Supervised Classification in High Dimensional Space: Geometrical, Statistical and Asymptotical Properties of Multivariate Data[1]

Luis Jimenez and David Landgrebe[2]

Dept. Of ECE, PO Box 5000
University Of Puerto Rico
Mayaguez PR
00681-5000
jimenez@ece.uprm.edu

School of Elect. & Comp. Eng.
Purdue University,
West Lafayette, IN
47907-1285
landgreb@ecn.purdue.edu

This paper appeared in the IEEE Transactions on Geoscience and Remote Sensing,Vol 37, No. 6, November 1999.

Abstract

As the number of spectral bands of high spectral resolution data increases, the capability to detect more detailed classes should also increase, and the classification accuracy should increase as well. Often the number of labeled samples used for supervised classification techniques is limited, thus limiting the precision with which class characteristics can be estimated. As the number of spectral bands becomes large, the limitation on performance imposed by the limited number of training samples can become severe. A number of techniques for case-specific feature extraction have been developed to reduce dimensionality without loss of class separability. Most of these techniques require the estimation of 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 may not be as effective as they could be.

This suggests the need for reducing the dimensionality via a preprocessing method that takes into consideration high dimensional feature space properties. Such reduction should enable the estimation of feature extraction parameters to be more accurate. Using a technique referred to as Projection Pursuit, such an algorithm has been developed. This technique is able to bypass many of the problems of the limitation of small numbers of training samples by making the computations in a lower dimensional space, and optimizing a function called the projection index. A current limitation on this method is that as the number of dimensions increases, it is highly probable to find a local maximum

[1] Work reported herein was funded in part by NASA Grant NAGW-3924.

[2] Corresponding author.

of the projection index that does not enable one to fully exploit hyperspectral data capabilities. A method to estimate an initial value that can lead to a maximum that increases significantly the classification accuracy will be presented. This method leads also to a high dimensional version of a feature selection algorithm, which requires significantly less computation than the normal procedure.

I. Introduction

A study of high dimensional space characteristics and its implication for hyperspectral data analysis, reported previously [1, 29], presented a number of unusual characteristics of high dimensional data. Examples are that the volume in a hypercube has a tendency to concentrate in the corners, and in a hyperellipsoid in an outside shell. From these and others, it is apparent that high dimensional space is mostly empty. Multivariate data is usually located in a lower dimensional subspace. As a consequence, it is possible to reduce the dimensionality without losing significant information and separability among classes, however, to do so, one must have a means for finding the right subspace. Another consequence of these characteristics is that local neighborhoods are almost surely empty, requiring the bandwidth of estimation to be large and producing the effect of losing detail upon density estimation.

There has been some empirical and analytical research to determine what are adequate numbers of training samples for a given number of features. It is well known that the optimum number of features for classification is limited by the number of training samples [2]. Fukunaga [3], for example, proved that in a given circumstance, 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, including radial basis functions neural networks, the situation is even worse. It has been estimated that as the number of dimensions increases, the training sample size needs to increase exponentially in order to have an effective estimate of the multivariate densities needed to perform a nonparametric classification [4] [5]. These limitations are what has been called the curse of dimensionality [6]. This condition has restricted severely the practical applications of statistical pattern recognition procedures in high dimensional data. Due to the difficulties of density estimation in nonparametric approaches, a properly designed parametric version of a data analysis algorithm may be expected to provide better performance where only limited numbers of labeled samples are available to provide the needed a priori information.

High dimensional space characteristics can present problems to current feature extraction algorithms, e.g. Principal Components, Feature Subset Selection, Discriminant Analysis and Decision Boundary Feature Extraction [7]. Principal Component Analysis 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 [4, pp. 206]. 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.

Some authors have proposed algorithms by which a subset of features can be chosen from the original set [8, p. 164 ff]. 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 for a feature selection algorithm to be optimal, the search for a subset of features must be exhaustive [9]. The number of combinations of bands increases exponentially as the dimensionality increases and, as a result, an exhaustive search quickly become impractical or impossible.

Discriminant Analysis is a method that reduces the dimensionality optimizing the Fisher ratio [10]. 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. Finally, it performs the computations at full dimensionality, requiring a large number of labeled samples in order to accurately estimate parameters.

Lee and Landgrebe [7] proposed an algorithm based directly on decision boundaries. This method also 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. Its only problem is that it demands a high number of training samples for high dimensional space. This occurs because it computes 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].

Another relevant characteristic of high dimensional space is the fact that the assumption of normality will be better grounded in the projected subspace than at full dimensionality. It has been proved [12] [13] that as the dimensionality tends to infinity, lower dimensional linear projections will approach a normality model with probability approaching one. Normality in this case implies a normal or a combination of normal distributions.

For the circumstance where there are only a limited number of training samples, 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 and increase the ratio of labeled samples per feature, giving a better parameter estimation and better classification accuracy. Such 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 [14] [15].

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. It is supervised due to the fact that it does use labeled samples to estimate the Bhattacharyya distance under a parametric assumption: the Gaussian distribution of classes. Under that assumption we estimate two parameters: the mean and the covariances. The calculation is performed in the lower dimensional subspace where the data is to be projected. Such preprocessing is to be used before a feature extraction algorithm and classification process, as shown in Figure 1.

The preprocessing method developed in this present work will take into account a priori, problem-specific information. It will be developed after considering the characteristics of high dimensional space geometry and the statistics of hyperspectral data mentioned. Its objective is to linearly combine features, at the same time preserving the distance between classes. 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 [8, pp. 340].

In Figure 1 the different feature spaces have been labeled 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 all three procedures, preprocessing, feature extraction and classification use labeled samples as a priori information.

The approach proposed here 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.

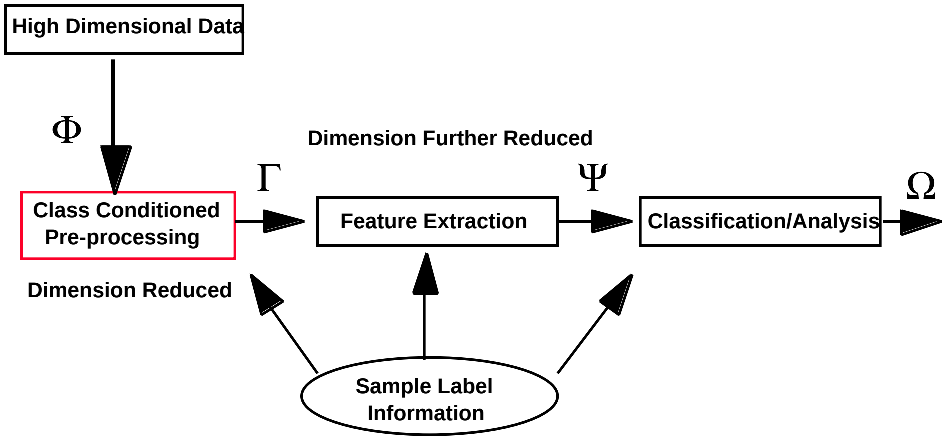


Fig. 1. Classification of high dimensional data including preprocessing of high dimensional data

Still, an algorithm is needed to find an initial choice for key parameters that enable it to arrive at an acceptable, though perhaps suboptimum solution. In a 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 [16]. 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.

II. Projection Pursuit and Dimension Reduction

Projection Pursuit has been defined as [4, pp. 208-212] "... 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 was used to select potentially interesting projections by the local optimization over projection directions of some index of performance. Projection Pursuit automatically picks a maximally effective 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 [17]. The idea was extended to projection pursuit regression [18] [19], and projection pursuit density estimation [20] [5]. Huber worked on the connection between projection pursuit and some other fields such as computer tomography, time series, and finite sample implementations [21].

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

- **X** is the initial multivariate data set (dxN). 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 $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$.

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(**A**) or I(**a**) in cases having a parametric vector instead of a matrix.

The choice of the projection index is the most critical aspect of this technique. What optimality means in this case depends on what function or projection index one uses. As mentioned before in remote sensing data analysis, optimality would certainly imply a projection that separates data into different meaningful classes that are exhaustive, separable, and of information value [8, pp. 340].

III. Parametric Projection Pursuit

Many nonparametric and unsupervised indices have been proposed with the purpose of maintaining the distance among clusters. One main advantage they have is the lack of assumption of an specific structure in the density functions and the lack of requirement of apriori knowledge in terms of labeled samples. That made those indices suitable for unsupervised approaches. At the same time those indices have significant disadvantages. There are a large number of free parameters in the estimation of the projection indices and the exact number is not well known in advance. This could lead to the problem of overfitting. In the case of having apriori knowledge in form of labeled samples, the unsupervised indices are not able to exploit such 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. Another disadvantage is related with the fact that some authors have suggested that data must be centered at zero and spherized in order to spread the data equally in all directions [4] . That action causes an enhanced contribution from noisy variables.

In the face of the disadvantages of the nonparametric and unsupervised projection indices discussed above, a parametric and supervised model is proposed in the present work. The analyst would use labeled samples in order to define classes explicitly, assuming the Gaussian distribution where only two parameters are required to be estimated. In addition, a convenient statistical distance among the classes plus some constraints on matrix **A** give sufficient flexibility for the development of a projection index that leads to the desired optimal situation.

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 2 and 3.

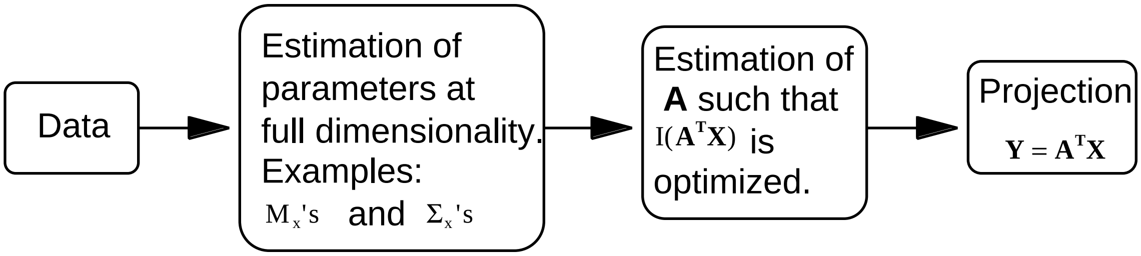

Data
Estimation of parameters at full dimensionality. Examples: M_x's and Σ_x's
Estimation of **A** such that $I(\mathbf{A}^T\mathbf{X})$ is optimized.
Projection $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$

Fig. 2. Discriminant Analysis process order.

Observe that Projection Pursuit starts with an a priori matrix $\hat{\mathbf{A}}$, then the parameters in a low dimensional space are estimated and matrix **A** is recomputed by optimizing the projection index $I(\mathbf{A}^T\mathbf{X})$. 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 Projection Pursuit to better handle the problem of small numbers of samples, the Hughes phenomena[3] [22], high dimensional geometrical and statistical properties, and the assumption of normality as previously mentioned. As seen in Figure 3, there is a feedback path from the output of the projected data **Y** to the block where the parameters are estimated again in the subspace. This enables the process of computing **A** with the new set of parameters. This process is continued until the increment in the iterations is below a certain level.

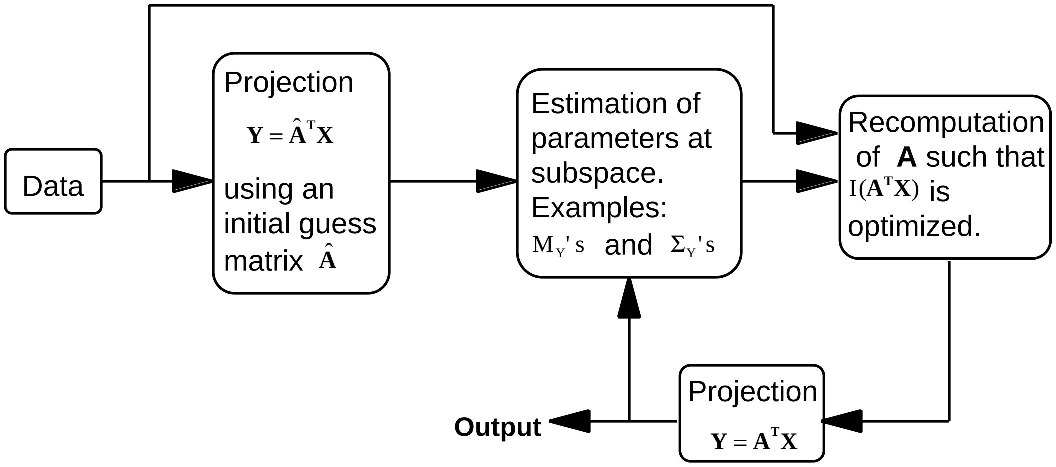

Data
Projection $\mathbf{Y} = \hat{\mathbf{A}}^T\mathbf{X}$ using an initial guess matrix $\hat{\mathbf{A}}$
Estimation of parameters at subspace. Examples: M_Y's and Σ_Y's
Recomputation of **A** such that $I(\mathbf{A}^T\mathbf{X})$ is optimized.
Projection $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$
Output

Fig. 3. Organization of the Projection Pursuit process.

[3] By Hughes effect is meant the effect whereby, for a fixed, finite number of training samples, as the number of features is increased, the classification accuracy first improves to a peak value, then declines. The decline is due to limitations on the precision of class statistics estimation resulting from the limited training data.

Given the objective of enhanced classification accuracy, we proposed the use of Bhattacharyya distance between two classes as the projection index, because of its relationship with Bayes classification accuracy and its use of both first and second order statistics [23]. The projection index for the two class case is thus:

$$I(\mathbf{A}^{T}\mathbf{X}) = \tfrac{1}{8}(\mathbf{M}_{2Y} - \mathbf{M}_{1Y})^{T}\left(\frac{\Sigma_{1Y} + \Sigma_{2Y}}{2}\right)^{-1}(M_{2Y} - M_{1Y}) + \tfrac{1}{2}\ln\left(\frac{\left|\frac{\Sigma_{1Y} + \Sigma_{2Y}}{2}\right|}{\sqrt{|\Sigma_{1Y}||\Sigma_{2Y}|}}\right) \tag{1}$$

where $\mathbf{M}_{jY}$ and Σ_{jY} are the mean vector and the covariance matrix respectively of the jth class in the projected subspace **Y**. In the case of more than two classes the minimum Bhattacharyya distance among the classes can be used after the Bhattacharyya distance is calculated for all combinations of pairs of two classes. Then the minimum of the Bhattacharyya distance is chosen as in (2).

$$I(\mathbf{A}^{T}\mathbf{X}) = \min_{i\in C}\left\{\tfrac{1}{8}(\mathbf{M}^{i}_{2Y} - \mathbf{M}^{i}_{1Y})^{T}\left(\frac{\Sigma^{i}_{1Y} + \Sigma^{i}_{2Y}}{2}\right)^{-1}(M^{i}_{2Y} - M^{i}_{1Y}) + \tfrac{1}{2}\ln\left[\frac{\left|\frac{\Sigma^{i}_{1Y} + \Sigma^{i}_{2Y}}{2}\right|}{\sqrt{|\Sigma^{i}_{1Y}||\Sigma^{i}_{2Y}|}}\right]\right\} \tag{2}$$

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

$$C = \frac{L!}{2!(L-2)!} \tag{3}$$

From ancillary sources, the analyst can define the classes and estimate the mean and covariance of each.

The computation of the parametric matrix **A** can lead to some problems. The columns of **A** should be linearly independent to ensure redundancies in the features of the projected data are avoided. 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, for example, to 20 and avoiding such problems in the process of optimization of the projection index could be difficult. In order to overcome such obstacles, a set of constraints on the matrix **A** are imposed.

Projecting Adjacent Groups of Features: Sequential Projection Pursuit

In this section the special constraints imposed on the **A** matrix will be explained. The approach is to divide the bands in the space Φ into a partition of groups of adjacent bands in order to project each group to one. **A** can be rewritten as: $\mathbf{A} = [\mathbf{A}_1 \ \mathbf{A}_2 \ \cdots \ \mathbf{A}_{Col-1} \ \mathbf{A}_{Col}]$, were $\mathbf{A}_i$ is the ith 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: $\mathbf{a}_i = \left[a_{1i} \quad a_{2i} \quad \ldots \quad a_{n_i i}\right]^T$. $\mathbf{A}_i$ will combine 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≠ j $\mathbf{A}_i^T \cdot \mathbf{A}_j = 0$.

The physical interpretation of the constraints is shown in Figure 4. Every group of n_i adjacent bands is 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 interpretation. These projections correspond in Figure 1 to a mapping from the original space Φ to the subspace Γ.

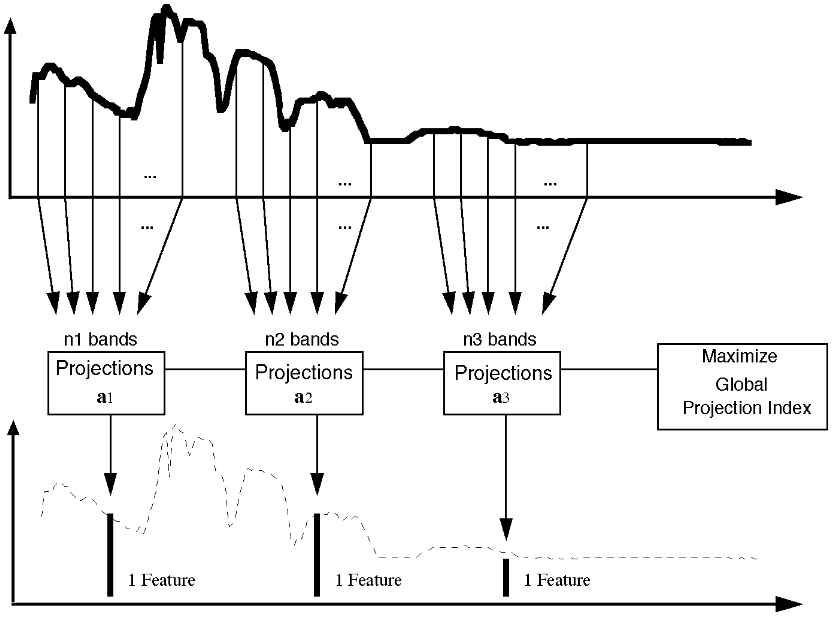


Fig. 4. Sequential Parametric Projection Pursuit.

Advantages that this set of constraints provide to the optimization process are (1) it is fast, (2) it preserves the order of the features in the class spectral response, (3) it is flexible in terms of the number of adjacent bands to be combined, (4) it can take into consideration any available ground truth information and the interest of the analyst, (5) the $\mathbf{A}_i$ columns are orthogonal, allowing the algorithm to avoid linear dependencies among $\mathbf{A}_i$'s, (6) it will make easier the process to construct an initial choice for matrix $\hat{\mathbf{A}}$. 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 bands? 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 $\mathbf{Y}$ is localized.

This algorithm will project every group of neighboring bands into one feature, maximizing the global projection index in the projected subspace Γ. The final number of features is the dimensionality of the projected subspace. This algorithm can be time consuming and the number of parameters that are required to be estimated is high. A modification is performed to optimize sequentially and iteratively the projection index. The rational of this is that at every iteration fewer parameters are estimated to perform the optimization. This approach will be called Sequential Parametric Projection Pursuit. The iterative procedure is as follows:

(1) An initial choice for every $\mathbf{a}_i$ for every group of adjacent bands is made and 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) Repeat the procedure for the ith group where $\mathbf{a}_i$ is calculated, optimizing against the global Bhattacharyya distance while maintaining the $\mathbf{a}_j$'s constant, where $i \neq j$.
(4) When the last group of adjacent bands is projected, repeat the process from step 2 (compute all the $\mathbf{a}_j$'s sequentially) until the maximization ceases increasing significantly. The significant increment is relative to each iteration. If one iteration (step 2 and 3) is complete and the percentage of maximization of the global projection index is less than a threshold, then it stops the process.

Figure 5 shows the process. The arrow points to the vector that is optimized at each step and the corresponding group of adjacent bands that are linearly combined to produce one feature.

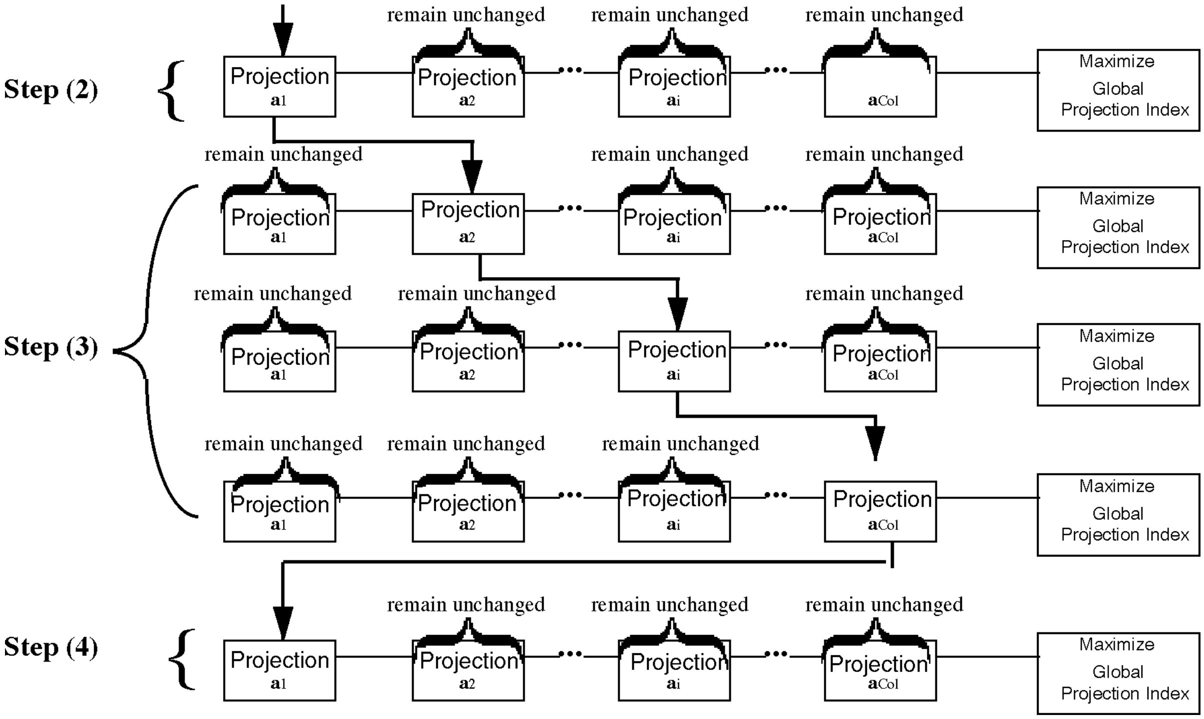


Fig. 5 Iterative procedure to maximize the global Projection Index.

In practice, due to the Gaussian assumption, the sequential training is faster and less prone to local minima.

IV. 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 1 is divided into two stages as shown in Figure 6. The first one has the objective of estimating an initial choice of matrix $\hat{\mathbf{A}}$. That matrix is suboptimum with respect to the maximization of the global projection index. The estimation of this parametric matrix is based on the initial vectors $\hat{\mathbf{a}}_i$'s and the number of adjacent bands n_i combined in each group in the partition of features shown in Figure 4. The second stage is the numerical optimization of the global projection index in order to estimate **A**, as explained earlier it uses an iterative form of optimization.

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

Each group of adjacent bands will have a set of trial values, $\hat{\mathbf{a}}_i$'s. In this section we will assume that the values of $\mathbf{n}_i$ are given. The procedure to calculate these values will be explained in the next section. The matrix $\hat{\mathbf{A}}$ will be constructed by choosing one trial value, $\hat{\mathbf{a}}_i$ from each set. Among these trial values 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:

$$\mathrm{Bhatt}_M = \frac{1}{8}\left(\mathbf{M}_2 - \mathbf{M}_1\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(\mathbf{M}_2 - \mathbf{M}_1\right) \tag{4}$$

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

$$\mathrm{Bhatt}_C = \frac{1}{2}\ln\left(\frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{\left|\Sigma_1\right|\left|\Sigma_2\right|}}\right) \tag{5}$$

This can be rewritten in the following form [23, pp. 455-457]:

$$\mathrm{Bhatt}_C = \frac{1}{4}\left\{\ln\left|\Sigma_2^{-1}\Sigma_1 + \Sigma_1^{-1}\Sigma_2 + 2\mathbf{I}\right| - n\ln 4\right\} \tag{6}$$

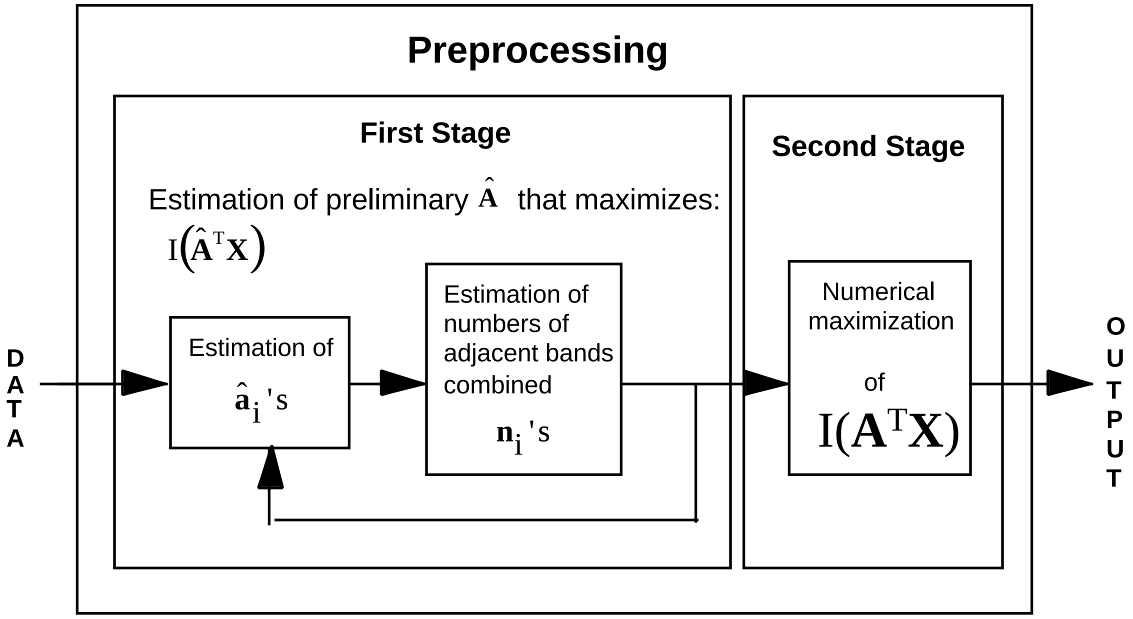

Preprocessing
First Stage
Estimation of preliminary $\hat{\mathbf{A}}$ that maximizes:
$I(\hat{\mathbf{A}}^T\mathbf{X})$
Second Stage
DATA
Estimation of
$\hat{\mathbf{a}}_i$'s
Estimation of numbers of adjacent bands combined
$\mathbf{n}_i$'s
Numerical maximization
of
$I(\mathbf{A}^T\mathbf{X})$
OUTPUT

Fig. 6. Preprocessing block.

The mean difference portion (Bhatt_M) is maximized by the vector $\mathbf{a}_{\mathrm{Mmax}}$ [23, pp. 455-457]:

$$\mathbf{a}_{\mathrm{Mmax}} = \left(\mathbf{M}_2 - \mathbf{M}_1\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \tag{7}$$

In order to compute the vector that maximizes the covariance difference element, a prior matrix Λ must be computed. That matrix is defined as:

$$\Lambda = \Sigma_2^{-1}\Sigma_1 \tag{8}$$

The vector that maximizes Bhatt_C, $\mathbf{a}_{\mathrm{Cmax}}$, is the eigenvector of Λ that corresponds to the eigenvalue that maximizes the function $f(\lambda_i)$:

$$\arg\max_{\lambda_i} f\left(\lambda_i\right) = \arg\max_{\lambda_i}\left[\lambda_i + \frac{1}{\lambda_i} + 2\right] \tag{9}$$

where λ_i is the ith eigenvalue of Λ. That vector optimizes the following linear transformation:

$$J(d) = \ln\left|\left(\mathbf{a}^T\Sigma_2\mathbf{a}\right)^{-1}\mathbf{a}^T\Sigma_1\mathbf{a} + \left(\mathbf{a}^T\Sigma_1\mathbf{a}\right)^{-1}\mathbf{a}^T\Sigma_1\mathbf{a} + 2\mathbf{I}_d\right| \tag{10}$$

where d is the dimensionality of the data. It follows that $\mathbf{a}_{\mathrm{Cmax}}$ maximizes Bhatt_C.

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. All these vectors are stored as a set of trial values for the **a**'s vectors. Each group of adjacent bands will have a set of these trial values as a form of bank of possible **a**'s vectors. 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 bands in each group of adjacent bands, then the total number of initial choices $\hat{\mathbf{a}}_i$'s in the ith group of adjacent bands are:

$$Total = 2\frac{K!}{2!(K-2)!} + n_i + 1 \tag{11}$$

The first element corresponds to twice the number of every combination of two classes, corresponding to $\mathbf{a}_{\mathrm{Mmax}}$ and $\mathbf{a}_{\mathrm{Cmax}}$. 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 of 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 n_i 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 increasing significantly.

This process produces a suboptimal matrix $\hat{\mathbf{A}}$ that will be used with the numerical optimization stage of the preprocessing block. Note that 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.

V. 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 6, 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 [24]. Also they have been applied in pattern recognition and remote sensing image analysis [25]. 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 [26]. Some authors have applied them in the design of hierarchical classifiers that decide at each node to which class a particular sample belongs [27]. The basic idea of the decision tree is to break a particular complex problem into a number of simpler ones that can be more easily solved. It is assumed that when the solutions are combined an approximately optimum global solution results.

It has been demonstrated that an optimal decision tree is an N-P complete problem [28]. 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 begin by separating 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 combine classes until the root node is reached. In the hybrid approach the bottom-up procedure is used to aid the top-down approach. Finally, in the tree growing-pruning approach, the tree is allowed to grow to its maximum size and then the tree is pruned.

Top-Down

This algorithm starts to construct the feature space as a partition of groups of adjacent bands. Each group of adjacent bands will be projected to one feature in the projected subspace.

The procedure is repeated successively in the following steps:

(1) Divide independently each group of adjacent bands into two new groups, creating new independent sets of groups of adjacent bands.
(2) For each set compute the global Projection Index and compute the increment in the Projection Index ΔB_i . ΔB_i is the increment in Bhattacharyya distance with respect to the previous division.
(3) 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:

$$\Delta BI_i = \frac{\max\left(\Delta B_i\right)}{PI_{i-1}} \tag{12}$$

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 significant, below a certain threshold, or until the algorithm reaches a maximum number of features establish by the analyst or by the number of label samples.

Bottom-Up

This algorithm starts with a number of features in the projected subspace where each one corresponds to one group of adjacent bands in the partition of the high dimensional space. The goal of this procedure is to reduce the number of dimensions of the lower dimensional subspace without reducing the Projection Index significantly.

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 $\hat{\mathbf{a}}_\mathbf{i}$'s will be calculated. 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:

$$\Delta BD_i = \frac{\min\left(\Delta B_i\right)}{PI_{i-1}} \tag{13}$$

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.

In both, Top-Down and Bottom-Up methods there are two basic assumptions that will be explained here.

- As the Top-Down method increases the dimensionality of the projected sub-space, the Global Projection Index used does not decrease. This ensures that ΔBI_i is less than 100%.
- As the Bottom Up method decreases the dimensionality of the projected sub-space, the Global Projection Index does not increase. This ensures that ΔBD_i is less than 100%.

For the case of using the Bhattacharyya distance as the Global Projection Index both assumptions were proved in [1].

A binary tree algorithm will be used here to estimate the suboptimum number of adjacent bands that should be linearly combined in order to reduce the dimensionality. The heuristic approach used is that of a hybrid decision tree. The following explains how a hybrid heuristic approach can be applied in an algorithm to accomplish the objective of the second block in the first stage of Figure 6 [29]. 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 the global projection index 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 of the projection index 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.

High Dimensional Projection Pursuit Feature Selection

Henceforth we will call the Parametric Sequential Projection Pursuit algorithm just Projection Pursuit. It will use the methods described previously, equivalent to stage 1 in Figure 6, in order to estimate $\hat{\mathbf{A}}$. Then it uses a numerical optimization algorithm equivalent to stage 2 in Figure 6 to finally compute **A**. This algorithm can easily be modified for a Projection Pursuit version of a supervised band subset selection algorithm (also named Feature Selection). Projection Pursuit Feature Selection (PP-Opt-FS) uses the method explained for constructing $\hat{\mathbf{A}}$ with a significant transformation. Every bank described of stored $\hat{\mathbf{a}}_i$'s will only contain vectors that choose one band 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 6, 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. The normal feature selection is an exhaustive procedure that searches for the best combination of bands. If such method were applied to obtained the best subset of 20 bands in a set of 200, one would have to make a number of computations in the order of 10^{26}. Projection Pursuit Feature Selection has significantly less computation for high dimensional data than a normal supervised feature selection algorithm.

Computational Complexity of the Binary Decision Trees

In this section the complexity of the decision trees used in the First Stage of Figure 6 is computed. Due to the binary nature of the decision trees, the branching factor is 2. If the Top-Down method is used at level **i** the number of nodes that are further expanded is 2**i**. At level 0 there is only one node (the root node), at level 1 two nodes are expanded, at level 2 there are four, etc. If the solution is located in level d then there are:

$$1+\sum_{i=1}^{d} 2\cdot i = 1+2\sum_{i=1}^{d} i = 1+2\cdot\left[\frac{1}{2}d(d+1)\right] = 1+d+d^2 \tag{14}$$

Due to the nature of the Bottom-Up method that will be used as an aid to the Top-Down scheme, its complexity will be discussed as part of the hybrids. It is important to realize that if there are B groups, then the Bottom-Up method will produce B-1 combination of nodes.

The hybrid II method is more complex than the hybrid I. At level **i** the Top-Down method expands to 2**i** nodes and grows only by a factor of **i** + 1 nodes. Then using the Bottom-Up method the number of nodes that are combined is **i** + 1 - 1 = i. If the solution is at a level d of the binary decision tree then the total number of nodes that are expanded and combined is:

$$1+\sum_{i=1}^{d}(2\cdot i+i) = 1+\sum_{i=1}^{d}3\cdot i = 1+3\sum_{i=1}^{d} i = 1+\frac{3}{2}\left[d(d+1)\right] \tag{15}$$

As shown by equation (11) assuming there are K classes and n_i bands in each group of adjacent bands, then the maximum number of initial choices $\hat{\mathbf{a}}_i$'s in the i^{th} group of adjacent bands is:

$$Total_{Max} = \max(\mathrm{Total}) = \max\left[2\frac{K!}{2!(K-2)!} + n_i + 1\right] = K(K-1) + MNB + 1 \tag{16}$$

where MNB is the maximum number of band analyzed. $\mathrm{Total}_{\mathrm{Max}}$ is the maximum number of initial choices per node. Then the total number of initial choices $\hat{\mathbf{a}}_i$'s produced in the decision tree is the total number of nodes times the total number of initial choices.

For the Top-Down method that amount is bounded by the following amount:

$$\mathrm{TNIC} \le (1 + d + d^2)\cdot\left[K(K-1) + MNB + 1\right] = O\left[d^2\left(K^2 + MNB\right)\right] \tag{17}$$

where TNIC is the total number of initial choices.

For the hybrid II method the amount is bounded by the following amount:

$$\mathrm{TNIC} \le \left[1 + \frac{3}{2}\left(d + d^2\right)\right]\cdot\left[K(K-1) + MNB + 1\right] = O\left[d^2\left(K^2 + MNB\right)\right] \tag{18}$$

Observe that both algorithms solve the problem in polynomial time.

VI. Experiments

A series of experiments was developed to demonstrate the algorithm. The hyperspectral data used in these experiments is a segment of one AVIRIS data scene taken of NW Indiana's Indian Pine test site. Figure 7 shows the AVIRIS scene used in the experiments. From the original 220 spectral channels 200 were used, discarding the atmospheric absorption bands[4]. The training samples required for classifying data and the testing samples use to estimate the classification accuracy were obtained from available ground truth. The estimated parameters were based on the spectral responses, and the classification was performed on a pixel by pixel basis.

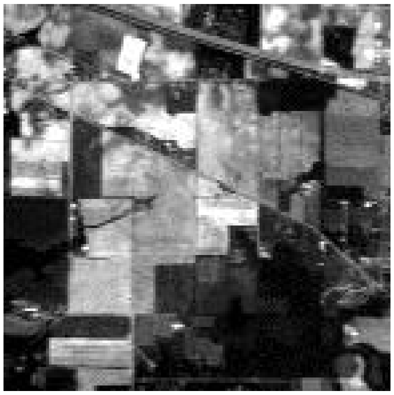


Fig. 7. AVIRIS NW Indiana's Indian Pine test site.

The procedure of the experiments is to use the first stage algorithm to calculate $\hat{\mathbf{A}}$ for Projection Pursuit and Projection Pursuit Feature Selection. Then **A** is calculated with a

[4] The atmospheric absorption bands are channels that are opaque, due to constituents of the atmosphere such as water vapor, CO_2 and O_3. Principal opaque bands in this range are near 0.9 and 1.4 µm. Since radiation measured by the sensor at these wavelengths originates from within the atmosphere and not the surface, it does not carry information relevant for classification purposes.

numerical analysis stage for Projection Pursuit. The results are compared with direct use of a feature extraction process in terms of classification accuracy. Two different experiments explore two cases: having a small number of classes and training samples and having a somewhat larger number of classes and training samples.

Experiment 1: Small number of classes and training samples.

Several different projection pursuit methods were compared with a more conventional feature extraction approach. In the present experiment four classes were defined: corn, corn-notill, soybean-min, and soybean-notill. These classes in this data set form a particularly challenging classification problem. The data were collected in early season when the corn and soybean vegetation had only about 5% ground cover. The notill and min designations indicate that these fields had undergone no or minimum tillage before planting so that substantial debris from the previous year crop was still present on the surface. Combine this with the natural variation in the soil patterns, and one has a very significant background variation against which to attempt to distinguish between corn and soybeans, which themselves are not greatly different spectrally. 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 1 shows the number of training and test samples for each class.

Table 1

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

This experiment will enable us to compare the direct use of Discriminant Analysis at full dimensionality, Projection Pursuit with only an iterative and sequential numerical optimization stage, Projection Pursuit with a first stage that estimates the initial matrix $\hat{\mathbf{A}}$ and a numerical optimization stage, and the Projection Pursuit Feature Selection algorithm where a subset of bands was chosen. All of these are under the condition of having a small number of labeled samples.

The methods used are as follows.

- (DA 100-20) The multispectral data was reduced in dimensionality from 200 dimensions in Φ space to a subspace. Using Discriminant Analysis at full dimensionality 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).

- (PP) Iterative Sequential Projection Pursuit with only a numerical optimization stage, the second stage in Figure 6, applied to the data in order to reduce the dimensionality, maximizing the minimum Bhattacharyya distance among the classes. It is iterative in the sense that the numerical optimization is performed in the iteration form explained in section III and shown in Figure 5. In this approach the number of adjacent bands combined in each group was 10 and the initial vector for maximization was chosen to be a vector that averages the adjacent bands on a group. This method does not use the decision tree to find the number of bands required to be combined. It is also named uniform band selection.

- (PP-Opt) Optimum Projection Pursuit with the first stage that estimates matrix $\hat{\mathbf{A}}$, and the numerical optimization stage, used to project from 200 to a 20 dimensional subspace Γ. The algorithm estimates the dimensionality of the data as 20.

- (PP-Opt-FS) Projection Pursuit Feature Selection was used to project the data to a subset of bands that is suboptimum in the sense of maximizing the Bhattacharyya distances among the classes. This algorithm uses the feature selection procedure described in section V and is also named as subset selection. The dimensionality of the subspace where the data was projected, Γ, was estimated as 16 by this algorithm.

The dimensionality of Ψ was estimated by Discriminant Analysis in the (DA 100-20) method. Iteratively Sequential Projection Pursuit (PP) arbitrarily reduces the dimensionality of Γ to 20 (20 groups of adjacent bands, projecting every group to 1 feature). This is due to the fact that it only uses the numerical maximization stage (second stage in Figure 6). Projection Pursuit Feature Selection (PP-Opt-FS) and the optimum version of Sequential Projection Pursuit (PP-Opt) were used as described previously. Both use the hybrid II heuristical approach to construct the a priori matrix $\hat{\mathbf{A}}$ with thresholds τ_{T-D} and τ_{D-T} equal to .005. Both procedures chose the dimensionality of the subspace Γ by the empirical estimation procedure described in section V. Estimating the n_i's the dimensionality is estimated as well ($\sum_i n_i = 200$ = Total number of bands).

In the Projection Pursuit based algorithms, after the dimensionality of the data is reduced, Discriminant Analysis is used as a feature extraction algorithm in order to project the data from Γ to Ψ. Two types of classifiers were used. The first one is a ML classifier and the second is a ML with 2% threshold. In the second, a threshold was applied to the standard classifier whereby, in case of normal distributions of the class data, 2% of the least likely points will be thresholded. In other words, this is a rejection criterion were the pixel $\boldsymbol{x}$ is rejected by the following rule:

$$\left(\boldsymbol{x}-\mathbf{M}_i\right)^T \Sigma_i^{-1}\left(\boldsymbol{x}-\mathbf{M}_i\right) > T \tag{19}$$

where $\mathbf{M}_i$ and Σ_i are the mean and covariance of class i to which the ML classifier assign pixel **X**. T is obtained in a Chi Square distribution table. When a pixel is rejected it is labeled as a thresholded pseudo class. In the case of having a normal distribution, the percentage of accuracy should drop 2% of the values in the ML classifier (without threshold). 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.

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 2.

Observe that the Projection Pursuit based algorithms preserved more information in terms of the 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., the 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 IV, has the best performance. It has an improvement of 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. Tables 3 and 4 show the number of bands in adjacent groups for PP-Opt and PP-Opt-FS.

Table 2
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

Table 3.
Number of bands in adjacent groups for PP-Opt

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}	n_{16}	n_{17}	n_{18}	n_{19}	n_{20}
Number of adjacent bands/ group	20	10	5	5	10	10	20	5	5	10	10	5	5	20	5	5	10	20	10	10

Table 4.
Number of bands in adjacent groups for PP-Opt-FS

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}	n_{16}
Number of adjacent bands/ group	6	6	7	6	9	10	6	6	3	4	12	12	13	25	25	50

Figure 8 and 9 will show the results of classification accuracy, comparing the results of direct projection from Φ space to Ψ using Discriminant Analysis (DA 100-20) with projecting the preprocessed data from the Γ subspace to Ψ. The comparison is in terms of test field classification accuracy. The Discriminant Analysis method was used to project data from the Γ subspace to Ψ after the Projection Pursuit based methods were applied. This provides a direct comparison against direct projection from Φ to Ψ (DA 100-20) because the same feature extraction procedure was used at Φ space and at Γ subspace. After Discriminant Analysis was applied to data sets preprocessed by the Projection Pursuit based algorithms, they were classified and the test fields classification results can be seen in Figures 8 and 9.

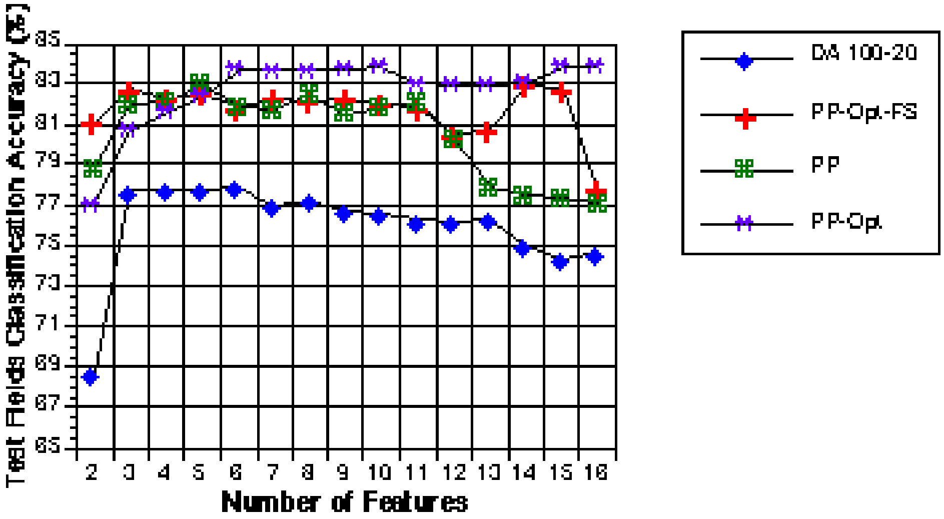


Fig. 8. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML classifier.

The classification accuracy results on the test fields using the Maximum Likelihood classifier can be seen in Figure 8. Observe in Figure 8 that Projection Pursuit's classification accuracies are much better than using direct Discriminant Analysis (100-20). Projection Pursuit Optimum becomes the best method as the number of dimension increases. It better overcomes the Hughes phenomena and deals with 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 well as PP-Opt or PP-Opt-FS as the number of dimensions increases. From Figure 9 it can be seen that the Projection Pursuit approaches performed significantly better, with a difference as much as 45%, compared to 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.

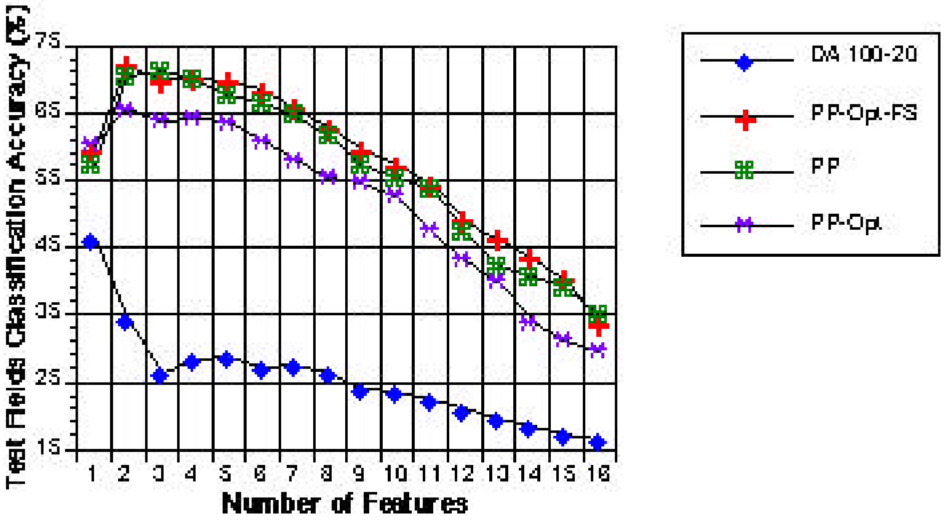


Fig. 9. Test fields classification accuracy comparison between direct use of Discriminant Analysis (DA 100-20) and the use of Discriminant Analysis after different methods based on Projection Pursuit (PP, PP-Opt, PP-Opt-FS) for ML with 2% threshold.

Experiment 2: Large number of classes and training samples.

The hyperspectral data used in these experiments is the same segment of AVIRIS data used in experiment 1. As before, 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 was 1790 and the total number of test samples was 1630. Table 5 shows the defined classes and their respective number of training and test samples.

Table 5

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 methods used are as follows.

- (DBFE) The multispectral data was reduced in dimensionality from 200 dimensions in Φ space to a subspace. Using Decision Boundary Feature Extraction at full dimensionality the data was reduced from 200 to a 22 dimensional subspace Ψ .

- (DAFE) The multispectral data was reduced in dimensionality from 200 dimensions in Φ space to a subspace. Using Discriminant Analysis Feature Extraction at full dimensionality the data was reduced from 200 to a 22 dimensional subspace Ψ.

- (PP-Opt) Optimum Projection Pursuit with the first stage that estimates matrix $\hat{\mathbf{A}}$, and the numerical optimization stage, used to project from 200 to a 20 dimensional subspace Γ. The algorithm estimates the dimensionality of the data as 22.

- (PP-Opt-FS) Projection Pursuit Feature Selection was used to project the data to a subset of bands that is suboptimum in the sense of maximizing the Bhattacharyya distances among the classes. This algorithm uses the feature selection procedure described in section V. The dimensionality of the subspace where the data was projected, Γ, was estimated as 22.

In the third and fourth methods Projection Pursuit (PP-Opt) and Projection Pursuit Feature Selection (PP-Opt-FS) were used to reduce the dimensionality from 200 to 22. These methods linearly project the data from Φ to Γ subspace. After the Projection Pursuit preprocessing method was applied, 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 dimensions. Two types of classifiers were again used: ML classifier and ML with 2% threshold.

The results of the minimum Bhattacharyya distances for Decision Boundary, Discriminant Analysis, Projection Pursuit (PP-Opt), and Projection Pursuit Feature Selection (PP-Opt-FS) are shown in table 6 for Γ in 22 dimensions. Tables 7 and 8 show the number of bands in adjacent groups for PP-Opt and PP-Opt-FS.

Table 6

Method	DBFE	DAFE	PP-Opt	PP-Opt-FS
Minimum Bhattacharyya Distance	2.64	1.52	2.75	1.90

Table 7.
Number of bands in adjacent groups for PP-Opt

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}	n_{16}	n_{17}	n_{18}	n_{19}	n_{20}	n_{21}	n_{22}
Number of adjacent bands/ group	7	6	6	6	6	3	3	7	6	6	6	7	6	25	12	6	7	6	6	13	25	25

Table 8.
Number of bands in adjacent groups for PP-Opt-FS

	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{15}	n_{16}	n_{17}	n_{18}	n_{19}	n_{20}	n_{21}	n_{22}
Number of adjacent bands/ group	25	4	3	3	1	1	1	12	6	3	3	13	13	12	25	3	3	6	13	13	12	25

Projection Pursuit (PP-Opt) was able to have a larger projection index than the other methods. The next sections will apply the feature extraction techniques after the use of Projection Pursuit-based algorithms and compare their results with direct application of Decision Boundary and Discriminant Analysis in Φ .

Decision Boundary Feature Extraction

The following experiments have the objective of testing how Projection Pursuit based algorithms enhance test field classification accuracy in the use of Decision Boundary Feature Extraction at 22 dimensions in Γ in comparison with direct use of Decision Boundary at full dimensionality in Φ space. Figures 10, 11, 12, and 13 show the results for ML classifications. The results of direct use of Decision Boundary at 200 dimensions are labeled DBFE, results of Decision Boundary applied after PP-Opt preprocessing are labeled PPDBFE and the results of Decision Boundary after PP-Opt-FS are labeled PPFSDBFE.

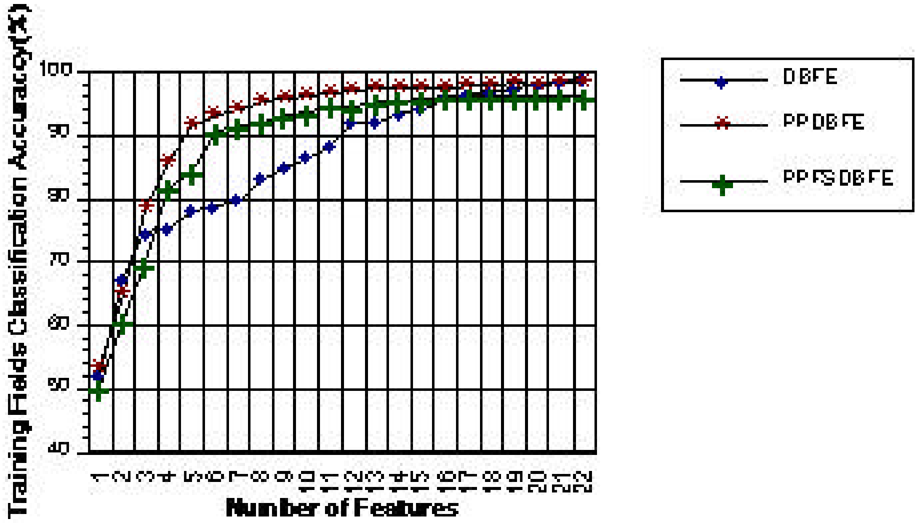


Fig. 10. Training fields classification accuracy comparison between direct use of Decision Boundary Feature Extraction (DBFE) and the use of Decision Boundary Feature Extraction after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML classifier.

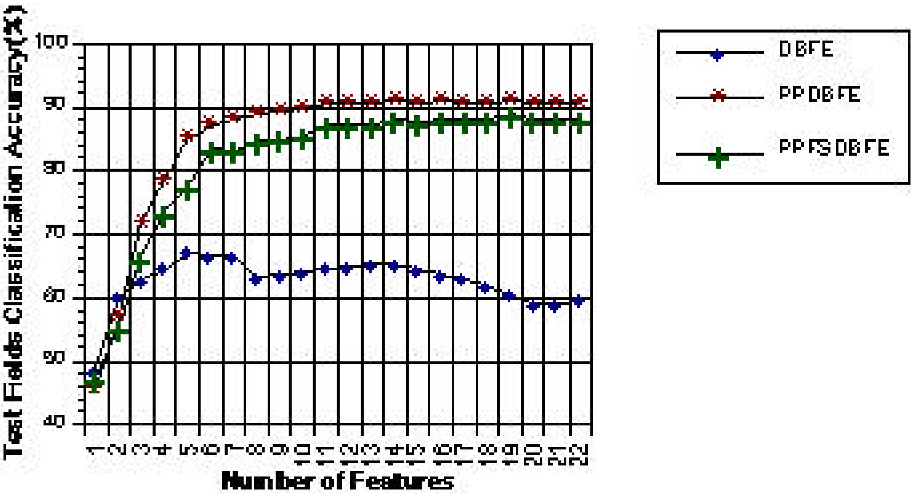


Fig. 11. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML classifier.

In terms of training fields, Projection Pursuit (PPDBFE) and Projection Pursuit Feature Selection (PPFSDBFE) increase in classification accuracy faster as a function of the number of features than direct use of Decision Boundary Feature Extraction (DBFE). As expected in a significant range PPFSDBFE results are in between PPDBFE and DBFE. At 22 dimensions the performance of PPDBFE and DBFE are close and both of them are superior to PPFSDBFE in accordance with the values of the minimum Bhattacharyya distance at 22 dimensions as shown in Table 4. 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 12 and 13 that PPDBFE and PPFSDBFE maintain the data more in clusters with the assumption of normality better supported. At 22 features there is a difference of 65% between Projection Pursuit based algorithms and direct application of Decision Boundary Feature Extraction in the test fields classification accuracy with the use of a 2% threshold.

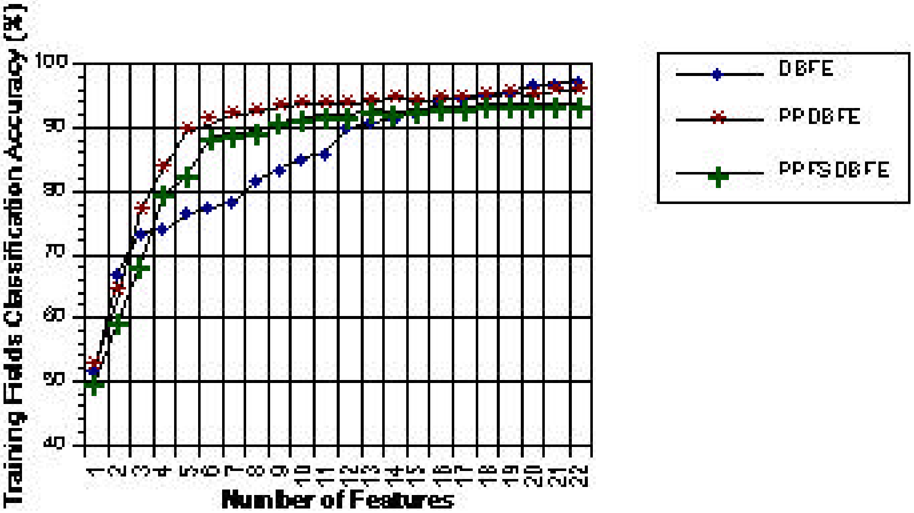


Fig. 12. Training fields classification accuracy comparison between direct use of Decision Boundary Feature Extraction (DBFE) and the use of Decision Boundary Feature Extraction after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML with 2% threshold.

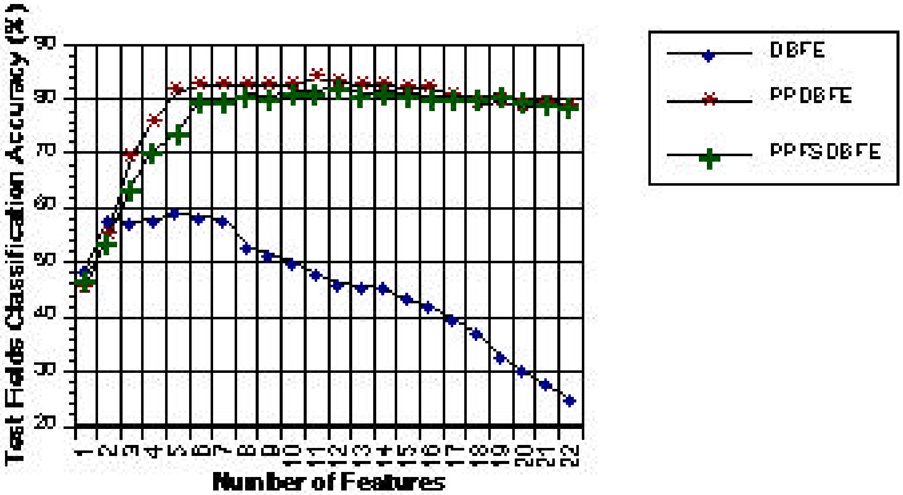


Fig. 13. Test fields classification accuracy comparison between direct use of Decision Boundary (DBFE) and the use of Decision Boundary after different methods based on Projection Pursuit (PPDBFE and PPFSDBFE) for ML with 2% threshold.

Observing figures 10, 11, 12 and 13 we can see the relative behavior of the performance curves for training and test sets. The training set accuracy keeps improving with the increasing features. The difference is that Projection Pursuit's methods keep growing faster relative to direct use of Decision Boundary at 200 bands.

In terms of the test sets the direct use of Decision Boundary increases the accuracy until 5 features, then it starts to decrease due to the limited number of training data causing overfitting. Projection Pursuit's based methods have classification accuracies that overcome the problem of overfitting. These classifiers keep increasing the accuracy until 11 features, and adding more features then produces only a slow reduction of it.

Discriminant Analysis

In this experiment three procedures were used to project the data to a 22 dimensional subspace. The first was direct application of Discriminant 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 was 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 14, 15, 16 and 17 show the results with the ML classifier.

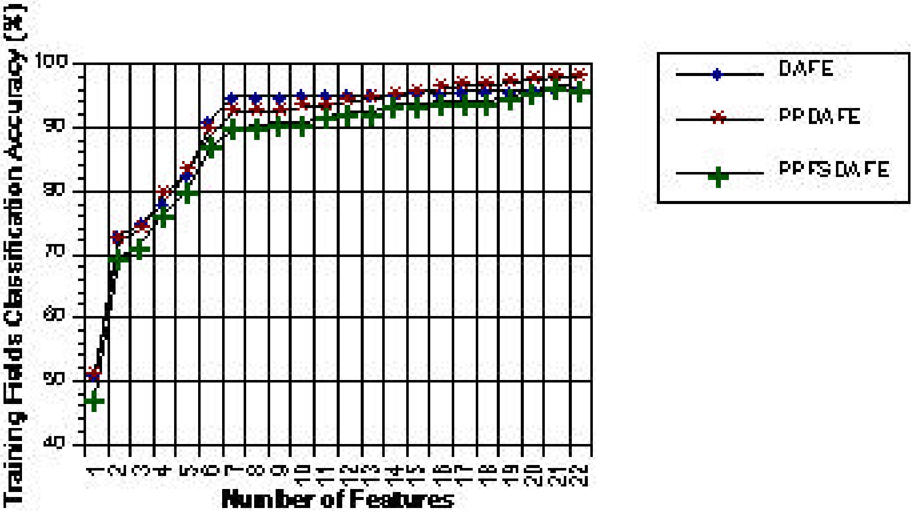


Fig. 14. Training fields classification accuracy comparison between direct use of Discriminant Analysis Feature Extraction (DAFE) and the use of Discriminant Analysis Feature Extraction after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML classifier.

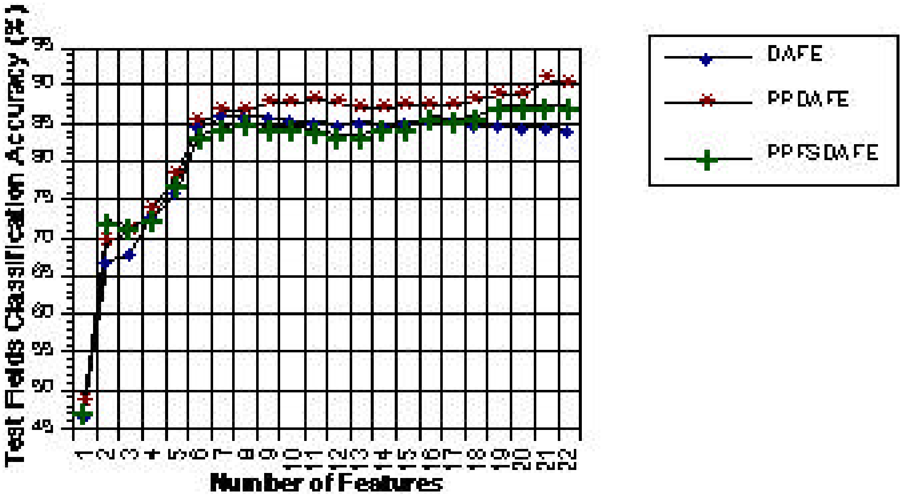


Fig. 15. Test fields classification accuracy comparison between direct use of Discriminant Analysis Feature Extraction (DAFE) and the use of Discriminant Analysis Feature Extraction after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML classifier.

In terms of the training fields, the classification results are very similar. In the test fields Projection Pursuit's algorithms perform better. The difference there is significant. It is not as dramatic as in Decision Boundary Feature Extraction because this last method of feature extraction requires more training samples per feature than Discriminant Analysis. Note in Figure 15 that PPDAFE and PPFSDAFE are able to grow after 7 features. This is due to the fact that the 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 stops growing 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.

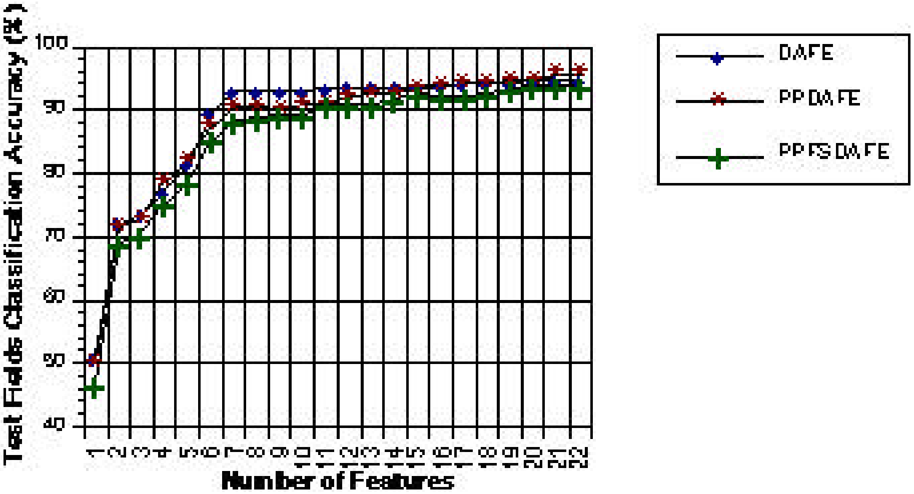


Fig. 16. Training fields classification accuracy comparison between direct use of Discriminant Analysis (DAFE) and the use of Discriminant Analysis Feature Extraction after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML with 2% threshold.

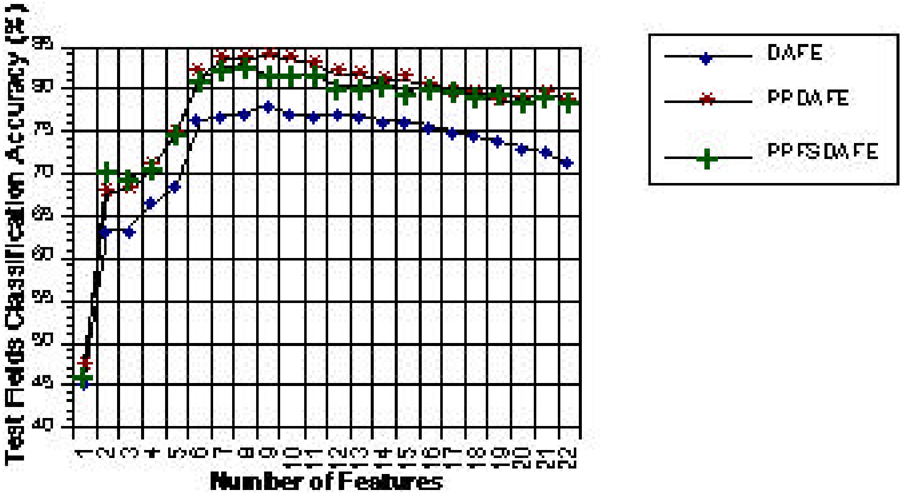


Fig. 17. Test fields classification accuracy comparison between direct use of Discriminant Analysis Feature Extraction (DAFE) and the use of Discriminant Analysis Feature Extraction after different methods based on Projection Pursuit (PPDAFE and PPFSDAFE) for ML with 2% threshold.

VII. 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 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 effective. Projection Pursuit is the method used to 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.

A first stage of preprocessing has been proposed in order to estimate a preliminary matrix $\hat{\mathbf{A}}$ 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 nonoptimal local maximum, and this 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 algorithms to extract more information from the training samples. That is shown in the enhancement of their training and test fields classification accuracy using the ML classifier. This is the case for small or relative large numbers of training samples and classes. This is due to the fact that Projection Pursuit contains the properties that a high dimensional reduction algorithm should have as explained in the Introduction. It mitigates the difficulties of high dimensional data by making the computations in a lower dimensional 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.

Additional details about Projection Pursuit in this context and high dimensional spaces are available in [29]. Further, as multispectral sensor systems continue to develop and do so in the direction of producing higher dimensional data, analysis algorithms for such more complex data necessarily become more complex. This tends to limit the availability of such more powerful algorithms to practitioners who might need to apply them and to researchers who wish to build upon them. To overcome this problem, we began several years ago to construct an application program for personal computers which contains a basic capability to analyze multispectral data, and then as new algorithms emerge from our research program to add these new algorithms to the program. The program, called MultiSpec, and its documentation is made available without charge to anyone who wishes it via the world wide web at

http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/

The algorithms used in this paper are available in the Macintosh version of MultiSpec at that location. It is planned that Projection Pursuit, along with Discriminate Analysis and Decision Boundary Feature Extraction will be available in the Windows version as well in the coming months.

VIII. References

[1] L. O. Jimenez and D. A. Landgrebe, “Supervised Classification in High Dimensional Space: Geometrical, Statistical, and Asymptotical Properties of Multivariate Data,” *IEEE Transactions on System, Man, and Cybernetics*, to appear.

[2] 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.

[3] 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.

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

[5] 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.

[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973, pp. 95.

[7] 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.

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

[9] 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.

[10] J. A. Richards, *Remote Sensing Digital Image Analysis, An Introduction*. 2nd ed. New York: Springer-Verlag, 1993, pp. 216.

[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, pp. 206-209.

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

[13] 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.

[14] 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.

[15] 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.

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

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

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

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

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

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

[22] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," IEEE Transactions on Information Theory, Vol. IT-14, No. 1, January 1968.

[23] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* San Diego, California: Academic Press, Inc., 1990, pp. 99-109.

[24] 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.

[25] 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.

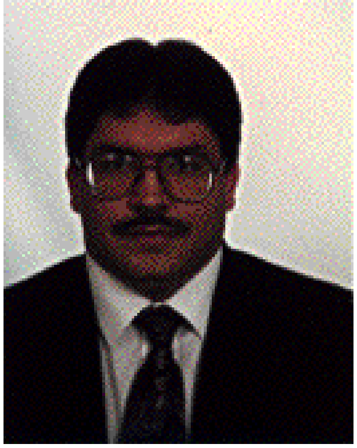
[26] S .M. Weiss and C.A. Kulikowski, *Computer Systems that Learn*, San Mateo California: Morgan Kaufmann Publishers, Inc., 1991, pp. 116-118.

[27] 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.

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

[29] L. Jimenez and D. A. Landgrebe, “High Dimensional Feature Reduction Via Projection Pursuit,” Technical Report TR-ECE 96-5 (PhD Thesis) School of Electrical & Computer Engineering, Purdue University, West Lafayette IN 47907-1285

Luis O. Jimenez



Dr. Luis O. Jimenez received the BSEE from University of Puerto Rico at Mayaguez, in 1989. He received his MSEE from University of Maryland at College Park in 1991 and his PhD from Purdue University in 1996. Currently he is an Assistant Professor of Electrical and Computer Engineering at the University of Puerto Rico, Mayaguez Campus. His research interest include Pattern Recognition, Remote Sensing, Feature Extraction, Artificial Intelligence, and Image Processing.

Dr. Jimenez is a associate member of the IEEE. He is also member of the Tau Beta Pi and Phi Kappa Phi honor societies.

David A. Landgrebe

Dr. Landgrebe holds the BSEE, MSEE, and PhD degrees from Purdue University. He is presently Professor of Electrical and Computer Engineering at Purdue University. His area of specialty in research is communication science and signal processing, especially as applied to Earth observational remote sensing. His contributions over the last 25 years in that field have related to the proper design from a signal processing point of view of multispectral imaging sensors, suitable spectral and spectral/spatial analysis algorithms, methods for designing and training classifier algorithms, and overall systems analysis. He was one of the originators of the multispectral approach to Earth observational remote sensing in the 1960's, was instrumental in the inclusion of the MSS on board Landsat 1, 2, and 3, and hosted and chaired the NASA meeting at which the bands and other key parameters were selected for the Thematic Mapper. He has been a member of a number of NASA and NRC advisory committees for this area since the 1960's.

He was President of the IEEE Geoscience and Remote Sensing Society for 1986 and 1987 and a member of its Administrative Committee from 1979 to 1990. He received that Society's Outstanding Service Award in 1988. He is a co-author of the text, *Remote Sensing: The Quantitative Approach*, and a contributor to the book, *Remote Sensing of Environment*, and the *ASP Manual of Remote Sensing (1st edition)*. He has been a member of the editorial board of the journal, *Remote Sensing of Environment*, since its inception.

Dr. Landgrebe is a Life Fellow of the Institute of Electrical and Electronic Engineers, a Fellow of the American Society of Photogrammetry and Remote Sensing, and a member of the American Society for Engineering Education, as well as Eta Kappa Nu, Tau Beta Pi, and Sigma Xi honor societies. He received the NASA Exceptional Scientific Achievement Medal in 1973 for his work in the field of machine analysis methods for remotely sensed Earth observational data. He was the 1990 recipient of the William T. Pecora Award, presented by NASA and the U.S. Department of Interior, for contributions to the field of remote sensing. He was the 1992 recipient of the IEEE Geoscience and Remote Sensing Society's Distinguished Achievement Award.