, they were classified and the test fields classification results can be seen in Figures 4.10, 4.11, 4.12, and 4.13. The classification accuracy results on the test fields using the Maximum Likelihood classifier can be seen in Figure 4.10.

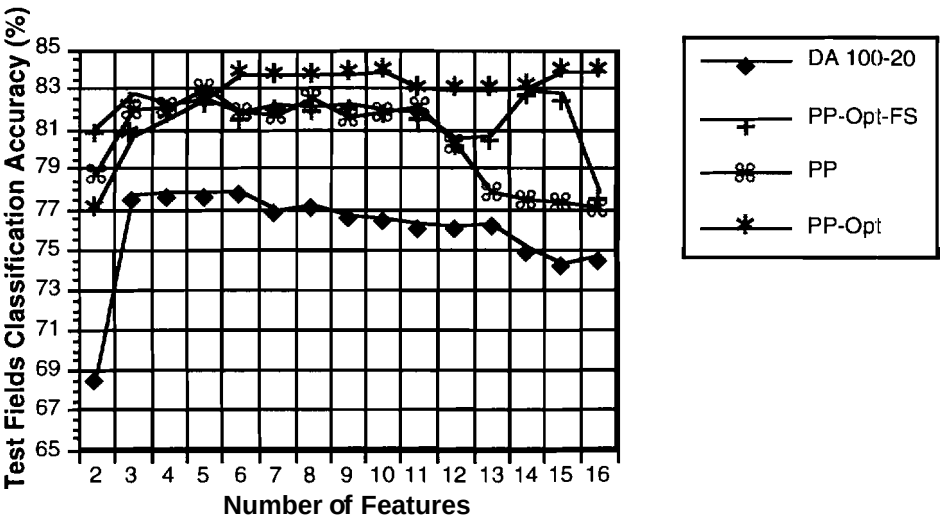


Fig. 4.10. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML classifier.

Observe in Figure 4.10 that Projection Pursuit's classification accuracies are much better than using direct Discriminant Analysis (100-20). Projection Pursuit optimization becomes the best method as the number of dimension increases. It better overcomes the Hughes phenomena and the geometrical and statistical properties of

high dimensional space. Projection Pursuit without the first stage of optimization (PP) did not handle the Hughes phenomena as the dimensions increase as well as PP-Opt or PP-Opt-FS. From Figure 4.11 it can be seen that the Projection Pursuit approaches performed significantly better, with a difference sometimes of 45%, than Discriminant Analysis directly applied to 100 dimensions, when a threshold is applied in a classifier. This may be due to the fact that in all approaches the computation is made in a small dimensional space where the assumption of normality is more suitable. This allows the computation to deal more effectively with the Hughes Phenomena, preserving more information and enabling Discriminant Analysis to make the computation at lower dimensionality with the same number of label samples.

ECHO and ECHO-2% have similar results than ML (which only takes into consideration spectral information) and it confirms what it had been said. The only difference is that the ECHO classifier accuracies are better due to the addition of spatial contextual information.

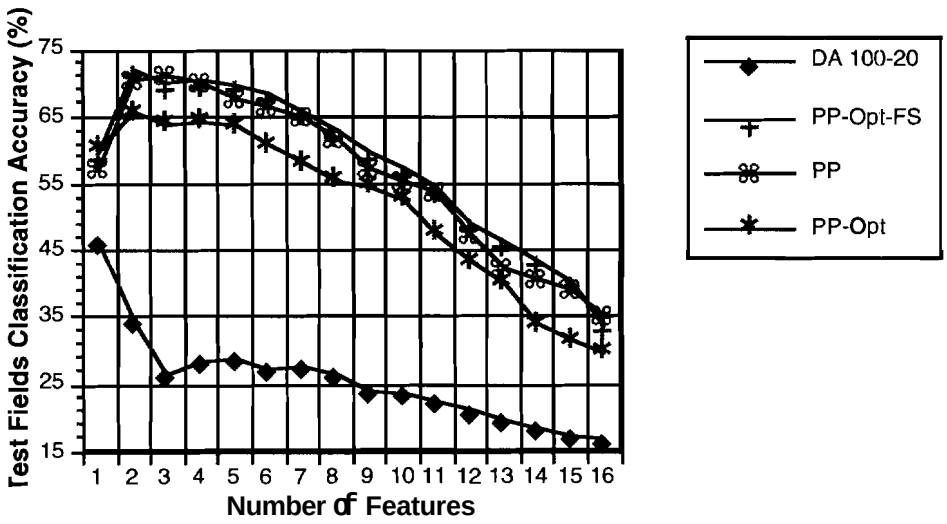


Fig. 4.11. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML with 2% threshold.

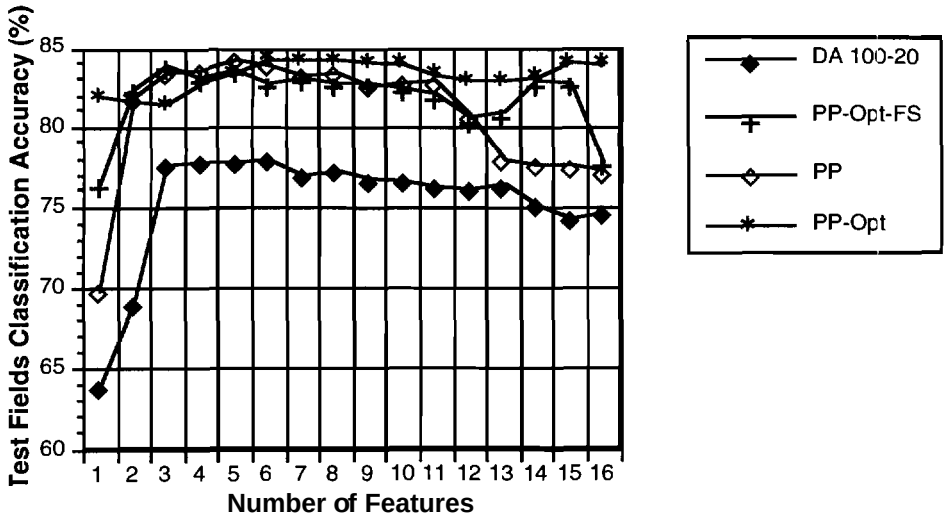


Fig. 4.12. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO classifier.

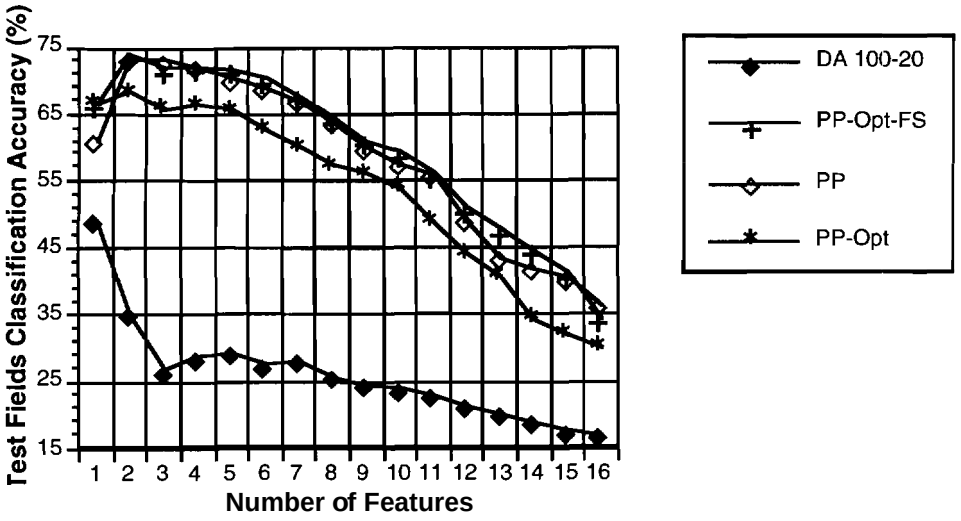


Fig. 4.13. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO with 2% threshold.

Decision Boundary

This feature extraction algorithm was used to project data from Γ to Ψ after the use of Projection Pursuit based algorithms and compare its results with direct use of Discriminant Analysis at high dimensional space. Decision Boundary could not be used at 200 bands to project the data from Φ to Ψ, because it required at least 201 samples per class. Test fields accuracy in Figures 4.14, 4.15, 4.16 and 4.17 show that

the difference between DA 100-20 and Decision Boundary applied after Projection Pursuit based algorithms at 20 dimensions is small. Still the classifications with thresholds show that Projection Pursuit based preprocessing approaches have a better grounded assumption of normality.

In this case there is no correlation between the minimum Bhattacharyya distance and the petformance of Decision Boundary. Projection Pursuit optimization has the poorest petformance. The results in the ML and ECHO classifiers could be explained by the fact that Decision Boundary demands more samples than Discriminant Analysis. It is more sensitive to the number of training samples than the separation of statistical classes. PP-Opt-FS classification results were better because it is doing the computation in a 16 dimensional space. It shows how sensitive the Decision Boundary method is to the number of label samples and the dimensionality parameters. The results suggest the use of a more relaxed threshold (> .005) with Decision Boundary. These results are more a comparison between Decision Boundary and Discriminant Analysis.

The ECHO classifier results confirm what had been said already with the ML results. One of the differences is that at an small dimensionality (2 features) PP-Opt-FS was able to obtain the maximum results, 85%. The second difference is that PP-Opt was able to maintain the data more in clusters in a small dimensionality (one feature)as shown in Figure 4.17.

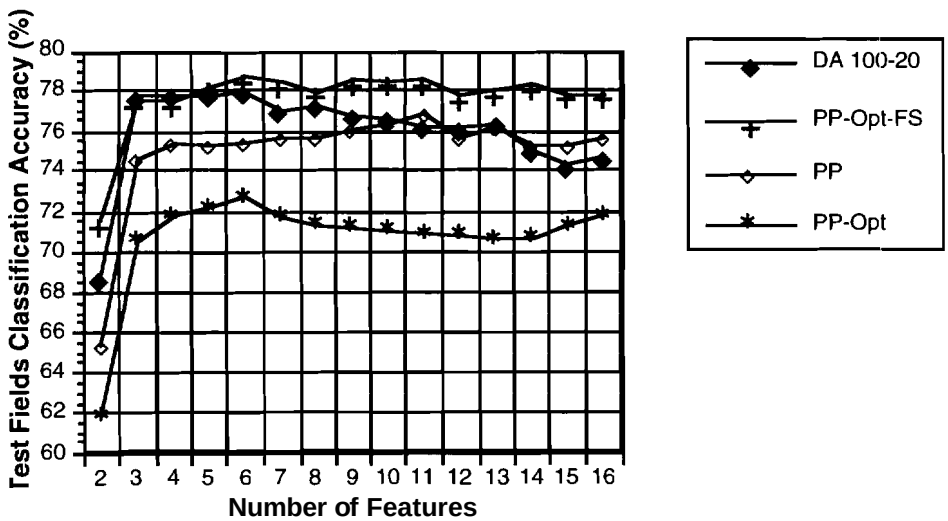


Fig. 4.14. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML classifier.

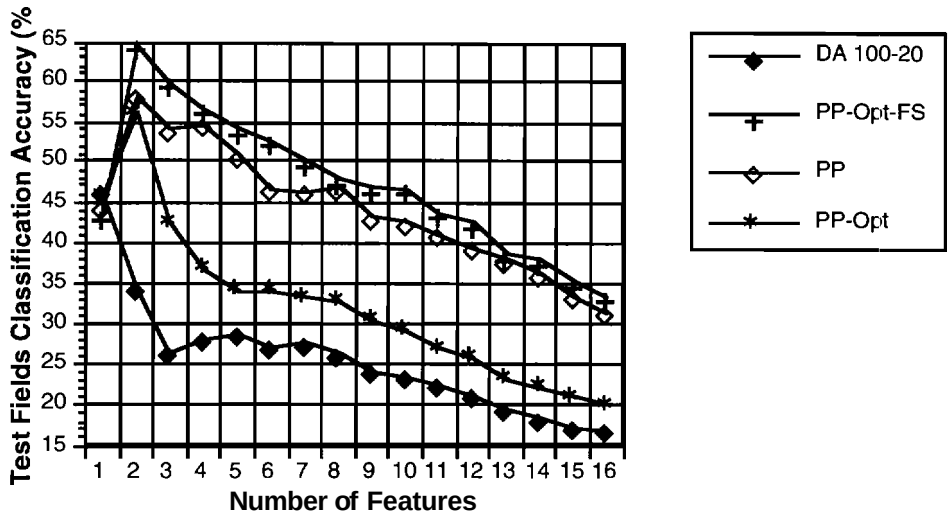


Fig. 4.15. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML with 2% threshold.

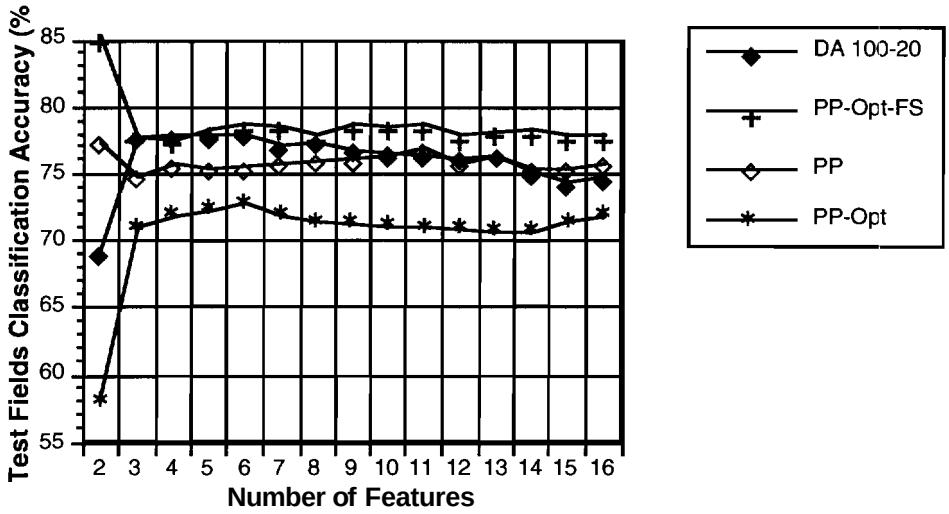


Fig. 4.16. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO classifier.

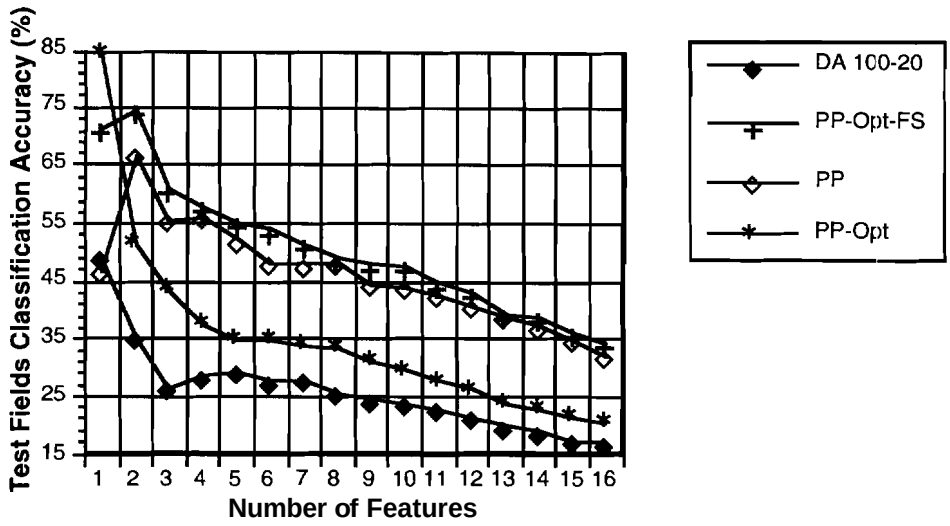


Fig. 4.17. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Decision Boundary after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO with 2% threshold.

Feature Selection

Feature selection could not be used in the 200 dimensional space Φ to project the data to the Ψ subspace. The reason is based on the fact that the number of calculations for feature selection in high dimensional space will be extremely high: $200!/((20!)(180!)) \approx 10^{27}$. Feature selection was applied, as previously done with Discriminant Analysis and Decision Boundary, after the use of Projection Pursuit based algorithms. The results in terms of classification accuracy, were compared with direct application of Discriminant Analysis (DA 100-20).

Here almost all Projection Pursuit based algorithms after **4** features had better results than Discriminant Analysis. The reason for that behavior is that most of the information in DA100-20 is in the first 3 features (number of classes -1). That is a limitation of Discriminant Analysis. Having such small number of labeled samples, whatever process that reaches a maximum first at a small number of features will dominate the Hughes Phenomena. It could be inferred in this case that it is probably that at lower dimensions, like three or four features, PP has a larger projection index than the other Projection Pursuit based algorithms.

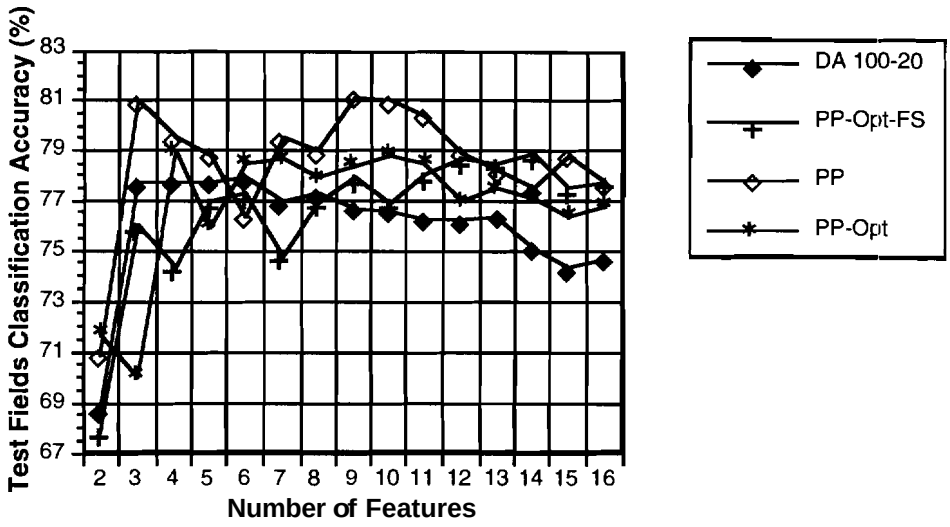


Fig. 4.18. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML classifier.

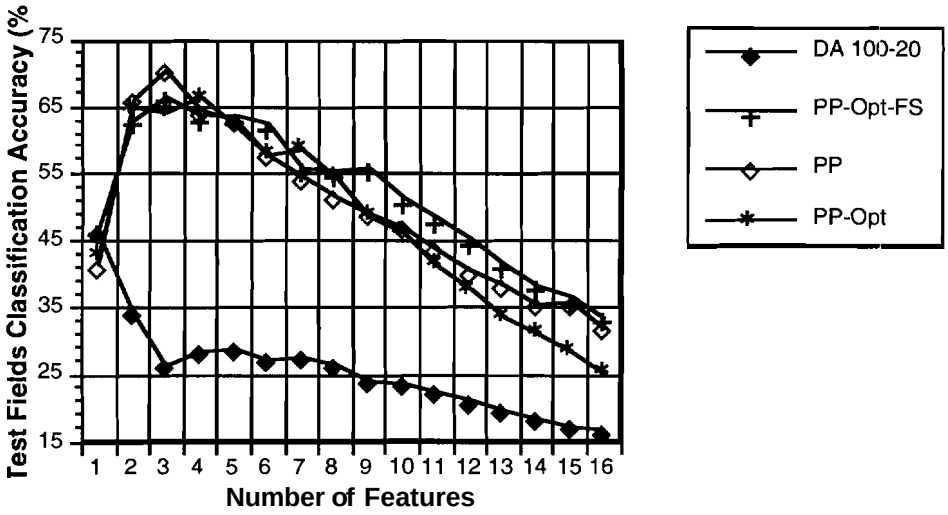


Fig. 4.19. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML with 2% threshold.

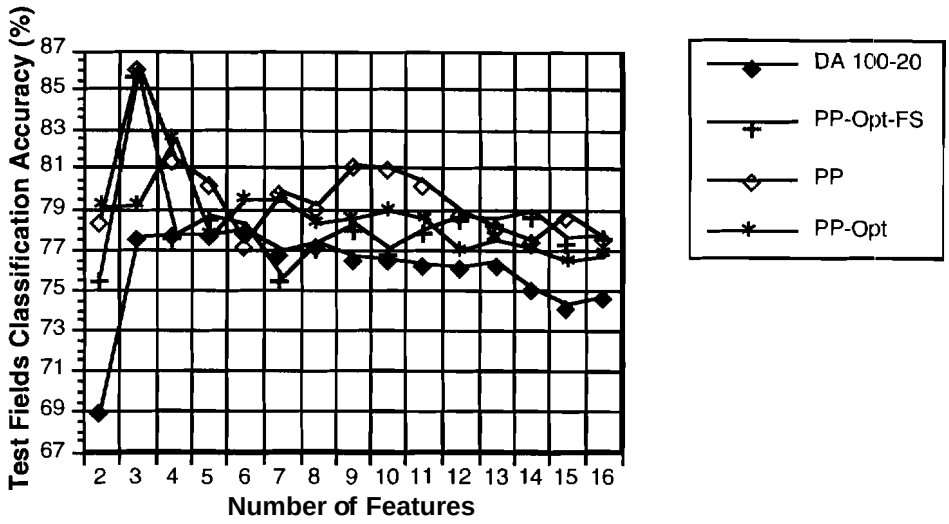


Fig. 4.20. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO classifier.

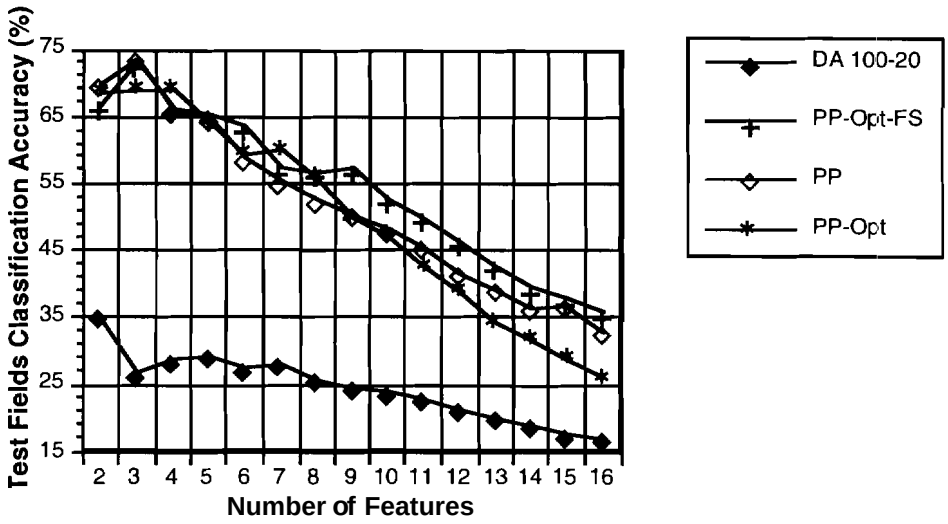


Fig. 4.21. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Feature Selection after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ECHO with 2% threshold.

4.6.3 Experiment 3

The multispectral data used in these experiments is a segment of AVIRIS data taken of NW Indiana's Indian Pine test site. From the original 220 spectral channels 200 were used, discarding the atmospheric absorption bands. In the present experiment, eight classes were defined. The total number of training samples is 1790 and the total number of test samples is 1630. Table 4.10 shows the defined classes and their respective number of training and test samples.

Table 4.10

Classes	Training Samples	Test Samples
Corn-min	229	232
Corn-notill	232	222
Soybean-notill	221	217
Soybean-min	236	262
Grass/Trees	227	216
Grass/Pasture	223	103
Woods	215	240
Hay-windrowed	207	138
Total	1790	1630

Four types of dimension reduction algorithms were used. The first is direct use of Decision Boundary Feature Extraction (DB 200-22) to reduce the dimensionality from 200 bands to 22 features. The second is direct use of Discriminant Analysis (DA 200-22) reducing the dimensionality again from 200 to 22. Both of these procedures perform a direct linear projection from Φ to Y. In the third and fourth methods Projection Pursuit and Projection Pursuit Feature Selection were used to reduce the dimensionality from 200 to 22. These methods linearly project the data from Φ to Γ subspace. After the preprocessing methods were used a feature extraction algorithm follows in order to project the data once more from Γ to Ψ subspace. Decision Boundary and Discriminant Analysis were used with the advantage of doing the computation with the same number of training samples in less number of dimensions.

Four types of classifiers were used: ML classifier, ML with 2% threshold, ECHO [45] [46] and ECHO with a 2% threshold. In the second and the fourth a threshold was applied to the standard classifiers whereby, in case of normal distributions of the class data, 2% of the least likely points will be thresholded. These 2% thresholds provide one indication of how well the data fit the normal model and how well the data is maintained in clusters. All of these classifiers performed a projection from Ψ to the resulted space Ω. All of these schemes of preprocessing, feature extraction, and data analysis are summarized in Table 4.11.

Table 4.11

Case	Preprocessing $\Phi \rightarrow \Gamma$	Feature Extraction $\Gamma \rightarrow \Psi$	Classifier $\Psi \rightarrow \Omega$
1	N/A	Direct use of Decision Boundary $\Phi \rightarrow \Psi$ (DB 200-22)	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%
2	N/A	Direct use of Discriminant Analysis $\Phi \rightarrow \Psi$ (DA 200-22)	(i) ML (ii) NIL-2% (iii) ECHO (iv) ECHO-2%
3	Projection Pursuit (PP)	(a) Decision Boundary (PPDBFE) (b) Discriminant Analysis (PPDAFE)	(i) ML (ii) NIL-2% (iii) ECHO (iv) ECHO-2%
4	Projection Pursuit Feature Selection (PPFS)	(a) Decision Boundary (PPFSDBFE) (b) Discriminant Analysis (PPFSDAFE)	(i) ML (ii) ML-2% (iii) ECHO (iv) ECHO-2%

Projection Pursuit

Table 4.12 shows the process of building a partition of groups of adjacent bands in order to build A for Projection Pursuit. The algorithm used is hybrid II with thresholds $\tau_{T-D} = .025$ and $\tau_{D-T} = .005$. Table 4.13 shows the minimum Bhattacharyya distance corresponding to each partition. The algorithm stops at 22 features, because it did not grow more than the threshold τ_{T-D}.

Table 4.12

Number of Features	**n**
1	[200]
2	[100 100]
3	[100 50 50]
4	[50 50 50 50]
5	[50 25 25 50 50]
6	[25 25 25 25 50 50]
7	[25 12 13 25 25 50 50]
8	[13 12 12 13 25 25 50 50]
9	[13 6 6 12 13 25 25 50 50]
10	[13 6 6 12 13 25 25 25 25 50]
11	[13 6 6 12 13 25 25 25 12 13 50]
12	[13 6 6 6 6 13 25 25 25 12 13 50]
13	[13 6 6 6 3 3 13 25 25 25 12 13 50]
14	[13 6 6 6 3 3 13 25 25 25 6 6 13 50]
15	[13 6 6 6 3 3 13 12 13 25 25 6 6 13 50]
16	[13 6 6 6 3 3 13 6 6 13 25 25 6 6 13 50]
17	[13 6 6 6 3 3 13 6 6 7 6 25 25 6 6 13 50]
18	[13 6 6 6 3 3 13 6 6 7 6 25 12 13 6 6 13 50]
19	[13 6 6 6 3 3 7 6 6 6 7 6 25 12 13 6 6 13 50]
20	[13 6 6 6 3 3 7 6 6 6 7 6 25 12 6 7 6 6 13 50]
21	[7 6 6 6 6 3 3 7 6 6 6 7 6 25 12 6 7 6 6 13 50]
22	[7 6 6 6 6 3 3 7 6 6 6 7 6 25 12 6 7 6 6 13 25 25]

Table 4.13

Number of Features	PP1 - Minimum Bhattacharyya Distance
1	.0158
2	.0684
3	.2730
4	.4416
5	.5783
6	.7035
7	.8950
8	.9947
9	1.1033
10	1.2690
11	1.3986
12	1.5594
13	1.6481
14	1.7704
15	1.8561
16	1.9477
17	1.9949
18	2.0598
19	2.1387
20	2.2000
21	2.2584
22	2.3190

Projection Pursuit Feature Selection

Table 4.14 shows the process of building a partition of group of adjacent bands in order to build the projection matrix **A.** Since there is no numerical optimization stage $\hat{\mathbf{A}} = \mathbf{A}$. The algorithm used is hybrid II with thresholds $\tau_{T-D} = .025$ and $\tau_{D-T} = .005$. Table 4.15 shows the minimum Bhattacharyya distance corresponding to each partition. Observe that the minimum Bhattacharyya distance for the **A** at each stage is less that with Projection Pursuit in table 4.13. That is expected since Projection Pursuit has in its banks of initial choices $\hat{\mathbf{a}}_i$'s the same vectors than Projection Pursuit Feature Selection in addition to others, as discussed in section 4.3.

Table 4.14

Number of Features	**n**
1	[200]
2	[100 100]
3	[100 50 50]
4	[50 50 50 50]
5	[50 25 25 50 50]
6	[25 25 25 25 50 50]
7	[25 13 12 25 25 50 50]
6	[25 13 12 50 50 50]
7	[25 7 6 12 50 50 50]
8	[25 4 3 6 12 50 50 50]
9	[25 4 3 6 12 50 25 25 50]
10	[25 4 3 6 12 25 25 25 25 50]
11	[25 4 3 6 12 25 25 25 12 13 50]
12	[25 4 3 6 12 12 13 25 25 12 13 50]
13	[25 4 3 6 12 12 13 25 25 12 13 25 25]
14	[25 4 3 6 12 12 13 25 25 6 6 13 25 25]
15	[25 4 3 6 12 12 13 25 25 3 3 6 13 25 25]
16	[25 4 3 6 12 12 13 25 25 3 3 6 13 13 12 25]
17	[25 4 3 3 3 12 12 13 25 25 3 3 6 13 13 12 25]
18	[25 4 3 3 3 12 12 13 13 12 25 3 3 6 13 13 12 25]
19	[25 4 3 3 3 12 6 6 13 13 12 25 3 3 6 13 13 12 25]
20	[25 4 3 3 1 2 12 6 6 13 13 12 25 3 3 6 13 13 12 25]
21	[25 4 3 3 1 1 1 12 6 6 13 13 12 25 3 3 6 13 13 12 25]
22	[25 4 3 3 1 1 1 12 6 3 3 13 13 12 25 3 3 6 13 13 12 25]

Table **4.15**

Number of Features	PP1 - Minimum Bhattacharyya Distance
1	.0147
2	.0741
3	.2645
4	.4056
5	.5069
6	.6202
7	.7331
6	.7483
7	.8241
8	.9272
9	**1.0058**
10	**1.0697**
11	**1.2144**
12	**1.2829**
13	**1.3435**
14	**1.4214**
15	**1.4749**
16	**1.5246**
17	**1.6135**
18	**1.6751**
19	**1.7392**
20	**1.8145**
21	**1.8728**
22	1.9020

Figure 4.22 shows how minimum Bhattacharyya distance in Deciision Boundary, Discriminant Analysis and the first stage of Projection Pursuit increases as the number of features increases. Observe that the first stage of Projection Pursuit (PP1) is the maximum at almost every value. Discriminant Analysis increases fast from 6 to 7 features. This is well explained by the fact that the first seven features (number of classes - 1) are estimated from the Fisher criterion, meanwhile the rest of the features are chosen randomly. Decision Boundary performs the poorest in the first fifteen features. From 16 to 20 is in the middle of Projection Pursuit first stage and Discriminant Analysis and at 22 features it becomes the best. At that number PP1 stop to increase significantly. Projection Pursuit Feature Selection (PPFS) performs closely to PP1 in the first number of features. As expected PP1 is an upper bound of PPFS. As the nurnber of used features increased, the differences between both methods increases as well. Still there is a range where PPFS is the second best option, better than direct application of feature extraction methods. The results suggest that this method is a good one to use in case of having a large separation among classes where the number of features required is small.

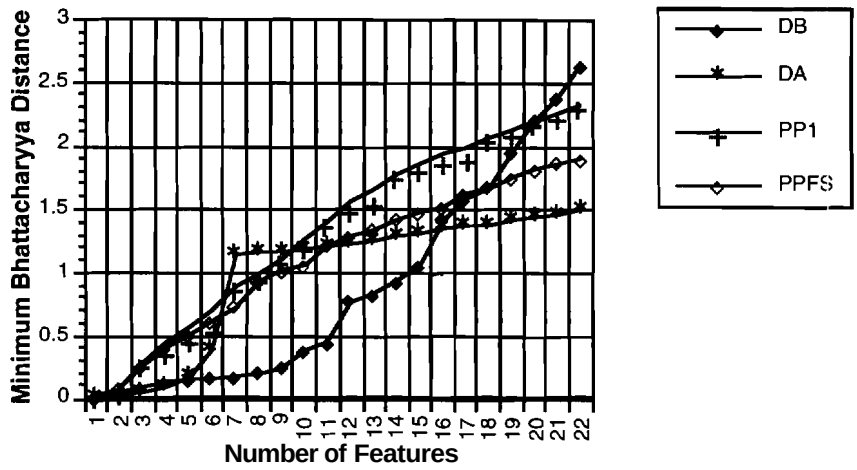


Fig. 4.22. Minimum Bhattacharyya distance.

Figure 4.23 shows for each method the percentage of growth of their respective different projection indices. For PP1 and PPFS the minimum Bhattacharyya distance is shown, for Discriminant Analysis it is the cumulative value of the Fisher criterion eigenvalues, and for Decision Boundary it is the cumulative value of the eigenvalues of a Decision Boundary Feature matrix. Observe that Discriminant Analysis stops to increase significantly in terms of its percentage of grow, at *7* features. Decision Boundary, PP1 and PPFS stop to increase significantly at around 20 to 22 features. This implies an agreement of these last three methods of what is the dimensionality of the training data.

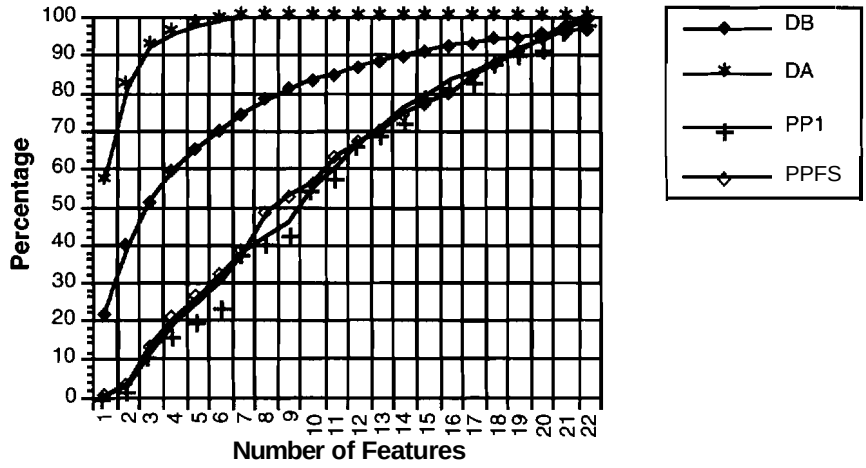


Fig. 4.23. Percentage of grow of the different methods.

In terms of their respective projection indices Figure 4.23 shows that Discriminant Analysis could not extract more information after *7* dimensions, Decision Boundary

after around 18 and PP1 and PPFS after 22. As a consequence no matter that Decision Boundary's minimum Bhattacharyya distance is larger after 21 features than PP1 and PPFS, the analyst would choose as a final number of dimensions, a number around 18 dimensions. These results show that the first stage of Projection Pursuit and Projection Pursuit Feature Selection are good estimators of the dimensionality of the space Γ. For Projection Pursuit a second stage numerical optimization method was performed, and its minimum Bhattacharyya distance was measured (PP2). The results of the minimum Bhattacharyya distances for Decision Boundary, Discriminant Analysis, PP1, PP2 and PPFS are shown in table 4.16 for Γ in 22 dimensions.

Table 4.16

Method	DB	DA	PP1	PP2	PPFS
Min. Bhatt. Dist.	2.64	1.52	2.32	2.75	1.90

With the numerical optimization stage, Projection Pursuit was able to have a larger projection index than the other methods. The next sections will apply the feature extractiori techniques after the use of Projection Pursuit' based 'algorithms and compare their results with direct application of Decision Boundary and Discriminant Analysis in Φ.

Feature Extraction Methods

Decision Boundary Feature Extraction

This part of the experiments has the objective of testing how Projection Pursuit based algorithms enhances test fields classification accuracy in the use of Decision Boundary at 22 dimensions in Γ in comparison with direct use of Decision Boundary at full dimensionality in Φ space. Figures 4.24, 4.25, 4.26, and 4.27 show the results for ML classifications. In terms of training fields, Projection Pursuit (PPDBFE) and Projection Pursuit Feature Selection (PPFSDBFE) increase in classification accuracy faster than direct use of Decision Boundary (DBFE). As expected in a significant range PPFSDBFE results are in between PPDBFE and DBFE. At 22 dimensions PPDBFE and DBFE are close and both of them are superior than PPFSDBFE in accordance with the values of the minimum Bhattacharyya distance at 22 dimensions as shown in table 4.16. In terms of test fields classification accuracy PPDBFE performs better with a difference from 25% to 30% with respect to DBFE. PPFSDBFE results are closer to

PPDBFE than DBFE. Observe in Figures 4.26 and 4.27 that PPDBFE and PPFSDBFE maintains the data more in clusters, and at the same time the assumption of normality is better supported. At 22 features there is a difference of 65% between Projection Pursuit based algorithms and direct application of Decision Boundary in the test fields classification accuracy with the use of a 2% threshold.

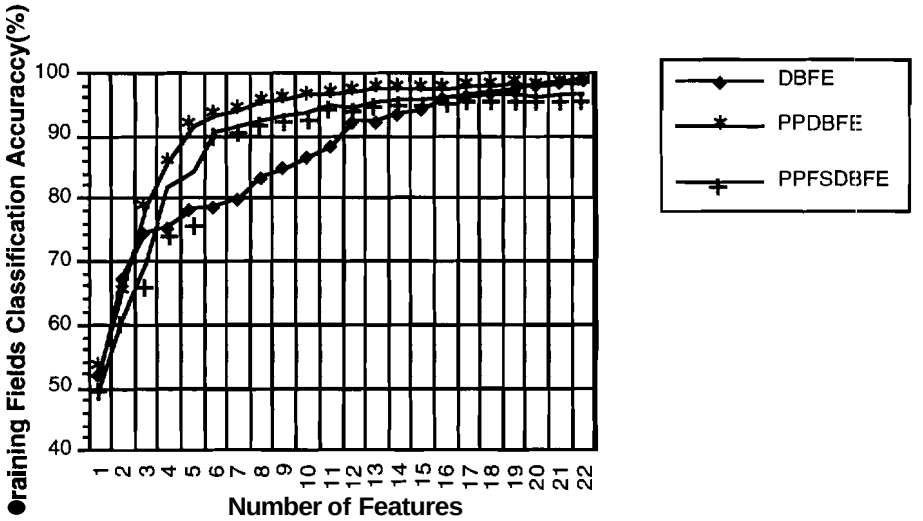


Fig. 4.24. Training fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML classifier.

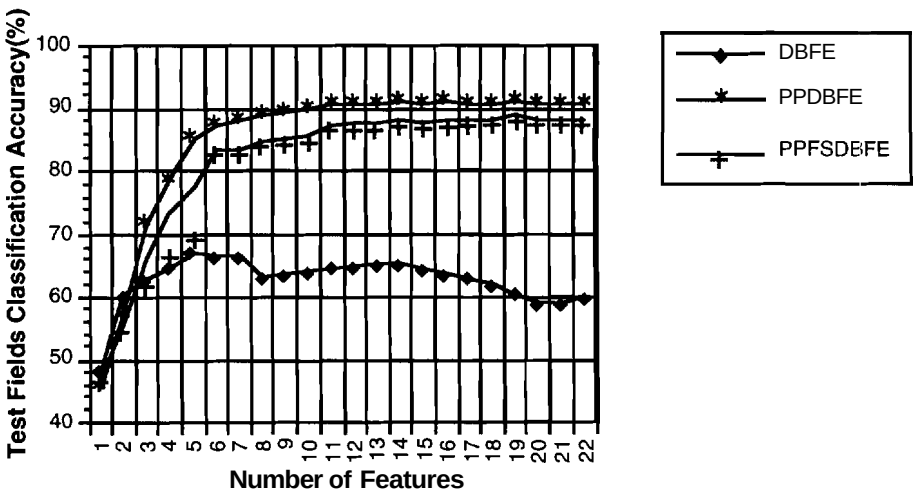


Fig. 4.25. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML classifier.

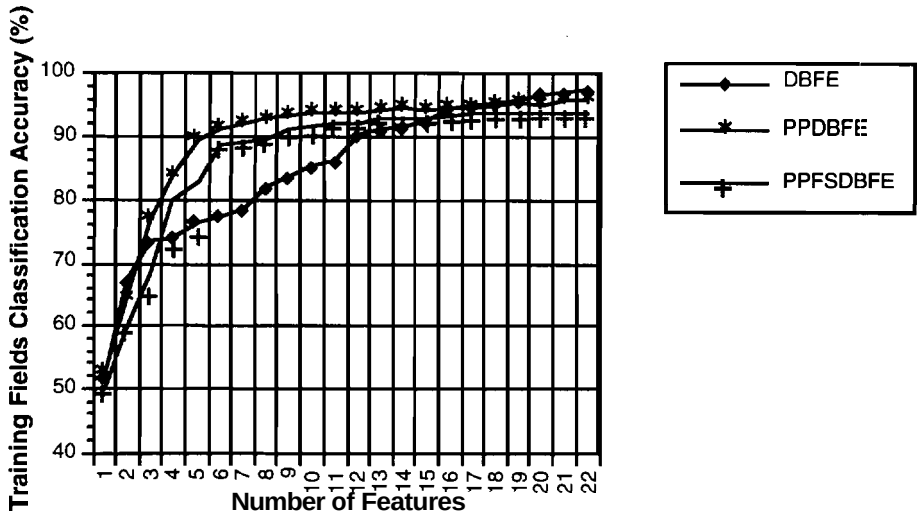


Fig. 4.26. Training fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for **ML** with 2% threshold.

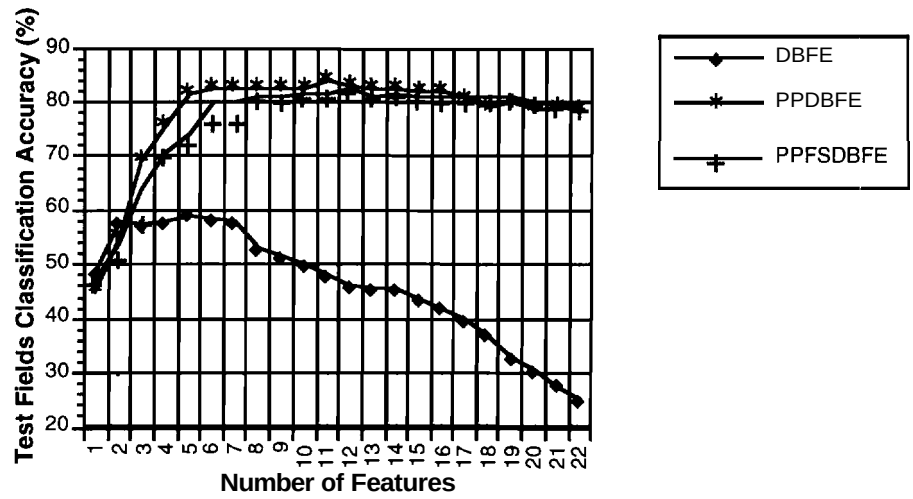


Fig. 4.27. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for **ML** with 2% threshold.

Figure 4.28, 4.29, 4.30, and 4.31 show the results for the ECHO classifier. The values of PPFSDBFE is closer to PPDBFE than in the ML's results. 'The differences between both of the Projection Pursuit's methods and direct use of Decision Boundary increases. In this case it goes from 15% up to 35% at 22 features. Note with the ECHO classifier, PPDBFE and PPFSDBFE arrive at their maximum (95%) and stay there, meanwhile for DBFE, the Hughes Phenomena start to play its role after 7 features. With the use of a threshold there is a greater difference at 22 features between

Projection Pursuit's based procedures and direct use of Decision Boundary than with ML at 22 features.

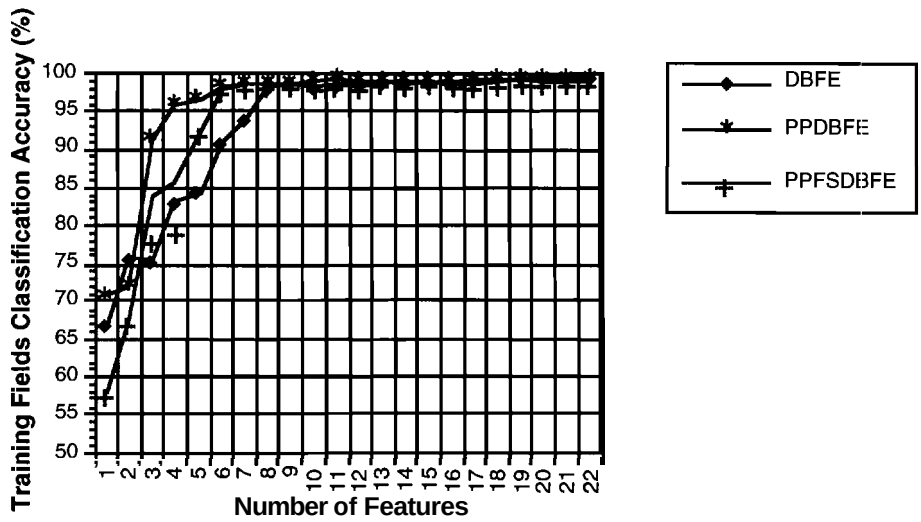


Fig. 4.28. Training fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ECHO classifier.

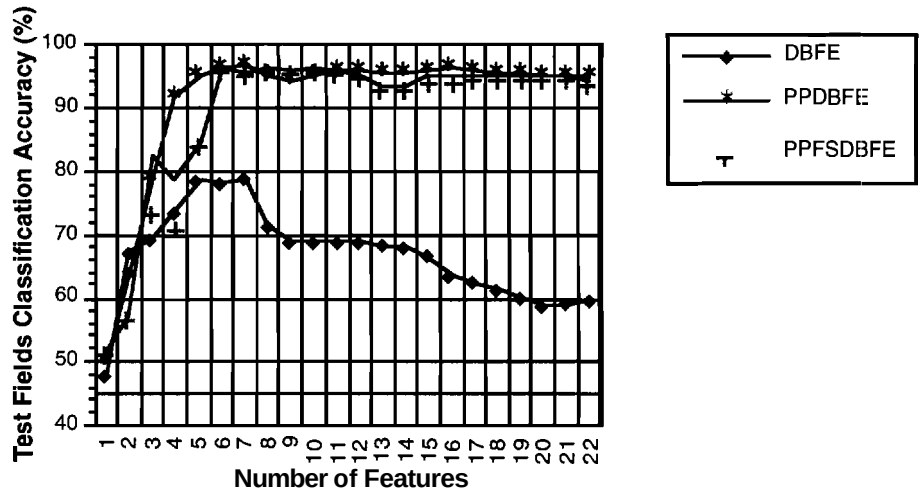


Fig. 4.29. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ECHO classifier.

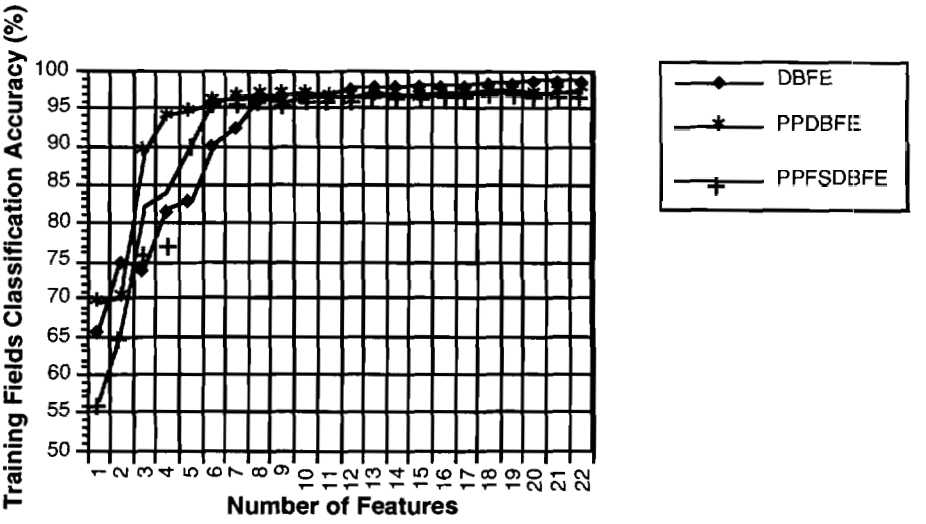


Fig. 4.30. Training fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ECHO with 2% threshold.

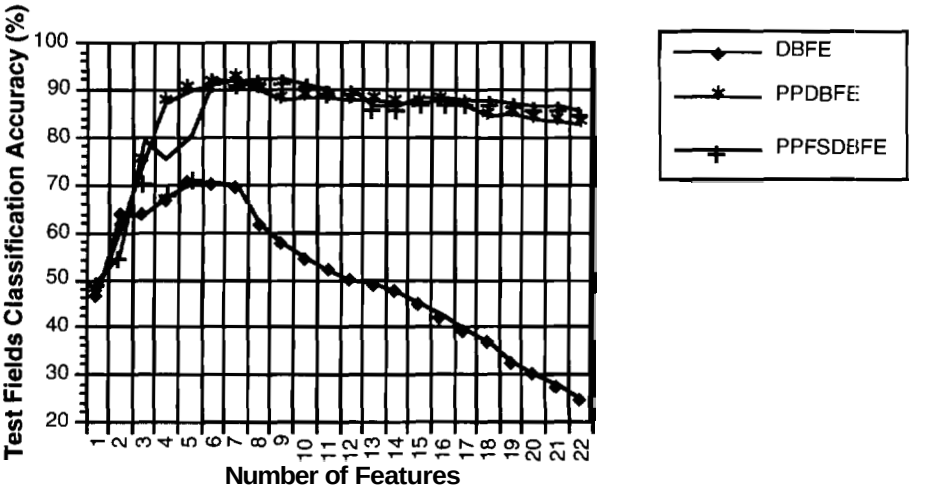


Fig. 4.31. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ECHO with 2% threshold.

Discriminant Analysis

In this experiment three procedures were used to project tha data to a 22 dimensional subspace. The first one was direct application of Discrirninant Analysis (DAFE) on the 200 dimensions at the Φ space. The second procedure used was Projection Pursuit to project the data from Φ to Γ. The third used is Projection Pursuit Feature Selection to project the data from Φ to Γ. After Projection Pursuit's based algorithms were used Discriminant Analysis was applied in the Γ subspace in order to

compare the test fields classification results (PPDAFE and PPFSDAFE) with direct use of Discriminant Analysis (DAFE).

Figure 4.32, 4.33, 4.34 and 4.35 show the results with the ML classifier. In terms of the training fields, the classification results are very similar. In the test fields Projection Pursuit's algorithms performs better. The difference there is significant. It is not as dramatic as in Decision Boundary because this last method of feature extraction requires rnore training samples per feature than Discriminant Analysis. Note in Figure 4.33 that PPDAFE and PPFSDAFE are able to grow after 7 features. Tliis is due to the fact that tlie minimum Bhattacharyya distance, which is a bound of Bayes classification accuracy, is maximized for the entire Γ subspace. Independent of the fact that for K classes Discriminant Analysis only calculates K-1 independent features that maximize the Fisher criterion, the addition of more features of the Γ subspace will contribute more to the separation of classes. As expected PPDAFE has the best performance and reaches an accuracy above 90%. Meanwhile DAFE stop to grow after 7 features and stays at 85% accuracy. With the use of the 2% threshold the ML's results of test fields classification accuracy of Projection Pursuit's procedures are better than direct use of Discriminant Analysis. This is due to the fact that the assumption of normality is better supported with the Projection Pursuit' algorithms.

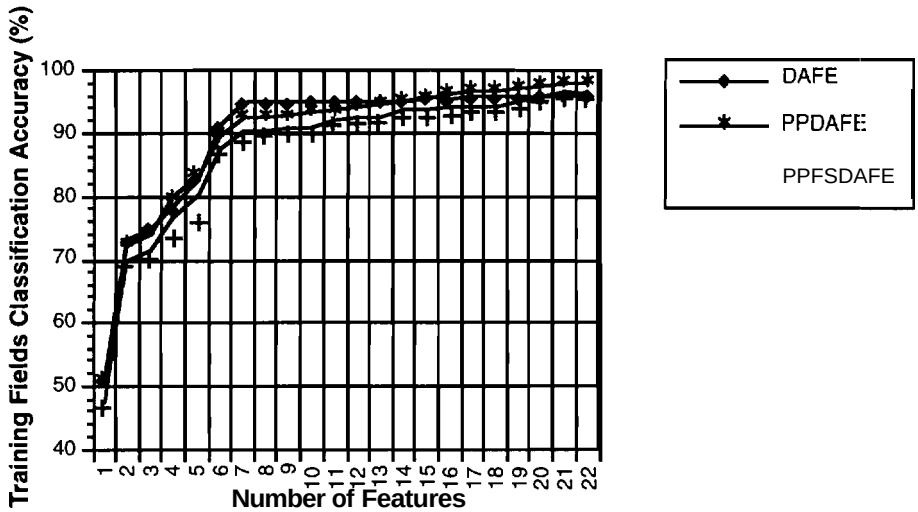


Fig. 4.32. Training fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML classifier.

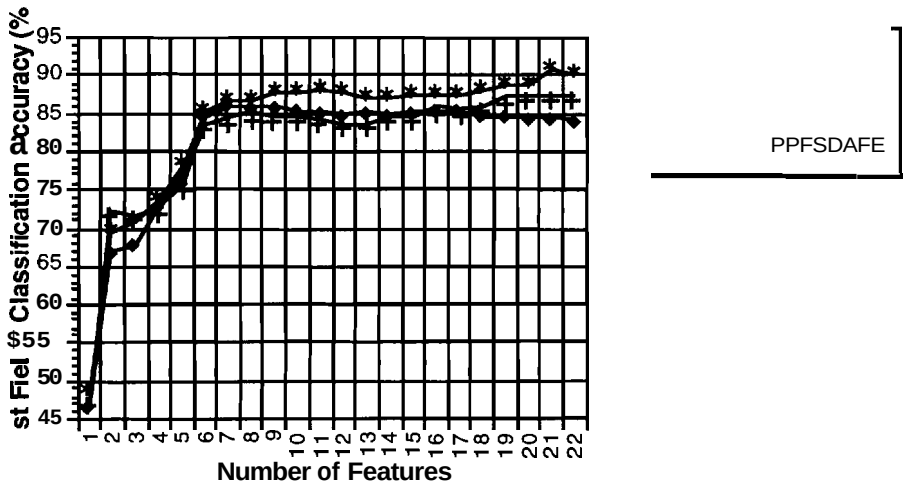


Fig. **4.33.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML classifier.

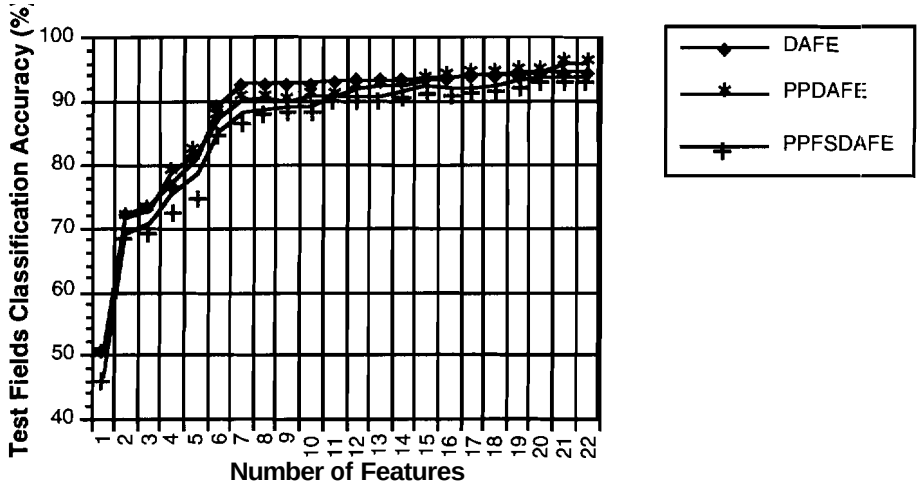


Fig. **4.34.** Training fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML with 2% threshold.

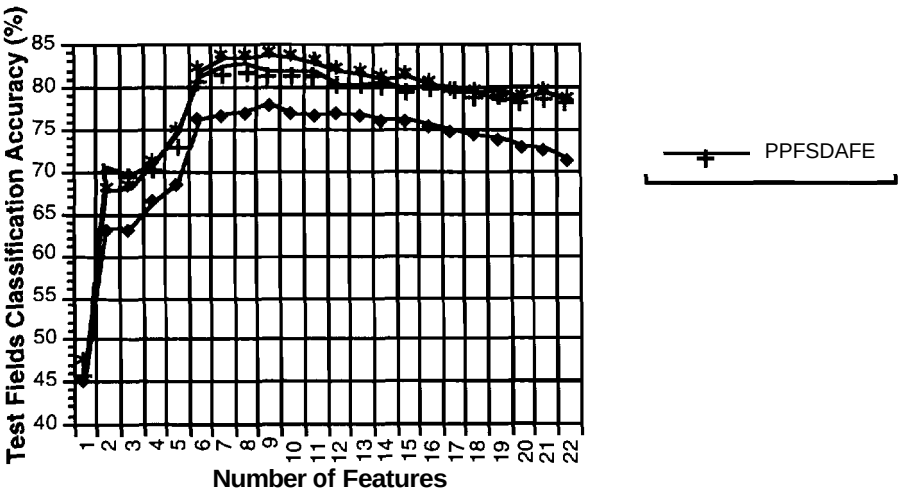


Fig. **4.35.** Test fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML with 2% threshold.

The ECHO classification confirms the ML results. Projection Pursuit algorithms enable Discriminant Analysis to arrive at the maximum and maintain the data more in clusters. All of this is based on the event that Projection Pursuit deals better with the Hughes Phenomena, high dimensional space characteristics and the assumption of normality

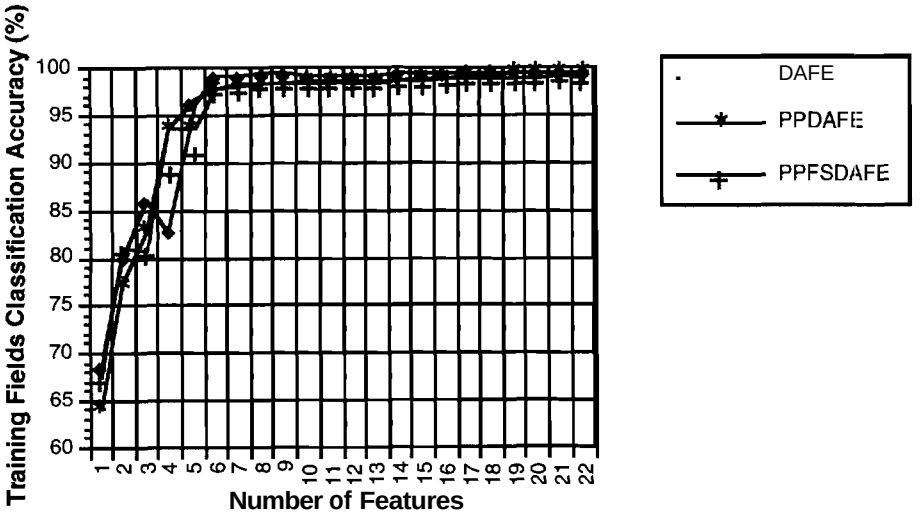


Fig. **4.36.** Training fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ECHO classifier.

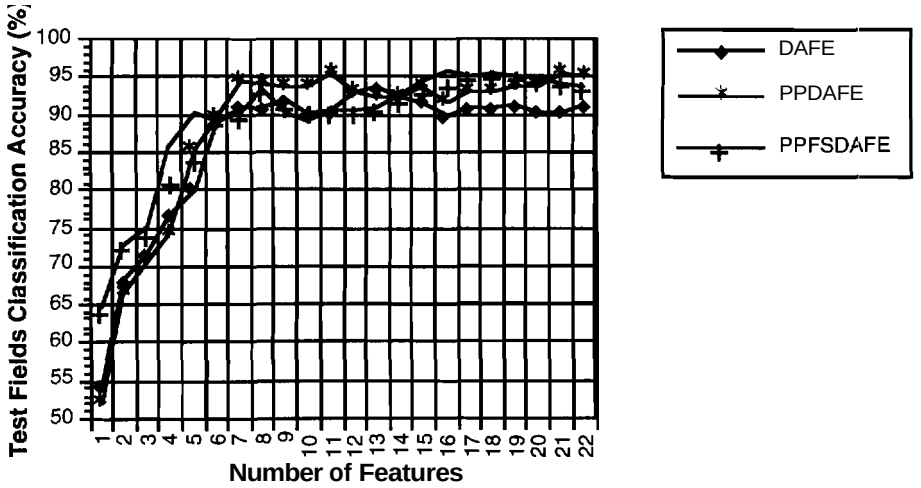


Fig. 4.37. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ECHO classifier.

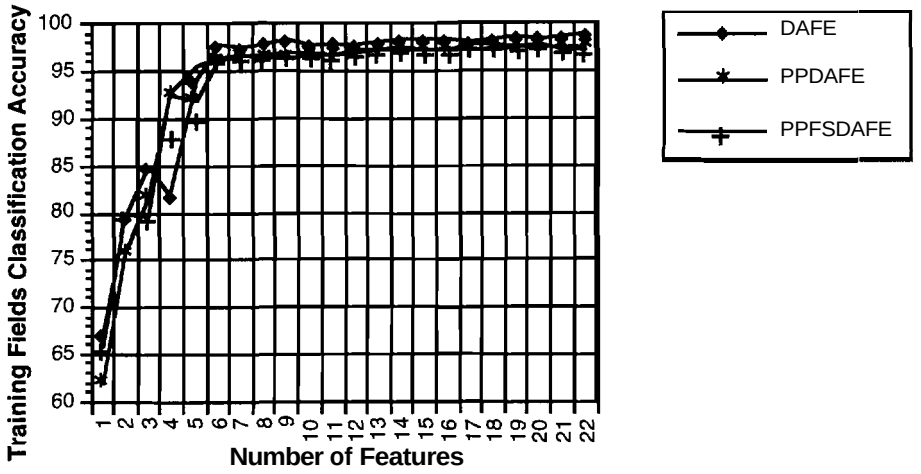


Fig. 4.38. Training fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ECHO with 2% threshold.

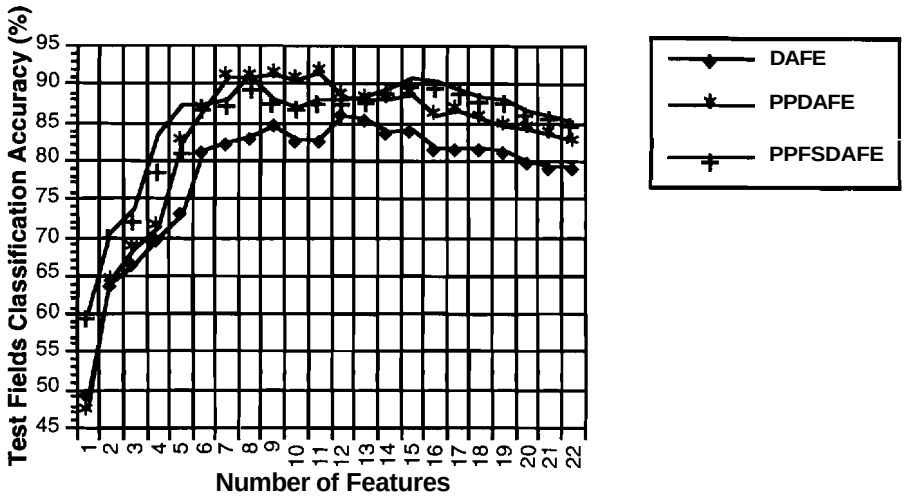


Fig. 4.39. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ECHO with 2% threshold.

4.7 Conclusion

In this chapter two Projection Pursuit based algorithms have been proposed to preprocess the data before a feature extraction and classification algorithms are applied. They are regular Projection Pursuit and Projection Pursuit Feature Selection. The minimum Bhattacharyya distance among the classes was used as the projection index to maximize in the parametric version of Projection Pursuit. The purpose of these algorithms is to overcome the problem of training the classifier with a small number of labeled samples in a high dimensional space with its inherent characteristics.

A first stage of preprocessing has been proposed in order to estimate an a priori matrix A for the numerical optimization process that Projection Pursuit requires. The first stage preprocessing algorithm was based on binary tree techniques. Its purpose is to avoid arriving at a non-optimal maximum, and it helps preserve information from the high dimensional space.

The technique developed for the first stage pre-processing enables also the development of a Projection Pursuit feature selection algorithm for high dimensional data where it overcomes the problem of large numbers of computations. Both of these techniques also estimate the dimensionality of the projected subspace.

The experiments performed in this chapter show that Projection Pursuit enables feature extraction algorithm to extract more information from the training samples. That is shown in the enhancement of their training and test fields classification accuracy in

the ML and ECHO classifiers. This is the case for small or relative large number of training samples and classes.

This is due to the fact that Projection Pursuit fulfills the properties that a high dimensional reduction algorithm should have as explained in chapter 2. It eludes the difficulties of high dimensional data by making the computations at a lower dimensionality of the projected subspace, enabling the feature extraction algorithms to have more accurate estimations of the statistical parameters. At that feature subspace the assumption of normality is better supported, permitting the classifier to have better results in terms of classification accuracy.

5. SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

5.1 Summary

The present research is related with the problem that the optirnum number of features for feature extraction and classification purposes in supervised classification techniques is limited by the number of training samples. That condition has restricted severely the practical applications of statistical pattern recognition procedures in high dimensional data. There is a need to reduce the dimensionality in a different way than using feature extraction techniques in order to avoid the problem sonietimes referred to as the curse of dimensionality.

Chapter 2 studied the characteristics and properties of high dimensional space. It was suggested that use of a preprocessing step before the application of feature extractions methods and classification techniques, as shown in Figure 2.18 would be beneficial. That suggestion was based on some conclusions that came out of the study. One conclusion was that, because of problems with nonparametric schemes, a new parametric method was needed which performs the computation at a lower dimensional space instead at full dimensionality. Performing the computation in a lower dimensional subspace that is a result of a linear projection from the original high dimensional space will make the assumption of normality better supported, giving a better parameter estimation, and better classification accuracy. Another important statement derived from the study is the need of taking into consideration first and second order statistics for measuring the distance among classes, as is done with Bhattacharyya distance.

Chapter 3 developed a preprocessing method taking into consideration the characteristics studied in chapter 2. A modified schema of supervised classification was proposed. Such modification is the result of the addition of a preprocessing algorithm with the purpose of reducing the dimensionality of the data projecting it to a subspace where feature extraction or feature selection are more suitable. Projection Pursuit was the method used to develop the algorithms for accomplishing such

preprocessing. A parametric version was developed and used based on the use of a projection index that uses a priori information such as labeled samples. Parametric Projectiori Pursuit fulfills the criteria established in chapter 2 for a preprocessing method using the minimum Bhattacharyya distance as the projection index to be minimized. This procedure, performing the computations at a lower dimensional subspace, makes the assumption of normality better supported with better estimations of parameters and features. All of this enables the algorithm to deal better with the Hughes phenomena, better maintaining the data in clusters, and resulting in better classification accuracy.

Based on that concept, two approaches were developed, Parallel and Sequential Parametric Projection Pursuit. The Parallel approach has the advantage of being faster, but it does not guaranteed that it will perform better in terms of the optimization of the overall projection index. The Sequential approach had the disadvantage of being slow if it is directly implemented. Such disadvantage could be overcome in a great extend with an iterative version. The advantage that Sequential Projection Pursuit has to offer is a direct control of the projection index over the projected subspace. The optimization of the global projection index allows more control and better performance against the problem of local maxima than local optimization in the Parallel approach. Still there was a need to compute an initial choice matrix A for the global optimization process.

In chapter 4 a first stage of preprocessing was proposed in order to estimate an a priori matrix A for the numerical optimization process that Projection Pursuit requires. The first stage preprocessing algorithm was based on binary decision tree techniques. Its purpose is to avoid arriving at a non-optimal local maximum, and thus helping preserve more information from the high dimensional space. The technique developed for the first stage preprocessing enables also the development of a Projection Pursuit Feature Selection algorithm for high dimensional data that overcomes the problem of large numbers of computations. Both of these techniques also estimate the dimensionality of the projected subspace. The empirical results of training and test fields classification accuracy were better than direct use of feature extraction procedures at high dimensional space. This is due to the fact that Projection Pursuit fulfills the requirements that a high dimensional reduction algorithm should have, as explained in chapter 2. It eludes the difficulties of high dimensional data by making the computaitions at a lower dimensionality of the projected subspace, enabling the feature extraction algorithms to have more accurate estimations of the statistical parameters.

5.2 Suggestion for Further Work

1. The exploration of Projection Pursuit's application in other areas of Statistical Pattern Recognition is highly encouraged. Among those areas is unsupervised learning, i.e. clustering. Most of the known clustering algorithms have problems in high dimensional space. It will be useful to design a scheme based on Projection Pursuit that performs the computations at a lower dimensional space. That will enable the clustering algorithm to extract more information about detailed classes from high dimensional data

2. Another possible area of Projection Pursuit's application could be classification. The present classifiers estimate the parameters at full dimensionality. It will be important for analyzing high dimensional data to develop new classifiers based on well recognized theories and Projection Pursuit, i.e. doing the computation of the parameters at a lower dimensional space.

3. In the present work a Parametric Projection Pursuit algorithm had been proposed in order to accomplish the objectives of a preprocessing method. A specific constraint to the matrix A was assumed and that resulted in the Parallel and Sequential Projection Pursuit approaches. Both of them, assuming that adjacent features are highly correlated, combines groups of adjacent bands into one feature. Other types of constrictions could be explored. This could result in different lower dimensional computations for Parametric Projection Pursuit. The only requisite is that independently of what constraints are imposed on A, its rows should be linearly independent.

4. In terms of the present research, it is suggested that there is a need for research on different projection indices. In terms of feature extraction and classification purposes, there is a need for parametric indices. Unsupervised classification requires a further development of nonparametric indices. It is suspected that different feature extraction algorithms, classifiers and clustering schemes will need different projection indices. There are other applications of remote sensing that could receive the benefits of Projection Pursuit and the development of a projection indices that imply what is the interesting characteristic of the data that is required to be maximized.

5. An empirical study is needed in order to estimate the optimum values of the thresholds $\tau_{T\text{-}D}$ and $\tau_{D\text{-}T}$. These values are required in order to make a comparison with equations **(4.7)** and **(4.8).** The values of the n_i's and the final number of dimensions are sensitive to these variables.

6. Parametric Projection Pursuit performs the computations at a lower dimensional space. It requires the use of a numerical optimization algorithm. A study of different numerical optimization methods will be useful for its application in high dimensional data. Because of the high dimensionality characteristic of the data, the number of local maxima could be high. The characteristic of being robust to the problem of local maxima should be the most relevant to be consider in the algorithm.

LIST OF REFERENCES

[1] J. A. Richards, Remote Sensing Digital *Image* Analysis, An Introduction. 2nd ed. New York: Springler-Verlag, 1993, pp. 1-6.

[2] P. H. Swain and S. M. Davis, eds., Remote Sensing: *The* Quantitative Approach. New-York: McGraw-Hill, 1978, pp. 5.

[3] D. A. Landgrebe, "On the use of Stochastic Process-Based Methods for the Analysis of Hyperspectral Data," in 1992 IEEE Proceedings of *the* International *Geoscience* and Remote Sensing Symposium, Houston, Texas, May 26-29, 1992, pp. 552-554.

[4] R. O. Duda and P. E. Hart, Pattern Classification and *Scene* Analysis. New York: John Wiley & Sons, 1973, pp. 1-5.

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

[6] A. K. Jain and W. G. Waller, "On the Optimal Number of Features in the Classification of Multivariate Gaussian Data," Pattern *Recognition*, Vol. 10, pp. 365-374, 1978.

[7] K. Fukunaga, "Effects of Sample Size in Classifier Design," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 8, pp. 873-885, 1989.

[8] D. W. Scott, Multivariate Density Estimation. New York: John Wiley & Sons, 1992, pp. 208-212.

[9] J. Hwang, S. Lay and A. Lippman, "Nonparametric Multivariate Density Estimation: A Comparative Study," IEEE Transactions on *Signal* Processing, Vol. 42, No. 10, pp. 2795-2810, 1994.

[10] C. Lee and D. A. Landgrebe, "Feature Extraction Based on Decision Boundaries," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 4, pp. 388-400, April 1993.

[11] C. Lee and D. A. Landgrebe, "Feature Extraction and Classification Algorithms for High Dimensional Data," School of Electrical Engineering Purdue University, Technical Report, TR-EE 93-1, 1993.

[12] M. G. Kendall, A Course in the Geometry of n-dimensions, New York: Hafner Publishing Co., 1961, pp. 35.

[13] E. J. Wegman, "Hyperdimensional Data Analysis Using Parallel Coordinates," *Journal* of the American Statistical Association, Vol. 85, No. 411, pp. 664-675, 1990.

[14] L. L. Scharf, Statistical Signal Processing. Detection, Estimation, and Time Series Analysis. Massachusetts: Addison-Wesley, 1991, pp. 62-64.

[15] C. Lee and D. A. Landgrebe, "Analyzing High Dimensional Multispectral Data," *IEEE* Transactions on Geoscience and Remote Sensing, Vol. 31, No. 4, pp. 792-800, July, 1993.

[16] P. Diaconis and D. Freedman, "Asymptotics of Graphical Projection Pursuit," The Annals of Statistics ,Vol. 12, No. 3, pp. 793-815, 1984.

[17] P. Hall and K. Li, "On Almost Linearity Of Low Dimensional Projections From High Dimensional Data," The Annals of Statistics, Vol. 21, No. 2, pp. 867-889, 1993.

[18] K. Fukunaga, Introduction to Statistical Pattern *Recognition*. San Diego, California: Academic Press, Inc., 1990, pp. 24-34.

[19] L. Jimenez and D. A. Landgrebe, "Projection Pursuit For High Dimensional Feature Reduction: Parallel And Sequential Approaches," presented at the International Geoscience and Remote Sensing *Symposium (IGARSS'95)*, Florence Italy, July 10-14, 1995.

[20] L. Jimenez and D. A. Landgrebe, "Projection Pursuit in High Dimensional Data Reduction: Initial Conditions, Feature Selection and the Assumption of Normality," presented at the IEEE *International* Conference on Systems, Man and Cybernetics, Vancouver Canada, October 1995.

[21] A. G. Wacker, Minimum Distance Approach to Classification, PhD Thesis, School of Electrical Engineering, Purdue University, May 1972.

[22] R. Solberg and T. Egeland, "Automatic Feature Selection In Hyperspectral Satellite Imagery," 1993 IEEE Proceedings of the International Geoscience and *Remote* Sensing Symposium, Tokyo, Japan, August 18-21, 1993, pp. 472-474.

[23] T. M. Cover and J. M. V. Campenhout, "On the Possible Ordering in the Measurement Selection Problem," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC 7, No. 9, pp. 657-661, 1977.

[24] T. Okada, and S. Tomita, "An optimal orthonormal system for discriminant analysis," Pattern Recognition, Vol. 18, pp. 139-144, 1985.

[25] Y. Hamamoto, Y. Matsuura, T. Kanaoka and S. Tomita, "A note on the orthonormal discriminant vector method for feature extraction," Pattern Recognition, Vol. 24, No. 7, pp. 681-684, 1991.

[26] Y. Hamamoto, T. Kanaoka and S. Tomita, "On a theoretical comparison between the orthonormal discriminant vector method and discriminant analysis," Pattern recognition, Vol. 26, No. 12, pp. 1863-1867, 1993.

[27] M. Aladjem, "Parametric and Nonparametric Linear Mappings of Multidimensional Data.", Pattern Recognition, Vol. 24, No. 6, pp. 543-553, 1991.

[28] S. Kiyasu and S. Fujimora, "Successive Feature Extraction Frorn Hyperspectral Data," in *1993* IEEE Proceedings of the International *Geoscience* and Remote Sensing Symposium, Tokyo, Japan, August 18-21, 1993, pp. 469-471.

[29] A. Biem and S. Katagiri, "Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Method," *1993* IEEE *International* Conference on Acoustics, Speech and Signal Processing, Minneapolis, MN, USA, Apr 27-30. 1993, Vol. 2, pp. 275-278.

[30] J. H. Friedman and J. W. Tukey, "A projection algorithm for exploratory data analysis," IEEE Trans. Comput., C-23, pp. 881-889, 1974.

[31] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," Journal of the American *Statistics* Association, Vol. 76, pp 817-823, 1981.

[32] P. Hall, "On Projection Pursuit Regression," The Annals of *Statistics*, Vol. 17, No. 2, pp. 573-588, 1989.

[33] J. H. Friedman, W. Stuetzle and A. Schroeder, "Projection Pursuit Density Estimation," Journal of the American Statistics Association, Vol. '79, pp. 599-608, 1984.

[34] P. J. Huber, "Projection Pursuit," The Annals of Statistics , Vol. 13 No. 2, pp. 435-475, 1985.

[35] K. Sutherland and W. Thompson, "Pursuit Projections: Keeping a Robot on Path," Proceedings of the *1994* IEEE International Conference on Robotics and Automation, San Diego, California, May 8-13, 1994.

[36] C. M. Bachmann, S. A. Musman, D. Luong and A. Schultz, "Unsupervised BCM Projection Pursuit Algorithms for Classification of Simulated Radar Presentations," Neural Networks, Vol. 7 No. 4, pp. 709-728, 1994.

[37] N. Intrator and L. N. Cooper, "Objective Function Formulation of the BCM Theory of Visual Cortical Plasticity: Statistical Connections, Stability Conditions," Neural Networks, Vol. 5, pp. 3-17, 1992.

[38] L. K. Jones, "Good Weights and Hyperbolic Kernels for Neural Networks, Projection Pursuit, and Pattern Classification: Fourier Strategies for Extracting Information from High Dimensional Data," IEEE Transactions on *Information* Theory, Vol. 40, No. 2, pp. 439-454, 1994.

[39] G. P. Nason, "Viewing multispectral images with projection pursuit," School of Mathematical Science, University of Bath, Bath, UK, Statistics Research Report 93:02, 1993.

[40] M. C. Jones and R. Sibson, "What is Projection Pursuit," *J.* R. Statistics *Soc.* A Part ▮ „pp. 1- 36, 1987.

[41] R. E. Blahut, Principles and Practice of *Information* Theory. Massachusetts: Addison-Wesley Publishing Company, 1987, pp. 246.

[42] P. Hall, "On Polynomial-Based Projection Indices for Exploratory Projection Pursuit," The Annals of Statistics, Vol. 17, No. 2, pp. 589-605, 1989.

[43] L. Bo, S. Zhenkang and S. Zhongkang, "A Pattern Recognition Method Using Projection Pursuit," Proceedings of the 1990 IEEE National Aerospace and Electronics Conference, Dayton, Ohio 1990, pp. 300-302.

[44] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C. Second Edition, Cambridge: Cambridge University Press, 1992, pp. 396.

[45] R. L. Kettig and D. A. Landgrebe, "Computer Classification of Remotely Sensed Multispectral Image Data by Extraction and Classification of Homogeneous Objects," IEEE Transactions on Geoscience Electronics, Volume GE-14, No. 1, pp. 19-26, January 1976.

[46] D. A. Landgrebe, "The Development of a Spectral-Spatial Classifier for Earth Observational Data," Pattern Recognition, Vol. 12, No. 3, pp. 165-175, 1980.

[47] P. Hall, "Estimating the direction in which a data set is most interesting," *Probability* Theory and Related Fields, Vol. 88, pp. 51-77, 1988.

[48] J. R. Quinlan, "Induction of Decision Trees," in Machine Learning, J. W. Shavlik and T. G. Dietterich, Eds., Boston: Kluwer Academic Publisher, 1986, pp. 81-106.

[49] S.M. Weiss and C.A. Kulikowski, Computer Systems that Learn, San Mateo California: Morgan Kaufmann Publishers, Inc., 1991, pp. 116-118.

[50] B. Kim, and D. A. Landgrebe, "Hierarchical Classification in High Dimensional, Numerous Class Cases," Purdue University, West Lafayette, IN, Technical Report TR-EE 90-47, June 1990.

[51] R. Safavian, and D. A. Landgrebe, "A survey of decision tree classifier methodology.", *IEEE* Transactions on Systems, Man, and Cybernetics, Vol. 21, No, 3, pp. 660-674, 1991.

[52] L. Hyafil and R. L. Rivest, "Constructing optimal decision trees is NP-complete," Information Processing Letters, Vol. 5, No. 1, pp. 15-17, 1976.

[53] H. T. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, Cambridge, Massachusetts: MIT Press, 1990, pp. 329.

APPENDIX A

The Bhattacharyya distance is the sum of the contribution of the difference of the means and the difference of the covariances. $\mu = \mu_M + \mu_C$, where

$$\mu_M = \frac{1}{8}(\boldsymbol{M}_2 - \boldsymbol{M}_1)^T \overline{\Sigma}^{-1}(\boldsymbol{M}_2 - \boldsymbol{M}_1),\ \overline{\Sigma} = \left[\frac{\Sigma_1 + \Sigma_2}{2}\right] \tag{A.1}$$

and

$$\mu_C = \frac{1}{2} ln\left(\frac{|\overline{\Sigma}|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right)$$

For the two class problem in a d-dimensional space assume, without generality, the following.

$$\Sigma_1 = \begin{bmatrix} \sigma_{11}^2 & 0 \\ & \ddots \\ 0 & \sigma_{1d}^2 \end{bmatrix},\ \Sigma_2 = \begin{bmatrix} \sigma_{21}^2 & 0 \\ & \ddots \\ 0 & \sigma_{2d}^2 \end{bmatrix} \tag{A.2}$$

$$(\boldsymbol{M}_2 - \boldsymbol{M}_1) = [\varepsilon_1 \quad \cdots \quad \varepsilon_d]^T \tag{A.3}$$

then

$$\overline{\Sigma} = \begin{bmatrix} \overline{\sigma}_1^2 & 0 \\ & \ddots \\ 0 & \overline{\sigma}_d^2 \end{bmatrix} = \begin{bmatrix} \frac{(\sigma_{11}^2 + \sigma_{21}^2)}{2} & 0 \\ & \ddots \\ 0 & \frac{(\sigma_{1d}^2 \sigma_{2d}^2)}{} \end{bmatrix} \tag{A.4}$$

For that case, the computation of the mean and covariances components of Bhattacharyya distance are:

$$\mu_M = \frac{1}{8}\sum_{i=1}^{d} \frac{\varepsilon_i^2}{\overline{\sigma}_i^2} \tag{A.5}$$

$$\mu_C = \frac{1}{2} ln\left(\prod_{i=1}^{d}\left(\frac{\sigma_{1i}^2 + \sigma_{2i}^2}{2\sigma_{1i}\sigma_{2i}}\right)\right) = \frac{1}{2}\sum_{i=1}^{d} ln\left(\frac{\sigma_{1i}^2 + \sigma_{2i}^2}{2\sigma_{1i}\sigma_{2i}}\right) \tag{A.6}$$

APPENDIX B

The amount of energy that real sensors receive and their bandwidth is finite. As a consequence we can model ε_i^2 as a random variable that is defined over the range $\varepsilon_i^2 \in (E_{min}, E_{max})$ such that $E_{max} < \infty, \forall i$.

Under the assumption that the $E(\varepsilon_i^2)$ exist then:

$$E_{min} \leq E\left(\varepsilon_i^2\right) \leq E_{max} \tag{B.1}$$

$$Var(\varepsilon_i^2) = E(\varepsilon_i^4) - E^2(\varepsilon_i^2) \leq E_{max}^2 - E_{min}^2 \tag{B.2}$$

Both are 'finite quantities.