A Robust Classification Procedure Based on Mixture Classifier and Nonparametric Weighted Feature Extraction

Bor-Chen Kuo

Department of Mathematics Education

National Taichung Teachers College, Taichung, Taiwan 403

Email: kbc@mail.ntctc.edu.tw

David A. Landgrebe

School of Electrical and Computer Engineering

Purdue University, West Lafayette, Indiana 47907-1285

Email: landgreb@ecn.purdue.edu

A Robust Classification Procedure Based on a Mixture Classifier using Nonparametric Weighted Feature Extraction[1]

Bor-Chen Kuo, Member, IEEE and David A. Landgrebe, Life Fellow, IEEE

Abstract

There are many factors to consider in carrying out a hyperspectral data classification, perhaps chief among them are class training sample size, dimensionality, and distribution separability. The intent of this study is to design a classification procedure which is robust and maximally effective, but which provides the analyst with significant assists, thus simplifying the analyst's task. The result is a quadratic mixture classifier based on Mixed-LOOC2 regularized discriminant analysis and Nonparametric Weighted Feature Extraction. This procedure has the advantage of providing improved classification accuracy compared to typical previous methods but requires minimal need to consider the factors mentioned above. Experimental results demonstrating these properties are presented.

[1] The work described in this paper was sponsored in part by the National Imagery and Mapping Agency under grant NMA 201-01-C-0023

1 Introduction

Hyperspectral data holds great potential compared to more conventional multispectral data in terms of its information-bearing properties. However, if this potential is to be realized, two areas of advancement are required. On the one hand, sound, fundamentally based data analysis technology must be defined that is quantifiably optimal in the sense of its information extraction capabilities. On the other, such optimal analysis methods must be straightforward to apply.

Among the ways to approach hyperspectral data analysis, a useful processing model that has evolved in the last several years [1] is shown schematically in Figure 1. Research has shown that achieving high precision in modeling the desired classes quantitatively is the critical element to the most effective analysis. Given the availability of data (box 1), the process begins by the analyst specifying what classes are desired, usually by labeling training samples for each class

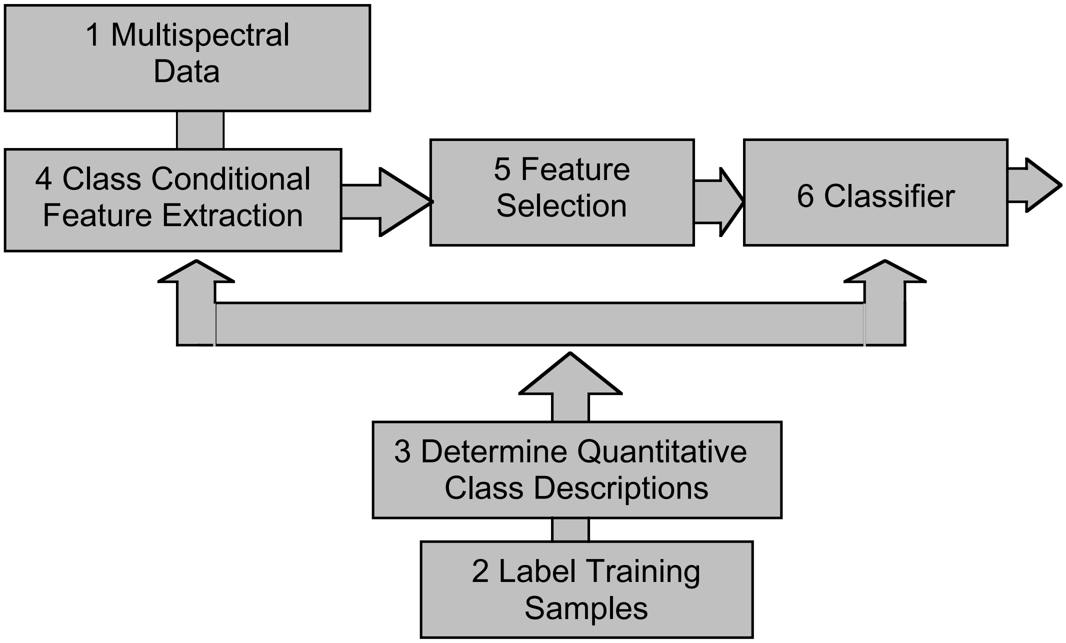
1 Multispectral Data
4 Class Conditional Feature Extraction
5 Feature Selection
6 Classifier
3 Determine Quantitative Class Descriptions
2 Label Training Samples

Figure 1. A schematic diagram for a hyperspectral data analysis procedure.

(box 2). New elements to class modeling that have proven important in the case of high dimensional data are those indicated by boxes in the diagram marked 3 and 4. These are the focus of this work and will be discussed in more detail shortly, however the reason for their importance in this context is as follows. Classification techniques in pattern recognition typically assume that there are enough training samples available to obtain reasonably accurate class descriptions in quantitative form. Unfortunately, the number of training samples required to train a classifier for high dimensional data is much greater than that required for conventional data, and gathering these training samples can be difficult and expensive. Therefore, the assumption that enough training samples are available to accurately estimate the class quantitative description is frequently not satisfied for high dimensional data. Small training sets usually cause Hughes phenomenon [16] and singularity problems. There are several ways to overcome these problems. In [2], these techniques are categorized into three groups:

a. Dimensionality reduction by feature extraction or feature selection.

b. Regularization of sample covariance matrix (e.g. [3], [4], [9]).

c. Structurization of a true covariance matrix described by a small number of parameters [2].

There are many types of classification algorithms. Perhaps the most common is the quadratic maximum likelihood classifier. Sometimes the data distribution is not so simple that using a normal distribution to describe it is enough. The Gaussian mixture density, which models the density as the sum of one or more weighted Gaussian components, is a compromise between a simple single Gaussian and non-parametric densities. It allows more flexibility than the single Gaussian density, yet requires fewer parameters to be estimated than non-parametric densities. Most methods in this area usually assume that, if one class can be divided into several normally distributed subgroups, then the sample size of each subgroup should not be less than the dimensionality.

The purpose of this paper is to design a classification procedure using a mixture classifier based on Mixed-LOOC2 and nonparametric weighted feature extraction (NWFE) [9], which can mitigate the effects of Hughes phenomenon and covariance singularity and is suitable for complex and high dimensional data.

2 Previous Works

2.1 Mixed-LOOC2

Many regularized covariance estimators have been developed to solve the singularity problem, for example: RDA [3], LOOC [6], BLOOC [7], and Mixed-LOOC2 [9]. In [9] it is shown that Mixed –LOOC2 has the advantages of both LOOC and BLOOC when it is applied to the uni-mode Gaussian quadratic classifier. The procedure, DAFE (or LDA) based on Mixed-LOOC2, overcomes the shortcoming that DAFE cannot be used when the training sample size is less than the dimensionality, and can provide higher accuracy when the sample size is limited.

The Mixed-LOOC2 was proposed as the following form:

$$\Sigma_i(\beta_i) = \beta_i \mathrm{A} + (1-\beta_i)\mathrm{B}$$

where $\mathrm{A} = \frac{tr(S_i)}{p}I,\ diag(S_i),\ S_i,\ \frac{tr(S)}{p}I,\ diag(S),$ or S , $\mathrm{B} = S_i,$ or $diag(S)$ and β_i is close to 1. S_i is the sample estimate of the covariance matrix for class i, and S is the covariance estimated from the training samples of all classes. B= S_i or *diag(S)* is chosen because if a class sample size is large, S_i will be a better choice. If the total training sample size is less than the dimensionality, then the common (pooled) covariance S is singular, but otherwise it has much less estimation error than S_i. For reducing estimation error and avoiding singularity, *diag(S)* will be a good choice. The selection criteria is the log leave-one-out likelihood function:

$$LOOL_i(\beta_i) = \frac{1}{N_i}\sum_{k=1}^{N_i} \ln[f(x_k | m_{i/k}, \Sigma_{i/k}(\beta_i))]$$

where f is the probability density function of each class.

2.2 Quadratic Mixture Classifiers

In order to model non-Gaussian classes, consider the quadratic mixture density, which is the weighted summation of K Gaussian density functions:

$$p(x) = \sum_{k=1}^{K} \alpha_k f(x \mid m_k, \Sigma_k)$$

where $f(x \mid m_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_k|}} \exp[-\frac{1}{2}(x - m_k)^T \Sigma_k^{-1}(x - m_k)]$, and p is the dimensionality of data. Each term f in the summation is called a component of the mixture density. The weights α_k, which must sum to unity, are *a priori* probabilities of the components. In practice the parameters of the mixture density function (K, α_κ, m_k, and Σ_κ for k = 1, 2, …K)) are usually unknown and must be estimated from the training samples. Multimode classes can be represented by a mixture density with one or more components representing each mode. Since the covariance matrix of each component should be invertible, ordinarily the sample size of each component should not be less than the dimensionality of the data. In this paper, the new mixture classifier will relieve this limitation.

There are two steps to design a quadratic mixture classifier. The first is parameter estimation and the second is model selection. In this study, NM (nearest means or K-mean) clustering and EM (expectation-maximization) clustering are used in the parameter estimation part. There are many indices for model selection. In this research, only the performances of AIC, BIC, NEC, and ICLBIC, described below, are tested.

In the multivariate mixture model, data $x_1,...,x_n$ in $\mathbf{R}^p$ are assumed to be a sample from a probability distribution with density

$$p(x) = \sum_{k=1}^{K} \alpha_k f(x, a_k)$$

where the α_k's are the mixing proportions $(0 < \alpha_k < 1)$ for all $k = 1,...,\mathrm{K}$ and $\Sigma_{k=1}^{K} \alpha_k = 1$. $f(x, a_k)$ denotes the p-dimensional Gaussian density with mean m_k and covariance matrix Σ_k with $a_k = (m_k, \Sigma_k)$. The maximized log likelihood of $\Psi = ((\alpha_1, a_1),...,(\alpha_K, a_K))$ for the sample $x_1,...,x_n$ is denoted

$$L(\Psi) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} \alpha_k f(x_i \mid m_k, \Sigma_k)]$$

with α_k and a_k denoting the maximum likelihood estimates of the corresponding parameters.

The Akaike information criterion (AIC; [10]) is defined as

$$AIC(\Psi) = -2L(\Psi) + 2v(\Psi)$$

where $v(\Psi)$ is the number of free parameters in the mixture model Ψ.

The Bayesian information criterion (BIC) [11] is given by

$$BIC(\Psi) = -2L(\Psi) + v(\Psi)\log n.$$

Let

$$t_{ik} = \frac{\alpha_k f(x_i \mid a_k)}{\sum_{j=1}^{k} \alpha_j f(x_i \mid a_j)}$$

be the estimated conditional probability that x_i arises from the k^{th} mixture component. The entropy is defined as

$$E(\Psi) = -\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik} \log t_{ik} \geq 0.$$

The entropy of the classification matrix t gives rise to several classification criteria [12], which are $E(\Psi)$, its normalized version

$$NEC(\Psi) = \frac{E(\Psi)}{L(\Psi) - L_1(\Psi)},$$

where $L_1(\Psi)$ denotes the maximized log-likelihood for a single Gaussian distribution.

And ICL-BIC in [13] is

$$ICL - BIC(\Psi) = -2L(\Psi) + 2E(\Psi) + v(\Psi)\log n$$

It was observed that AIC is order inconsistent and tends to overfit models [12]. BIC based on the Bayes factor could improve the overfitting problem by using the sample size in the penalized term. NEC and ICL-BIC are attempted to overcome the shortcomings of classification likelihood criterion and BIC [13].

2.3 Nonparametric Weighted Feature Extraction (NWFE)[2]

Discriminant Analysis Feature Extraction (DAFE) or Linear Discriminant Analysis (LDA) is often used for dimension reduction in classification problems. The advantage of DAFE is that it is distribution-free but there are three major disadvantages in DAFE. One is that it works well only if the distributions of classes are normal-like distributions [8]. When the distributions of classes are nonnormal-like or multi-modal mixture distributions, the performance of DAFE is not satisfactory. The second disadvantage of DAFE is the rank of the within-scatter matrixS_b is number of classes (L) –1, so generally only L-1 features can be extracted. From [8], we know that unless a posterior probability function is specified, L–1 features are suboptimal in a Bayes sense, although they are optimal based on the chosen criterion. In real situations, the data distributions are often complicated and not normal-like, therefore only using L-1 features is not usually sufficient for real data. The third limitation is that if the within-class covariance is singular, which often occurs in high dimensional problems, DAFE will have a poor performance on classification. NWFE is developed to solve those problems [9]. The results of simulated and

[2] NWFE has been implemented into MultiSpec© which is available at http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/.

real data experiments show that the performance of NWFE is better than those of DAFE, aPAC-LDR [14], and NDA [15].

The main ideas of NWFE are putting different weights on every sample to compute the “local means” and defining new nonparametric between-class and within-class scatter matrices to obtain more features. In NWFE, the nonparametric between-class scatter matrix is defined as

$$S_b = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j \neq i}}^{L} \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,j)}}{n_i} (x_k^{(i)} - M_j(x_k^{(i)}))(x_k^{(i)} - M_j(x_k^{(i)}))^T$$

where $x_k^{(i)}$ refers to the k-th sample from class i, and P_i is the prior probability of class i.

The scatter matrix weight $\lambda_k^{(i,j)}$ is a function of $x_k^{(i)}$ and $M_j(x_k^{(i)})$, and defined as:

$$\lambda_k^{(i,j)} = \frac{dist(x_k^{(i)}, M_j(x_k^{(i)}))^{-1}}{\sum_{l=1}^{n_j} dist(x_l^{(i)}, M_j(x_l^{(i)}))^{-1}},$$ where $dist(a,b)$ means the Euclidean distance from a to b.

If the distance between $x_k^{(i)}$ and $M_j(x_k^{(i)})$ is small then its weight $\lambda_k^{(i,j)}$ will be close to 1; otherwise, $\lambda_k^{(i,j)}$ will be close to 0 and sum of all $\lambda_k^{(i,j)}$ for class *i* is 1.

$M_j(x_k^{(i)})$ is the local mean of $x_k^{(i)}$ in the class j and defined as:

$$M_j(x_k^{(i)}) = \sum_{l=1}^{n_j} w_{kl}^{(i,j)} x_l^{(j)}, \text{ where } w_{kl}^{(i,j)} = \frac{dist(x_k^{(i)}, x_l^{(j)})^{-1}}{\sum_{l=1}^{n_j} dist(x_k^{(i)}, x_l^{(j)})^{-1}}.$$

The weight $w_{kl}^{(i,j)}$ for computing local means is a function of $x_k^{(i)}$ and $x_l^{(j)}$ If the distance between $x_k^{(i)}$ and $M_j(x_k^{(i)})$ is small then its weight $\lambda_k^{(i,j)}$ will be close to 1; otherwise, $\lambda_k^{(i,j)}$ will be close to 0 and sum of total $\lambda_k^{(i,j)}$ for class *i* is 1.

The nonparametric within-class scatter matrix is defined as

$$S_W = \sum_{i=1}^{L} P_i \sum_{k=1}^{n_i} \frac{\lambda_k^{(i,i)}}{n_i} (x_k^{(i)} - M_i(x_k^{(i)}))(x_k^{(i)} - M_i(x_k^{(i)}))^T$$

The optimal features are determined by optimizing the criteria given by

$$J_{NWFE} = tr(S_W^{-1} S_b)$$

To reduce the effect of the cross products of between-class distances and prevent the singularity, we will regularize S_W by

$$S_W = 0.5 S_W + 0.5 \mathrm{diag}(S_W)$$

Finally the NWFE algorithm is

1. Compute the distances between each pair of sample points and form the distance matrix.

2. Compute $w_l^{(i,j)}$ using the distance matrix

3. Use $w_l^{(i,j)}$ to compute local means $M_j(x_k^{(i)})$

4. Compute scatter matrix weight $\lambda_k^{(i,j)}$.

5. Compute S_b and S_w.

6. Select the m eigenvectors of $S_w^{-1}S_b$, $\psi_1, \psi_2, \cdots, \psi_m$, which correspond to the m largest eigenvalues to form the transformation matrix $A_m = [\psi_1, \psi_2, \cdots, \psi_m]$

3 Gaussian Mixture Classifier Based on Mixed-LOOC2

3.1 Mixture Classifier Using Mixed-LOOC2 and Nearest Means Clustering

The algorithm of a mixture classifier using Mixed-LOOC2 and nearest means (NM) clustering is

Step 1. Compute Mixed-LOOC2 of each class and for each class, use nearest means clustering to find the components.

Step 2. Compute Mixed-LOOC2 of each component in the classes.

Step 3. Compute the model selection index using Mixed-LOOC2 to replace ML covariance estimate.

Step 4. If the number of components in classes is 1, then use the Mixed-LOOC2 of this class as its covariance estimator.

Step 5. Compute the mixture density function to form the Bayesian mixture classifier.

3.2 Mixture Classifier Using Mixed-LOOC2 and EM clustering

The algorithm of a mixture classifier using Mixed-LOOC2 and EM clustering is

Step 1 Compute Mixed-LOOC2 of each class.

Step 2 For each class, use EM clustering to find the components. But, in the estimating covariance step of EM clustering, the ML estimator of each component should be replaced by Mixed-LOOC2.

Step 3 Compute the model selection index using Mixed-LOOC2 to replace ML covariance estimate.

Step 4 If the number of components in classes is 1, then using the Mixed-LOOC2 of this class as its covariance estimator.

4 Simulated and Real Data Experiments for Mixture Classifier Based on Mixed-LOOC2

4.1 Simulation Data Experiment Design

In simulation experiments, the performances of mixture classifiers based on NM and EM clustering with model selection indices AIC, BIC, NEC, ICL-BIC and their Mixed-LOOC2 versions are compared.

In classification problems, there are two kinds of mixture situations. One is the components of each class are grouped together and do not mix significantly with those of other classes as seen in Figure 2(a). The other is that

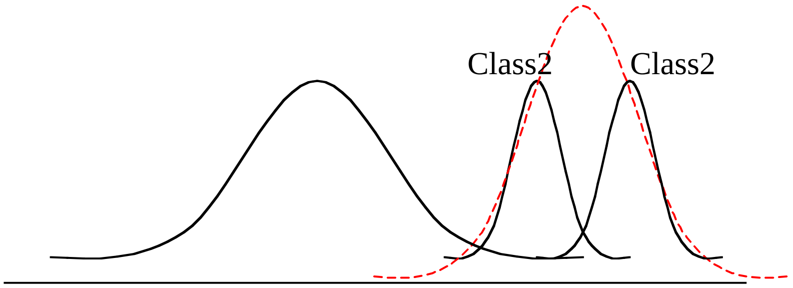

Figure 2(a) Class 1 is not between subcomponents of class 2.

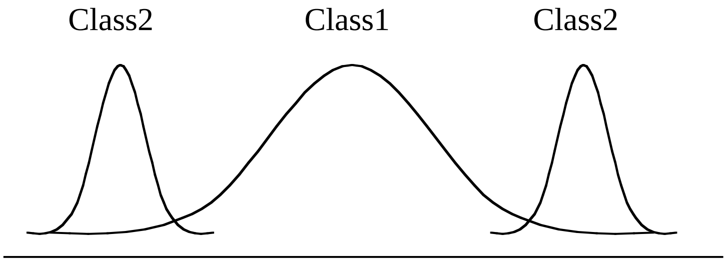

Figure 2(b) Class 1 is between subcomponents of class 2 the components of different classes mix together, as in Figure 2(b). In first case, the mixture classifier may have performance similar to a simple quadratic classifier if the class sample sizes are large enough. But when the class sample is small then the performance of a mixture classifier may not be as good as that of simple quadratic classifier due to estimation error. In second case, the mixture classifier would be expected to do a better job when the class sample sizes are large enough, but if class sample is small then the mixture classifier may have more severe problems.

The simulation study will focus on the second situation and try to find out which combination of parameter estimation and model selection will give a better result. The class sample sizes and the class mean vectors and covariance matrices of simulated data are in Table 1(a). The clustering algorithm used in experiments 1 and 2 is NM clustering and that used in experiments 3 and 4 is EM clustering. Five different dimensionality (2,4,10,20,60) and three different class sample sizes are tested. In each situation (Table 1(b)), 10 random training and testing data sets are generated for computing the accuracies of algorithms, and the standard deviations of the accuracies.

Table 1(a) The class mean vectors and covariance matrices of simulated data

		class 1	class 2	
Dim=2,4,10,20,60		component 1	component 1	component 2
Mean Vector		[0,0,…,0]	[1,1, …,1]	[-1,-1, …,-1]
Covariance	Exp 1 and 3	I	0.1I	0.1I
	Exp 2 and 4	I	I	I
Training Class Sample Size(Ni)		30, 60, 300	15,30,150	15,30,150
Testing Class Sample Size		30, 60, 300	15,30,150	15,30,150

Table 1(b) The dimensionality and class sample size of simulated data

Situation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dim	2	4	10	20	60	2	4	10	20	60	2	4	10	20	60
Ni	30	60	300	30	60	300	30	60	300	30	60	300	30	60	300

4.2 Real Data Experiment Design

Hyperspectral data from the Washington, DC Mall is used in real data experiments, and the better clustering algorithm, chosen from the results of simulation studies, is used. Two different class sample sizes (20 and 100) and two different dimensionalities (20 and 7) are used in Experiment 5. There are 191 bands in the DC Mall image data, and every 10-th band and 30-th band, which begins from the first one, are selected for the 20 and 7 bands cases. At each situation, 10 random training and testing data sets are generated for computing the testing sample accuracies of algorithms, and the number of subcomponents in each class.

5. Experiment Results

For convenience, denote the mixture classifier built on the original model selection index as the index itself (for example: AIC) and the mixture classifier built on the model selection index based on Mixed-LOOC2 as the index itself with a “Mix” suffix in tables and figures.

5.1 Simulation Experiment Results

The partial results of experiments 1 to 4 are displayed in tables 2(a), (b), (c), (d) and figures 3(a), and (b). Detailed results are in [9]. The results displayed in the figures are the accuracies using BIC_Mix in situations, 1 to 15 (in table 1(b)). They show that

1. Generally speaking, the mixture classifier BIC_Mix gave better performance than the others.

2. The shadowed parts in the tables indicate those cases that the performance of the mixture classifier BIC_Mix is significantly better than that of the 1-mode quadratic

classifier. In those unmarked situations, these two classifiers have equivalent performances.

3. From tables 2(a), (b), (c), (d), the performance of the mixture classifier using NM clustering was better than that of the mixture classifier using EM clustering.

4. The tables 2(a) and (b) (NM cases) show that if the subcomponents are well separated (I-0.1I case) then mixture classifiers (with/without using Mixed-LOOC2) have advantages in low dimensionality situations. When the dimensionality goes up, only the mixture classifiers using Mixed-LOOC2 have similar results to a simple quadratic classifier. Those not using Mixed-LOOC2 yield poorer results due to estimation error increasing. If the subcomponents are well separated (I-I case) then increasing the dimensionality will help the mixture classifiers using Mixed-LOOC2 to obtain better performance but will reduce the accuracy of those not using Mixed-LOOC2.

Table 2(a) Accuracies of experiment 1 (I-0.1I case) using NM clustering

			Model Selection								
Situation	Dimensionality	Sample Size	1 mode	AIC	AIC_Mix	BIC	BIC_Mix	NEC	NEC_Mix	ICLBIC	ICLBIC_Mix
1	2	30	0.7333	0.8333	0.8567	0.8433	0.8567	0.7383	0.7583	0.735	0.7417
2	2	60	0.7742	0.8617	0.8608	0.8617	0.8608	0.8042	0.8233	0.7708	0.7742
3	2	300	0.7758	0.8788	0.88	0.8788	0.88	0.8717	0.8702	0.8717	0.869
4	4	30	0.9167	0.83	0.9617	0.8333	0.9617	0.9167	0.9183	0.9167	0.89
5	4	60	0.9158	0.9408	0.9625	0.9475	0.9625	0.9092	0.9325	0.9167	0.9142
6	4	300	0.9225	0.968	0.9703	0.968	0.9703	0.9655	0.9663	0.9655	0.9668
7	10	30	0.9683	0.7233	0.9017	0.755	0.9617	0.9683	0.9517	0.9683	0.95
8	10	60	0.99	0.8075	1	0.8075	0.99	0.99	0.99	0.99	0.99
9	10	300	0.9945	0.9995	0.9997	0.9995	0.9997	0.9975	0.9997	0.9945	0.9947
10	20	30	0.945	0.7567	0.985	0.7567	0.97	0.945	0.97	0.71	0.74
11	20	60	0.9967	0.6892	0.9933	0.7342	0.9933	0.9967	0.9933	0.8975	0.8958
12	20	300	1	0.9228	1	0.9228	1	0.9995	1	1	1
13	60	30	0.5	0.5	0.9983	0.5	1	0.5	0.9983	0.5	0.9983
14	60	60	0.5	0.5	0.9992	0.5	1	0.5	0.9992	0.5	0.9992
15	60	300	1	1	1	1	1	1	1	1	1

Table 2(b) Accuracies of experiment 2 (I-I case) using NM clustering

			Model Selection								
Situation	Dimensionality	Sample Size	1 mode	AIC	AIC_Mix	BIC	BIC_Mix	NEC	NEC_Mix	ICLBIC	ICLBIC_Mix
1	2	30	0.6333	0.6233	0.645	0.6333	0.6467	0.6033	0.64	0.6333	0.6467
2	2	60	0.6575	0.6583	0.6608	0.6575	0.6575	0.6658	0.6608	0.6575	0.6575
3	2	300	0.6773	0.6815	0.6842	0.679	0.682	0.6138	0.6167	0.6773	0.6795
4	4	30	0.6767	0.6983	0.6933	0.6583	0.6867	0.6767	0.6917	0.6767	0.6867
5	4	60	0.7358	0.7058	0.7467	0.6967	0.7425	0.6767	0.7467	0.73	0.7425
6	4	300	0.7785	0.7848	0.7887	0.7848	0.7813	0.7663	0.7733	0.7785	0.7823
7	10	30	0.71	0.66	0.835	0.63	0.835	0.71	0.835	0.6983	0.805
8	10	60	0.745	0.7333	0.8433	0.685	0.8433	0.745	0.8275	0.745	0.8433
9	10	300	0.8735	0.8632	0.9193	0.8537	0.9032	0.8832	0.9137	0.8735	0.9032
10	20	30	0.665	0.5983	0.9233	0.5983	0.9233	0.665	0.9233	0.655	0.885
11	20	60	0.7483	0.655	0.935	0.6442	0.935	0.7483	0.935	0.7483	0.935
12	20	300	0.8878	0.8607	0.9832	0.805	0.9417	0.8438	0.9648	0.8878	0.9417
13	60	30	0.5	0.5	0.9717	0.5	0.9717	0.5	0.9717	0.5	0.9717
14	60	60	0.5	0.5	0.9683	0.5	0.9683	0.5	0.9683	0.5	0.9683
15	60	300	0.8147	0.8147	0.9683	0.8147	0.9683	0.8147	0.9683	0.8147	0.9683

Table 2(c) Results of experiment 3 (I-0.1I case) using EM clustering

			Model Selection								
Situation	Dimensionality	Sample Size	1 mode	AIC	AIC_Mix	BIC	BIC_Mix	NEC	NEC_Mix	ICLBIC	ICLBIC_Mix
1	2	30	0.7333	0.745	0.705	0.8067	0.8167	0.815	0.8217	0.8433	0.8267
2	2	60	0.7742	0.7858	0.825	0.8483	0.8542	0.8583	0.8583	0.8533	0.855
3	2	300	0.7758	0.833	0.8743	0.8383	0.8778	0.8782	0.8782	0.8782	0.869
4	4	30	0.9167	0.83	0.8333	0.9167	0.9183	0.915	0.8983	0.92	0.925
5	4	60	0.9158	0.825	0.9158	0.9383	0.9492	0.93	0.9467	0.9458	0.935
6	4	300	0.9225	0.9667	0.9622	0.922	0.9683	0.9633	0.9685	0.963	0.9683
7	10	30	0.9683	0.8717	0.8117	0.9683	0.9683	0.9283	0.9283	0.9683	0.9683
8	10	60	0.99	0.9567	0.9325	0.99	0.99	0.9658	0.9775	0.99	0.99
9	10	300	0.9945	0.9787	0.9947	0.9995	0.9988	0.9995	0.9992	0.9985	0.9985
10	20	30	0.945	0.945	0.945	0.945	0.945	0.945	0.945	0.945	0.945
11	20	60	0.9967	0.9158	0.9092	0.9967	0.9975	0.9617	0.9925	0.9967	0.9967
12	20	300	1	0.9997	0.9997	1	1	0.9998	1	1	1
13	60	30	0.5	0.5	0.9983	0.5	0.9983	0.5	0.9983	0.5	0.9983
14	60	60	0.5	0.5	0.9992	0.5	0.9992	0.5	0.95	0.5	0.9992
15	60	300	1	1	1	1	1	1	1	1	1

Table 2(d) Results of experiment 4 (I-I case) using EM clustering

			Model Selection								
Situation	Dimensionality	Sample Size	1 mode	AIC	AIC_Mix	BIC	BIC_Mix	NEC	NEC_Mix	ICLBIC	ICLBIC_Mix
1	2	30	0.6333	0.5617	0.5633	0.6117	0.615	0.62	0.6267	0.6267	0.6267
2	2	60	0.6575	0.5983	0.5875	0.6575	0.6575	0.6333	0.6383	0.6575	0.6575
3	2	300	0.6773	0.6693	0.6735	0.6802	0.6802	0.6773	0.6773	0.6773	0.6773
4	4	30	0.6767	0.6033	0.7615	0.6617	0.7782	0.6433	0.782	0.6767	0.7785
5	4	60	0.7358	0.655	0.6483	0.7358	0.7358	0.6817	0.7158	0.7358	0.7358
6	4	300	0.7785	0.766	0.7615	0.7782	0.7782	0.7825	0.782	0.7785	0.7785
7	10	30	0.71	0.6583	0.625	0.71	0.71	0.66	0.6417	0.71	0.71
8	10	60	0.745	0.7408	0.7008	0.745	0.745	0.69	0.6842	0.745	0.745
9	10	300	0.8735	0.8953	0.891	0.8735	0.8735	0.8892	0.882	0.8735	0.8735
10	20	30	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665	0.665
11	20	60	0.7483	0.7242	0.7317	0.7483	0.7483	0.6725	0.6767	0.7483	0.7483
12	20	300	0.8878	0.9023	0.9023	0.8878	0.8878	0.8282	0.8282	0.8878	0.8878
13	60	30	0.5	0.5	0.9717	0.5	0.9717	0.5	0.7783	0.5	0.9717
14	60	60	0.5	0.5	0.9683	0.5	0.9683	0.5	0.6775	0.5	0.9683
15	60	300	0.8147	0.8033	0.9387	0.8033	0.8147	0.9733	0.7935	0.8033	0.8147

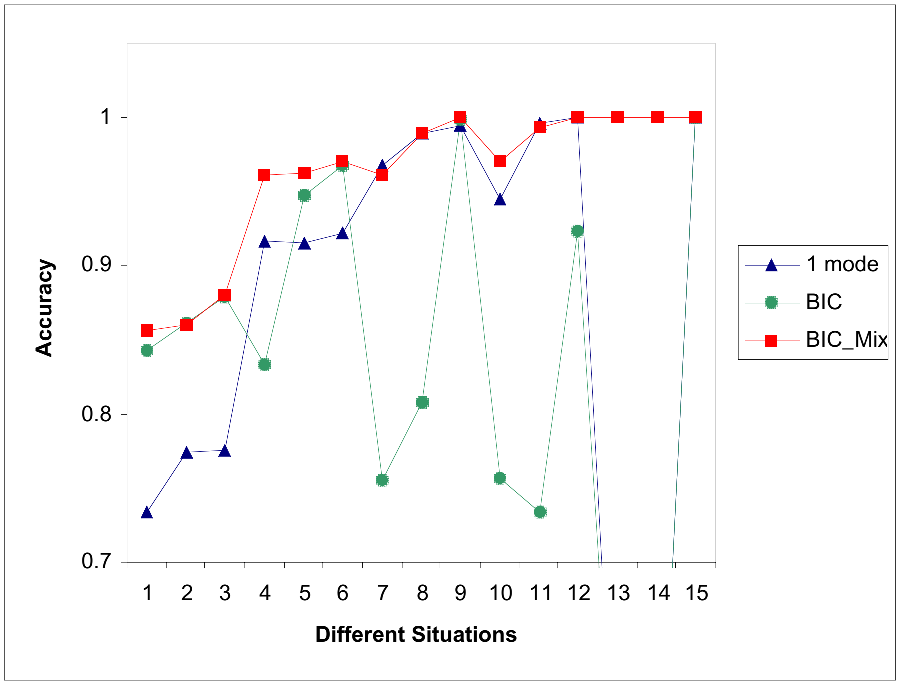

Figure 3(a) Some results of experiment 1 (I-0.1I case) using NM clustering

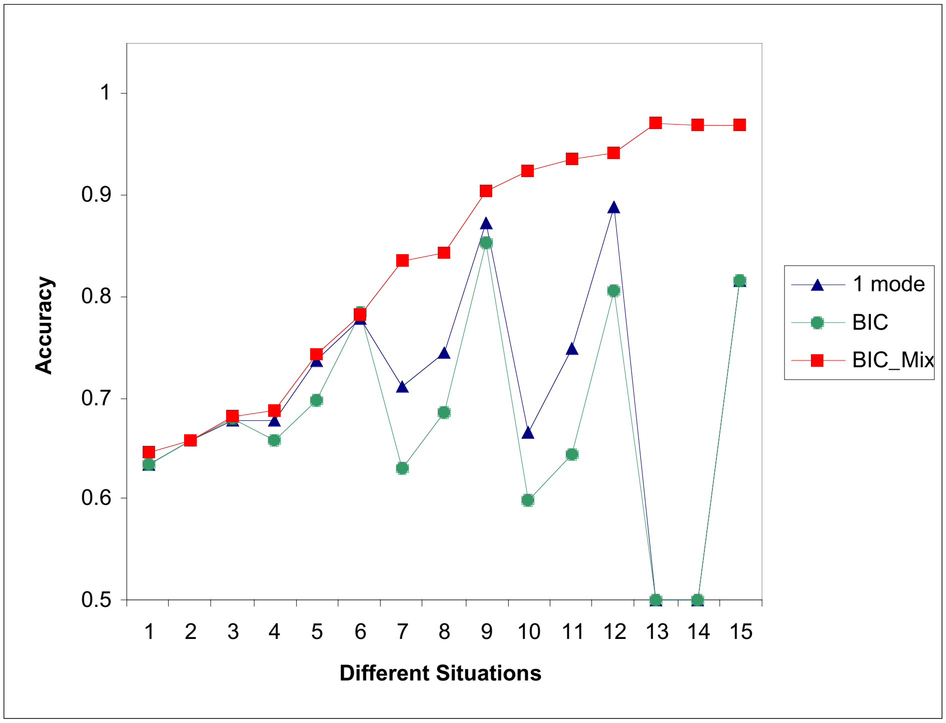

Figure 3(b) Some results of experiment 2 (I-I case) using NM clustering

5.2 Real Data Experiment Results

The results of the simulation study suggests that NM clustering is a better choice to build a mixture classifier, so NM clustering is used on real data experiments. The results are in Table 3. It shows that BIC_Mix has better performance than the others in almost all cases.

Table 3 Results of experiment using NM clustering

Accuracy										
Model Selection		1 mode	AIC	AIC_Mix	BIC	BIC_Mix	NEC	NEC_Mix	ICLBIC	ICLBIC_Mix
Dimensionality	Sample Size									
20	100	0.949	0.9024	0.951	0.925	0.9475	0.7065	0.9199	0.949	0.9312
7	100	0.7394	0.8315	0.8365	0.8314	0.8384	0.7667	0.7503	0.7366	0.7408
20	20	0.4154	0.4154	0.7789	0.4154	0.7789	0.4154	0.6822	0.4154	0.6822
7	20	0.7011	0.6484	0.7163	0.6959	0.7163	0.7011	0.7056	0.7011	0.7056

The above results show that sometimes an original mixture classifier outperforms a simple quadratic classifier but sometimes not. The proposed mixture classifier using BIC_Mix has the advantages of both classifiers and outperforms those two in some situations. Before classifying hyperspectral image data, feature extraction is usually a preprocessing step. The effect of combining feature extraction and mixture classification will be discussed in the next section.

6 Using Mixture Classifier Based on Mix-LOOC2 after Feature Extraction

From the above results, it appears that a mixture classifier based on Mix-LOOC2 is a good choice for classifying data in the original space. But using that mixture classifier in hyperdimensional data is not efficient and will suffer from the Hughes phenomenon more seriously. Before classifying hyperdimensional data, feature extraction is usually used to transform the data from the original hyperdimensional space into a lower dimensional feature space. This section is to explore the performances of combining feature extraction and the mixture classifier based on Mixed-LOOC2 procedures.

6.1 Experiment Design

In this section, the performances of the following four classification procedures are compared.

1. Using DAFE features applied to the simple Gaussian quadratic classifier (DAFE+GC). This is the previous, conventionally used approach and serves as a baseline for comparison.
2. Using DAFE features applied to the mixture classifier based on BIC and Mixed-LOOC2 covariance estimator (DAFE+MC-Mix2).
3. Using NWFE applied to the simple Gaussian quadratic classifier (NWFE+GC).
4. Using NWFE features applied to a mixture classifier based on BIC and Mixed-LOOC2 covariance estimator (NWFE+MC-Mix2).

The experiment data are again in two parts, simulated and real data. Ten simulated data sets with 30 and 60 dimensions and mixture distributions are used in Experiment 6 to compute the average accuracy of four different procedures. The mean vectors and covariance matrices used to generate the data are shown in Table 4. Ten randomly sampled DC Mall and Purdue campus data sets are used in Experiment 7 to compute the average accuracy of the four different procedures. The dimensionality of the DC Mall data sets is 191 and that of the Purdue campus data sets is 126. The class training sample sizes of all real data experiments are 40 pixels.

Table 4 Design of Experiment 6

	class 1		class 2		class 3	
Dim=30, 60, 120	component 1	component 2	component 1	component 2	component 1	component 2
Mean Vector	[2,2,0,…,0]	[0,0,…,0]	[2,4,…,0]	[4,-2,0,…,0]	[-2,0,…,0]	[6,0,…,0]
Covariance	0.1I					
Training Sample Size	20	20	20	20	20	20
Testing Sample Size	200	200	200	200	200	200
	class 4		class 5		class 6	
Dim=30, 120	component 1	component 2	component 1	component 2	component 1	component 2
Mean Vector	[-2,-2,0,…,0]	[0,6,…,0]	[2,-4,…,0]	[-4,2,0,…,0]	[2,0,…,0]	[-6,0,…,0]
Covariance	0.1I					
Training Sample Size	20	20	20	20	20	20
Testing Sample Size	200	200	200	200	200	200

6.2 Experiment Results

The results of experiment 6 are displayed in figures 4(a), (b). The results of experiment 7 are displayed in figures 5(a), (b). They show the following.

1. Figures 4(a) and (b) show that using 2 features from NWFE and the mixture classifier based on Mixed-LOOC2 yields the best performance. It implies that NWFE may preserve the original data distribution situation better than DAFE does.

2. Figure 5(a) shows that the performances of NWFE+GC and NWFE+MC-Mix2 are similar but the performance of DAFE+MC-Mix2 is much better than that of DAFE+GC.

3. Figure 5(b) shows that the performances of DAFE+GC and DAFE+MC-Mix2 are similar but the performance of NWFE+MC-Mix2 is better than that of NWFE+GC.

4. Generally speaking, using the procedure NWFE+MC-Mix2 yielded better results and reduced the Hughes phenomenon. However, it requires more computation time.

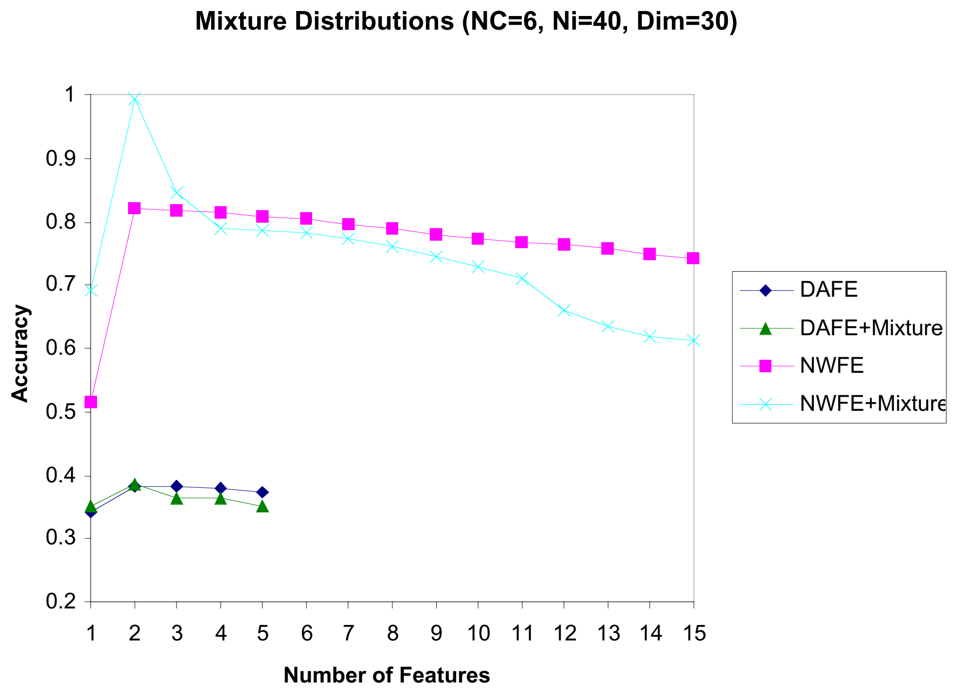

Figure 4(a) Mean of accuracies of simulated data sets (dim=30)

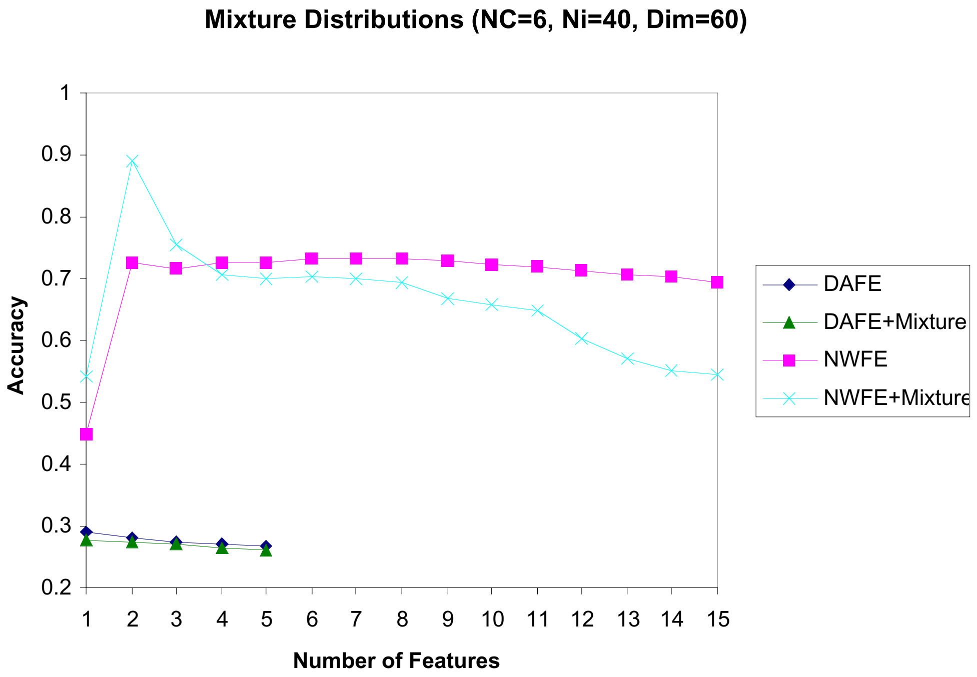

Figure 4(b) Mean of accuracies of simulated data sets (dim=60)

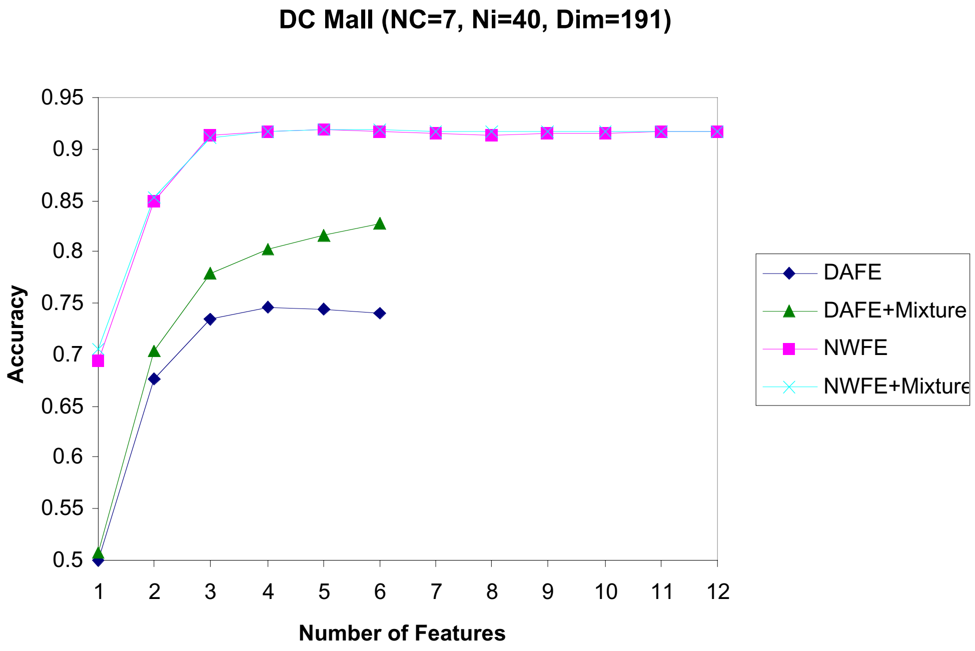

Figure 5(a) Mean of accuracies of DC Mall data sets (dim=191)

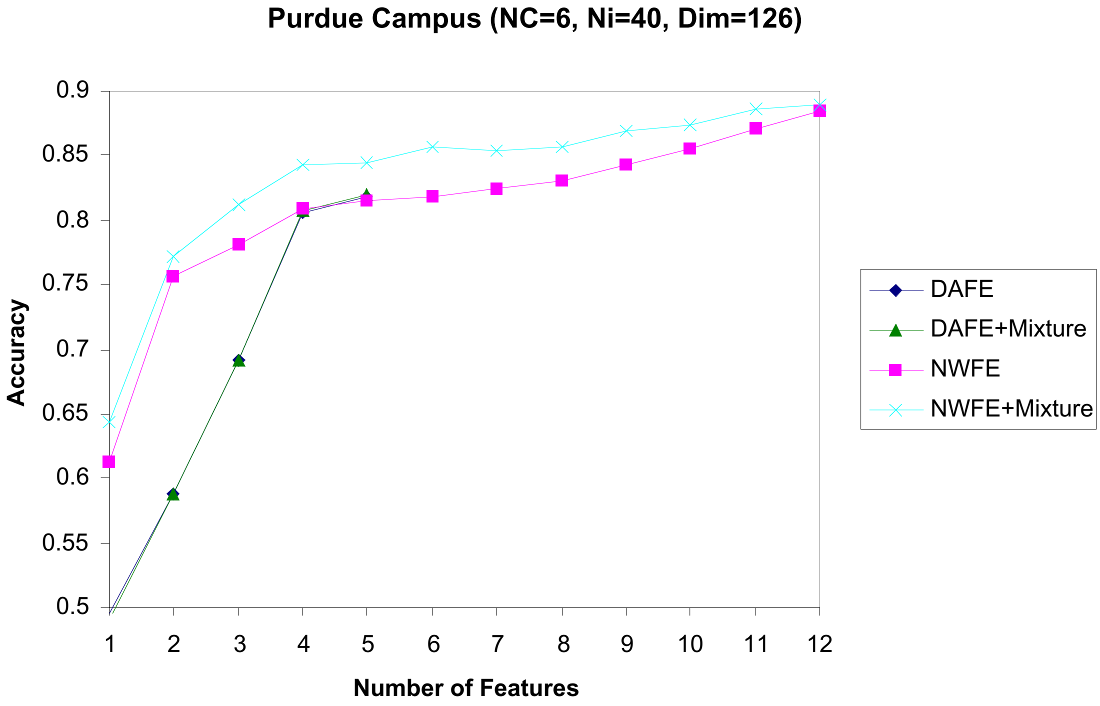

Figure 5(c) Mean of accuracies of Purdue campus data sets (dim=126)

7 Concluding Comments

It has long been known that modeling each class in a data set with a single mode Gaussian density is frequently not a good model. The use of “Gaussian subclasses” to provide a better class model has long been in use, and has shown itself to be an effective way to proceed. This is basically what has been called here a mixture classifier. The problem has been that deciding just how many “subclasses” to use for each class and how to train each has been a substantial challenge to the analyst. Devising an effective scheme for doing this should be a significant aid to the analyst.

In this paper, Mixed-LOOC2 is used with the parameter estimation and model selection steps of mixture classifiers. Experimental results show that the proposed mixture classifier using nearest mean clustering and BIC_Mix has the advantages of both quadratic and original mixture classifier and outperforms those two in some situations.

The performances of combining feature extraction (DAFE and NWFE) and the mixture classifier based on Mixed-LOOC2 procedures are tested. The simulated and real data results show that using NWFE then the mixture classifier based on nearest mean clustering and BIC_Mix index is a robust classification procedure for hyperspectral data.

Estimating mixture distributions and regularized covariance are time consuming. This procedure, which combined these two approaches together, needs more computational time. While computational load is not a major concern, this procedure should be widely applicable.

Reference

[1] David Landgrebe, "Information Extraction Principles and Methods for Multispectral and Hyperspectral Image Data," Chapter 1 of *Information Processing for Remote Sensing*, edited by C. H. Chen, published by the World Scientific Publishing Co., Inc., 1060 Main Street, River Edge, NJ 07661, USA, 1999*.

[2] S. Raudys and A. Saudargiene, “Structures of the Covariance Matrices in Classifier Design”, *Advances in Pattern Recognition*, A. Amin, D. Dori, P. Pudil, and H. Freeman, ed., Berlin Heidelberg: Springer-Verlag pp.583-592, 1998.

[3] J.H. Friedman, “ Regularized Discriminant Analysis,” *Journal of the American Statistical Association*, vol. 84, pp. 165-175, March 1989

[4] W. Rayens and T. Greene, “ Covariance pooling and stabilization for classification.” *Computational Statistics and Data Analysis*, vol. 11, pp. 17-42, 1991

[5] J. P. Hoffbeck and D.A. Landgrebe, Classification of High Dimensional Multispectral Data, Purdue University, West Lafayette, IN., TR-EE 95-14, May, 1995, pp.43-71*.

[6] J. P. Hoffbeck and D.A. Landgrebe, “ Covariance matrix estimation and classification with limited training data” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol 18, No. 7, pp. 763-767, July 1996*.

[7] S. Tadjudin and D.A. Landgrebe, Classification of High Dimensional Data with Limited Training Samples, Purdue University, West Lafayette, IN., TR-EE 98-8, April, 1998, pp35-82*.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, San Diego: Academic Press Inc., 1990.

* Available for download at http://dynamo.ecn.purdue.edu/~landgreb/publications.html.

[9] B-C. Kuo, *Improved Statistics Estimation and Feature Extraction for Hyperspectral Data Classification.* Ph.D. thesis. Purdue University, December 2001*.

[10] H. Akaike, A New Look at the Statistical Identification Model, *IEEE Trans. On Automatic Control*, vol. 19, pp. 716-732, 1974.

[11] G. Schwarz, Estimating the Dimension of a Model, *Annals of Statistics*, vol. 6, pp. 461-464, 1978.

[12] G. Celeux, and G. Soromenho, An Entropy Criterion for Assessing the Number of Clusters in a Mixture Model, *Classification Journal*, vol. 13, pp. 195-212, 1996.

[13] G. McLachlan and D. Peel, Finite Mixture Models, New York: John Wiley & Sons Inc., 2000.

[14] R, P. W. Duin and R. Haeb-Umbach, "Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 762-766, 2001.

[15] K. Fukunaga and M. Mantock, Nonparametric Discriminant Analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, pp. 671-678, 1983.

[16] G. F. Hughes, " On the mean accuracy of statistical pattern recognition", *IEEE Trans. Information Theory*, 1968, Vol. IT-14, No. 1, pp 55-63

* Available for download at http://dynamo.ecn.purdue.edu/~landgreb/publications.html.