Hyperspectral Data Analysis and Supervised Feature Reduction Via Projection Pursuit

Medical Image Analysis

Luis O. Jimenez and David A. Landgrebe

Ion Marqués, Grupo de Inteligencia Computacional, UPV/EHU

09-03-2012

Publishing

- Journal JCR: 2.485 (2ª en Remote Sensing).
- Manuscript received July 1997.
- Revised October 1998.
- Published November 1999.

Overview

Projection Pursuit

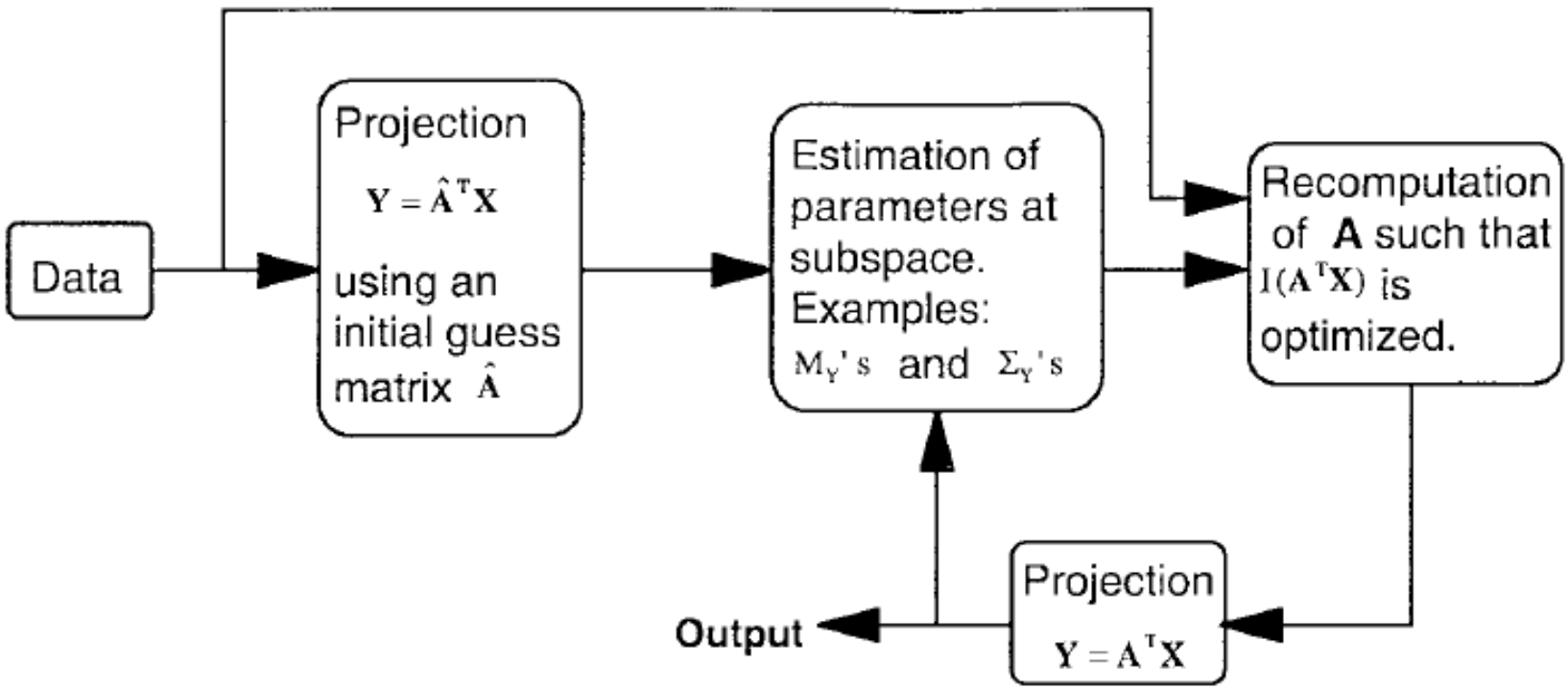
"... the numerical optimization of a criterion in search of the most interesting low-dimensional linear projection of a high-dimensional data cloud."
So, if we have initial data X, dimensionally reduced data Y and a parametric orthonormal matrix A where $Y = A^T X$,
PP is the method that computes A, optimizing the porjection index $I(A^T X)$.

Parametric Projection Pursuit

Why parametric and supervised?

- 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 a priori knowledge in the form of labeled samples, the unsupervised indices are not able to exploit such information.
- Some authors have suggested that 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.

Parametric Projection Pursuit



Organization of the PP process.

Parametric Projection Pursuit

- 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-order and second-order statistics.
- 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:

Parametric Projection Pursuit

$$I(\boldsymbol{A}^T\boldsymbol{X}) = \min_{i \in C} \left\{ \frac{1}{8} (\boldsymbol{M}^i_{2Y} - \boldsymbol{M}^i_{1Y})^T \left(\frac{\Sigma^i_{1Y} + \Sigma^i_{2Y}}{2} \right)^{-1} \right.$$

$$\left. \cdot (M^i_{2Y} - M^i_{1Y}) + \frac{1}{2} \ln \left[\frac{\left| \frac{\Sigma^i_{1Y} + \Sigma^i_{2Y}}{2} \right|}{\sqrt{|\Sigma^i_{1Y}||\Sigma^i_{2Y}|}} \right] \right\}. \tag{2}$$

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

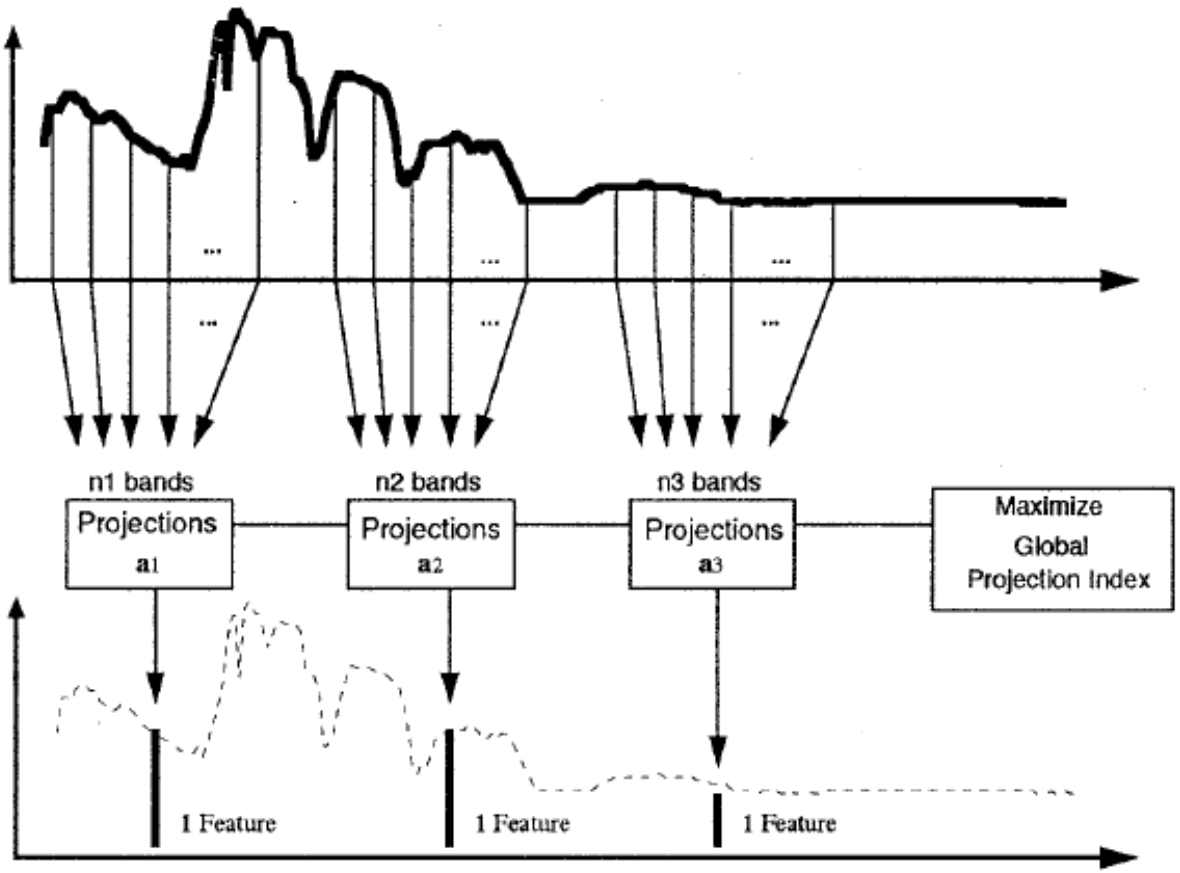
$$C = \frac{L!}{2!(L-2)!}. \tag{3}$$

M^i_{jY} is the mean vector and Σ^i_{jY} the covariance matrix of j-th class in the projected subspace Y for the combination of pair of classes i.

Sequential Parametric PP

They want to ensure linear independence on A and reduce dimensionality, so:

- A is definded as $A = [A_1 \ldots A_{Col-1} A_{Col-2}]$. Every column of A will be filled with zeros, except at a group of adjacent positions $A_i = [0..0\, \mathbf{a}_i\, 0..0]$. A_i will combine some adjacent bands, the columns must be orthogonals and no two A_i's may have nonzeros at the same locations. In other words, for $i \neq j;\ A_i^T \cdot A_j = 0$.



Sequential Parametric PP

1) An initial choice for every $\boldsymbol{a}_i$, for every group of adjacent bands, is made and stored.
2) Maintaining the rest of the $\boldsymbol{a}_i$'s constant, compute $\boldsymbol{a}_1$ (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 in which $\boldsymbol{a}_i$ is calculated, optimizing against the global Bhattacharyya distance while maintaining the $\boldsymbol{a}_j$'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 $\boldsymbol{a}_j$'s sequentially) until the maximization ceases increasing significantly. The significant increment is relative to each iteration. If one iteration (steps 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.

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 is divided into two stages:

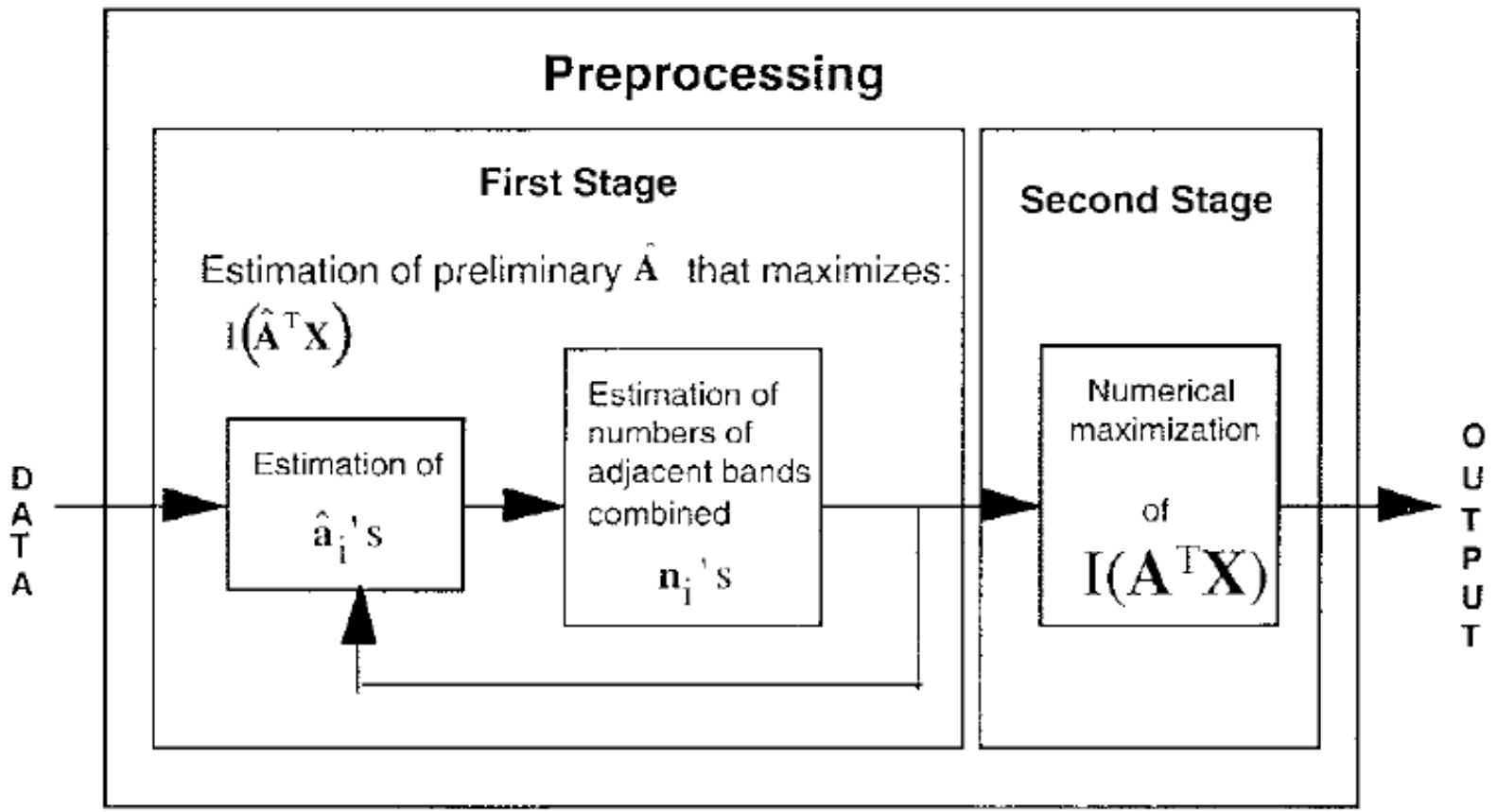


Fig. 6. Preprocessing block.

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

$$\text{Bhatt}_M = \frac{1}{8}(\boldsymbol{M}_2 - \boldsymbol{M}_1)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\boldsymbol{M}_2 - \boldsymbol{M}_1). \quad (4)$$

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

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

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

$$\text{Bhatt}_C = \tfrac{1}{4}\{\ln|\Sigma_2^{-1}\Sigma_1 + \Sigma_1^{-1}\Sigma_2 + 2\boldsymbol{I}| - n\ln 4\} \quad (6)$$

The mean-difference portion (Bhatt_M) is maximized by the vector $\boldsymbol{a}_{M\max}$ [23, pp. 455–457]

$$\boldsymbol{a}_{M\max} = (\boldsymbol{M}_2 - \boldsymbol{M}_1)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1}. \quad (7)$$

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

$$\Lambda = \Sigma_2^{-1}\Sigma_1. \quad (8)$$

Preprocessing block stages and the initial conditions

The vector that maximizes Bhatt_C ($\boldsymbol{a}_{C\,\max}$) is the eigenvector of Λ that corresponds to the eigenvalue that maximizes the function $f(\lambda_i)$

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

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

$$J(d) = \ln \left| (\boldsymbol{a}^T \Sigma_2 \boldsymbol{a})^{-1} \boldsymbol{a}^T \Sigma_1 \boldsymbol{a} + (\boldsymbol{a}^T \Sigma_1 \boldsymbol{a})^{-1} \boldsymbol{a}^T \Sigma_1 \boldsymbol{a}_2 \boldsymbol{I}_d \right| \tag{10}$$

where d is the dimensionality of the data. It follows that $\boldsymbol{a}_{C\,\max}$ maximizes Bhatt_C.

Preprocessing block stages and the initial conditions

The process of building the initial choice of matrix $\hat{\boldsymbol{A}}$ from the estimated $\hat{\boldsymbol{a}}_i$ 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 PP algorithm. The procedure is as follows.

1) Choose one $\hat{\boldsymbol{a}}_i$ from each bank for every group of n_i-adjacent bands. Every $\hat{\boldsymbol{a}}_i$ belongs to the proper place in the ith column of $\hat{\boldsymbol{A}}$ that corresponds to the ith group of adjacent bands.
2) Maintaining the rest of the $\hat{\boldsymbol{a}}_i$'s constant, choose the $\hat{\boldsymbol{a}}_i$ from the first bank of samples that maximizes the global-projection index.
3) Repeat the procedure for each group, such that the $\hat{\boldsymbol{a}}_i$ is chosen from the ith bank of samples. Meanwhile, the $\hat{\boldsymbol{a}}_j$'s for $i \neq j$ will be held constant.
4) Once the last $\hat{\boldsymbol{a}}_i$ is chosen, repeat the process from step 2 until the maximization converges or stops increasing significantly.

Estimating number of adjacent bands n_i

- They use decision trees estimate the value of the n_i's.
- They present top-down, botton-up and hybrid heuristic methods of decision-tree classifiers .
- They do nos clearly state which they used.

Experiment 1

METHODS:

- **DA 100-20**: The multispectral data was reduced in dimensionality from 200 dimensions. Using DA at full dimensionality, the data was reduced from 100 bands (one in every two bands from the original 200) to a 20-dimensional subspace (20-D) . From the original number of bands, 100 were used because of the limited number of training samples (179).
- **PP**: Here is an iterative sequential PP with only a numerical-optimization stage applied to the data in order to reduce the dimensionality, maximizing the minimum-Bhattacharyya distance among the classes. This mehtods does not use the decision tree to find the number of bands required to be combined. 10 bands combined per group.
- **PP-Opt**: Optimum PP with the first stage, which estimates matrix and the numerical-optimization stage, used to project from 200 to a 20-D subspace . The algorithm estimates the dimensionality of the data as 20.
- **PP-Opt-FS**: This 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

Experiment 1

DATABASE: **AVIRIS**
CLASS DISTRIBUTION:

TABLE I

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

TABLE II
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

Experiment 1

TABLE III
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 IV
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

Experiment 1

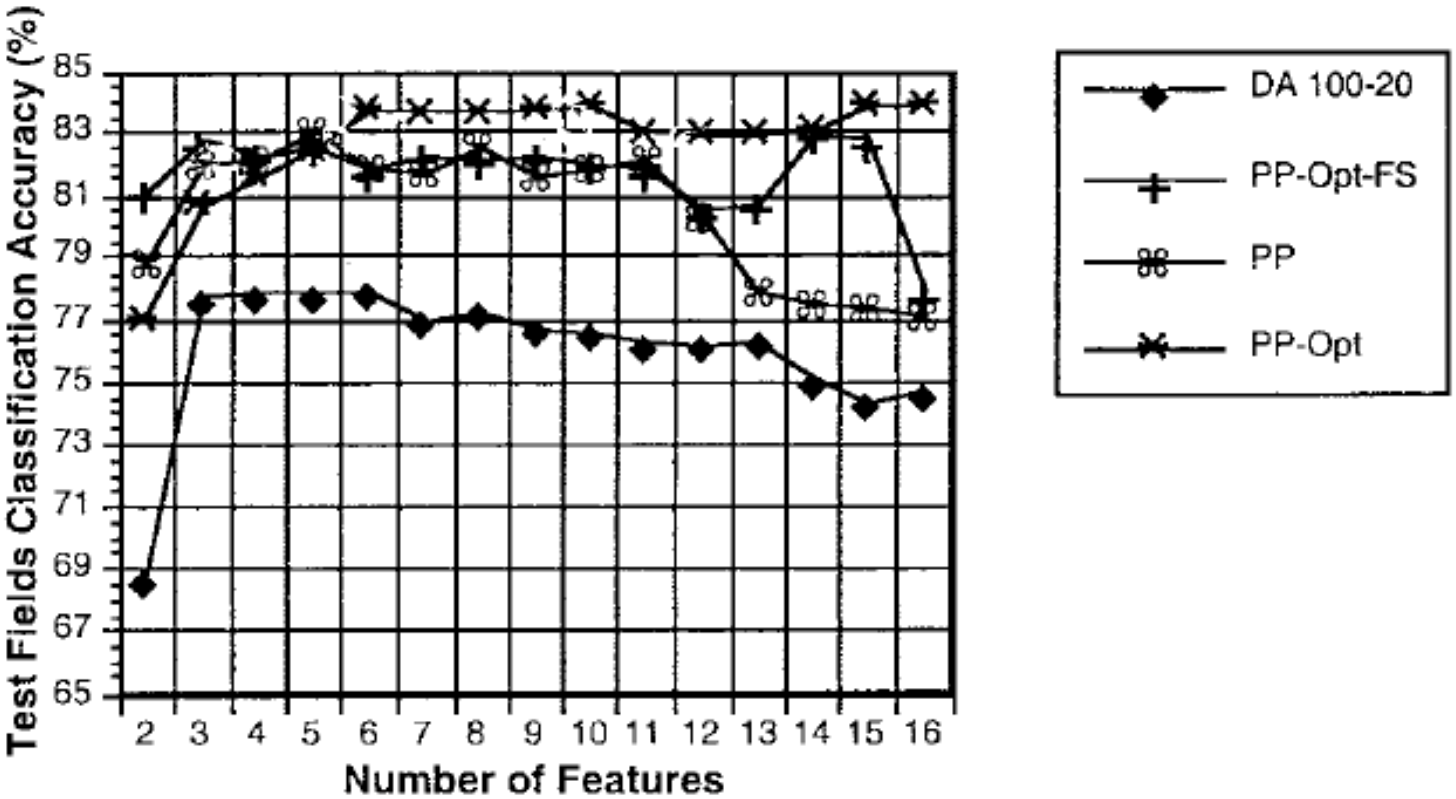


Fig. 8. Test-fields classification accuracy comparison between direct use of DA (DA 100-20) and the use of DA after different methods based on PP, PP-Opt, and PP-Opt-FS methods for an ML classifier.

Experiment 1

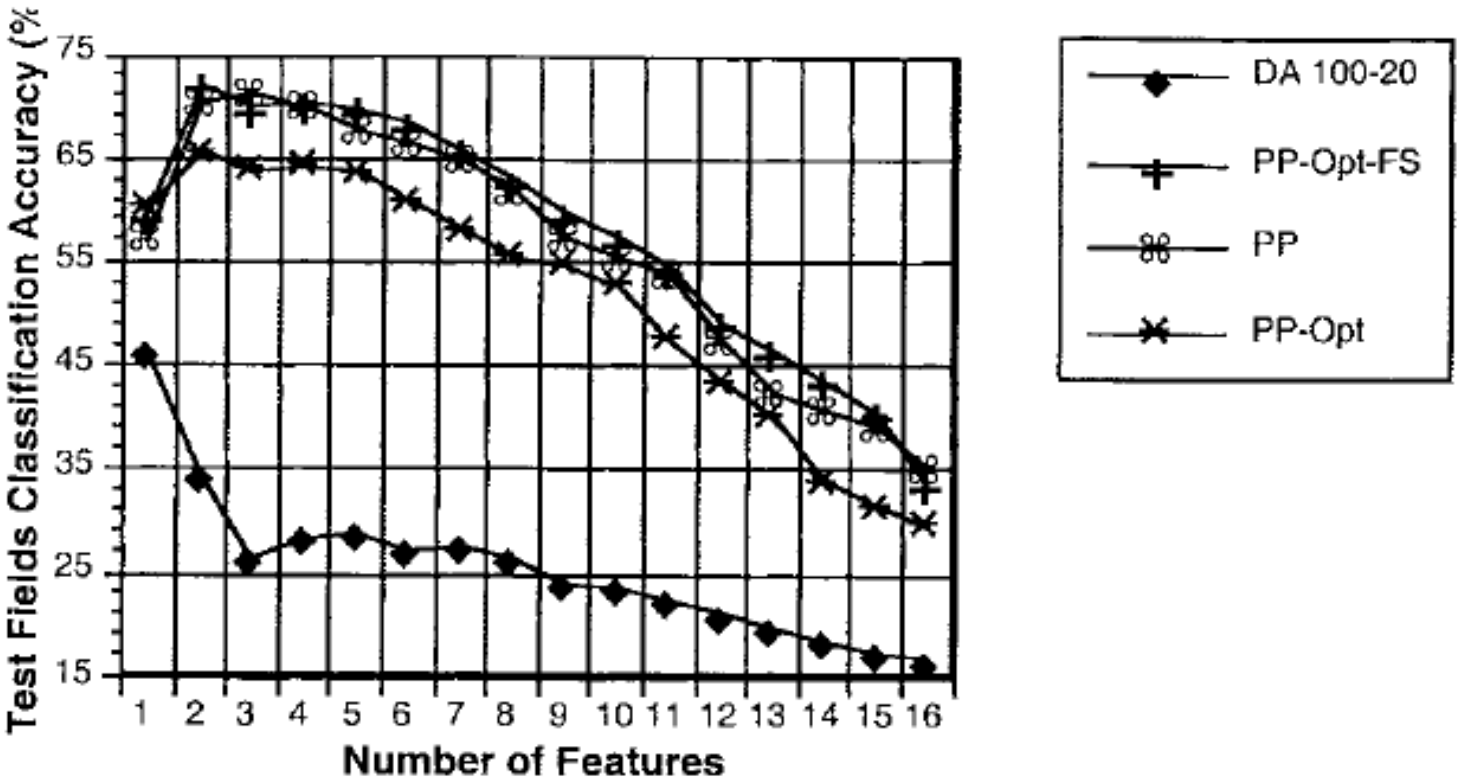


Fig. 9. Test-fields classification accuracy comparison between direct use of DA (DA 100-20) and the use of DA after different methods based on PP, PP-Opt, and PP-Opt-FS methods for an ML with 2% threshold.

Experiment 2

Methods: DBFE, DAFE, PP-Opt and PP-Opt-FS

DAFE:

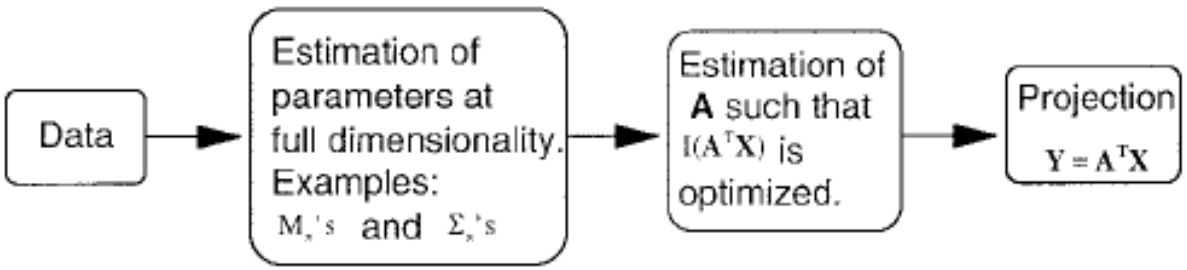


Fig. 2. DAFE process order.

DBFE stands for Decision Boundary Feature Extraction.

Experiment 2

DATABASE: **AVIRIS**
CLASS DISTRIBUTION:

TABLE V

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

TABLE VI

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

Experiment 2

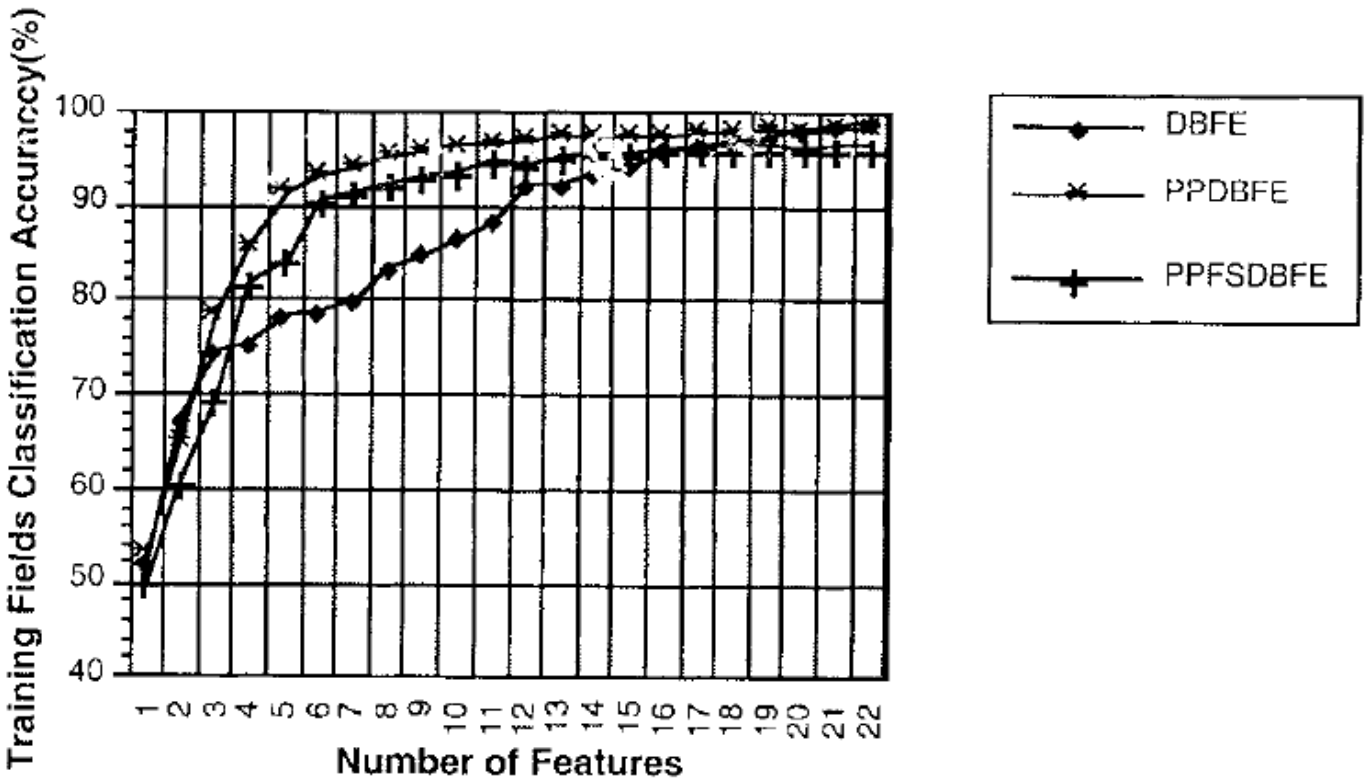


Fig. 10. Training-fields classification accuracy comparison between direct use of DBFE and the use of DBFE after different methods based on PPDBFE and PPFSDBFE for an ML classifier.

Experiment 2

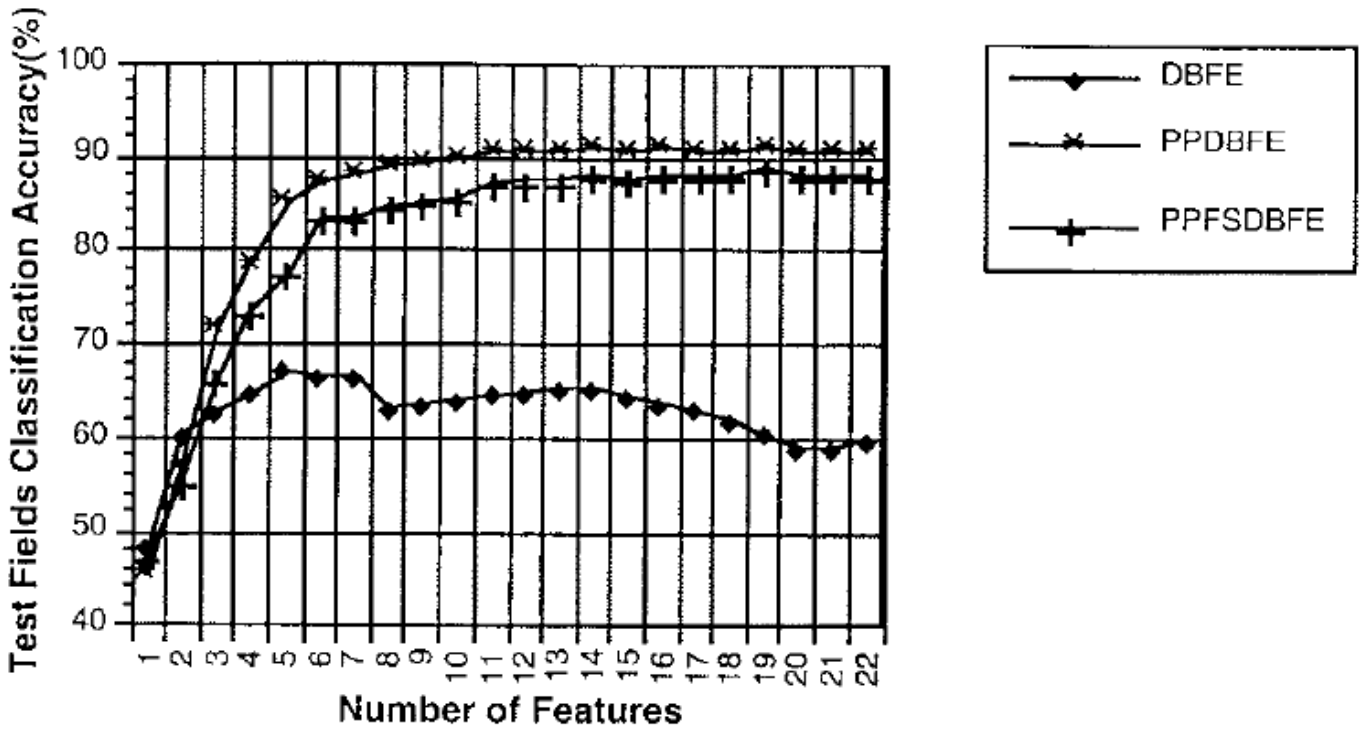


Fig. 11. Test-fields classification accuracy comparison between DBFE and the use of DBFE after different methods based on PPDBFE and PPFSDBFE for an ML classifier.

Experiments 2

There are more graphics with similar results. (Very bad quality, welcome to 1999).