# Decision Boundary Feature Extraction for Non-Parametric Classification

**Chulhee Lee and David A. Landgrebe**

School of Electrical Engineering
Purdue University, W. Lafayette, IN 47907-1285
Tel:(317)494-3486, FAX:(317)494-3358
landgreb @ecn.purdue.edu

# DECISION BOUNDARY FEATURE EXTRACTION FOR NON-PARAMETRIC CLASSIFICATION[1]

## Abstract

Feature extraction has long been an important topic in pattern recognition. Although many authors have studied feature extraction for parametric classifiers, relatively few feature extraction algorithms are available for non-parametric classifiers. In this paper we propose a new feature extraction algorithm based on decision boundaries for non-parametric classifiers. We note that feature extraction for pattern recognition is equivalent to retaining "discriminantly informative features" and a discriminantly informative feature is related to the decision boundary. Since non-parametric classifiers do not define decision boundaries in analytic form, the decision boundary and normal vectors must be estimated numerically. We propose a procedure to extract discriminantly informative features based on a decision boundary for non-parametric classification. Experiments show that the proposed algorithm finds effective features for the non-parametric classifier with Parzen density estimation.

## I. INTRODUCTION

Feature extraction has long been an important topic in pattern recognition and has been studied by many authors. Linear feature extraction can be viewed as finding a set of vectors which effectively represent the information content of an observation while reducing the dimensionality. The performance criteria for feature extraction could be different depending on the application. In signal representation, a widely used criteria is mean square error, and the Karhunen-Loeve transformation is one of the techniques under such a criterion; it is optimum in the sense that the mean square error is minimum for a given number of features. In pattern recognition, however, it is desirable to extract features which are focused on discriminating between classes.

[1] This work was funded in part by NASA under grant NAGW-925.

Many algorithms have been proposed for feature extraction for pattern recognition in order to best deal with the varying circumstances which might arise in a pattern recognition problem [1-13]. For example, in some applications, design data (i.e., training set data) may be quite sparse. One approach in this circumstance is to base feature extraction on the statistics of the whole data set [5]. Although this is not optimal in a theoretical sense, it can be used even when underlying class densities are unknown, or precise estimates of them are not possible. In addition, such methods can be used for both parametric and non parametric classifiers. Since, in many cases, it may be difficult to obtain enough training samples, feature extraction methods based on the whole data may be a good and useful solution.

Another approach for feature extraction is to find feature vectors under criteria functions. In discriminant analysis [1], a within-class scatter matrix $\Sigma_W$ and a between-class scatter matrix $\Sigma_b$ are used to formulate a criterion function. A typical criterion is

$$J_1 = \mathrm{tr}(\Sigma_W^{-1}\Sigma_b) \qquad (1)$$

where $\Sigma_W = \sum_i P(\omega_i)\Sigma_i$ (within class scatter matrix)

$\Sigma_b = \sum_i P(\omega_i)(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)$ (between class scatter matrix)

$\mathbf{M}_0 = \sum_i P(\omega_i)\mathbf{M}_i$

Here $\mathbf{M}_i$ , $\Sigma_i$ , and $P(\omega_i)$ are the mean vector, the covariance matrix, and the prior probability of class $\omega_i$, respectively. New feature vectors are selected to maximize the criterion. However, since discriminant analysis mainly utilizes class mean differences, the feature vector selected by discriminant analysis is not reliable if mean vectors are near to one another. And, by using the lumped covariance in the criterion, discriminant analysis may lose some information contained in class covariance differences. Another problem with the criteria functions using scatter matrices is that the criteria generally do not have a direct relationship to the error probability.

Features can be selected under a criterion which is related to the probability of error. The Bhattacharyya distance is a measure of statistical separability and is defined as follows [1]:

$$\mu(\tfrac{1}{2}) = -\ln \int_S [\, p(\mathbf{X}/\omega_1)p(\mathbf{X}/\omega_2)\,]^{\frac{1}{2}} d\mathbf{X} \qquad (2)$$

If normal density functions are assumed, (2) reduces to

$$\mu(\tfrac{1}{2}) = \frac{1}{8}(M_2 - M_1)^t \left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1}(M_2-M_1) + \frac{1}{2}\ln \frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{(|\Sigma_1||\Sigma_2|)^{\frac{1}{2}}}$$

Feature selection algorithms utilizing statistical separability were successfully applied in many applications [1-3, 6-7]. Although theoretically it is possible to calculate (2) for a non-parametric classifier such as Parzen density estimator, in practice, it is frequently not feasible due to a prohibitively long computing time, particularly for high dimensional data.

Although many authors have studied feature extraction for parametric classifiers, relatively few algorithms are available for non-parametric classifiers. The lack of practical feature extraction algorithms for the non-parametric classifier is mainly due to the nature of a non-parametric classifier. Without an assumption about the underlying density functions, feature extraction for non-parametric classifiers is often practically not feasible or very time consuming. Though some general approaches such as principal component analysis might be used for non-parametric classifiers, those methods generally do not find optimum feature sets in the sense of class discrimination.

Some authors studied non-parametric feature extraction [8, 9]. Fukunaga proposed a non-parametric discriminant analysis which is based on non-parametric extensions of commonly used scatter matrices [8]. Patrick proposed a non-parametric feature extraction process where a nonquadratic distance function defined between classes is used to define the best linear subspace [9].

Short and Fukunaga showed that, by problem localization, most pattern recognition problems can be solved using simple parametric forms, while global parametric solution may be untractable [10]. Short and Fukunaga also proposed a feature extraction algorithm using problem localization [11]. They considered feature extraction as a mean-square estimation of the Bayes risk vector. The problem is simplified by partitioning the distribution space into local subregions and performing a linear estimation in each subregion.

Though the computation cost of non-parametric classifiers is often much larger than that of parametric classifiers, there are some cases where the use of non-parametric classifiers is desirable. For instance, if underlying densities are unknown or problems involve complex densities which cannot be approximated by the common parametric density functions, use of non-parametric classifier may be necessary. However, for high dimensional data and multi-source data, the computation cost of non-parametric classifiers can be very large. As a result, there is a greater need for a practical feature extraction algorithm which can take a full advantage of non-parametric classifiers which can define an arbitrary decision boundary.

In this paper, we extend a recently proposed parametric feature extraction algorithm approach [14] to the non-parametric case. The method is based directly on the decision boundary. Instead of utilizing distributions of data, we explore the decision boundary which the employed classifier defines. It has been shown that all feature vectors which are helpful in discriminating between classes can be obtained from the decision boundary [14, 15]. Thus, by extracting features directly from the decision boundary which a non-parametric classifier defines, one can fully explore the advantage of the non-parametric classifier. Since the decision boundary can not be expressed analytically in the non-parametric case, the proposed algorithm finds points on the decision boundary numerically. From these points, feature vectors are extracted. The proposed algorithm predicts the minimum number of features to achieve the same classification accuracy as in the original space while at the same time finding the needed feature vectors.

## II. DECISION BOUNDARY FEATURE EXTRACTION

Consider briefly Bayes' decision rule for minimum error, used in the proposed feature extraction algorithm. Let **X** be an observation in the N-dimensional Euclidean space $E^N$ under hypothesis $H_i$: **X** $\omega_i$ i=1,2. Decisions will be made according to the following rule [1].

$$\text{Decide } \omega_1 \text{ if } h(\mathbf{X}) < t$$
$$\text{else } \omega_2$$

$$\text{where} \quad h(\mathbf{X}) = -\ln\frac{P(\mathbf{X}|\omega_1)}{P(\mathbf{X}|\omega_2)} \tag{3}$$
$$t = \ln\frac{P(\omega_1)}{P(\omega_2)}.$$

In Lee and Landgrebe [14, 15], we defined discriminantly informative features and discriminantly redundant features as follows:

**Definition 1.** A feature vector is discriminantly informative if there exists at least one observation whose classification result would be changed as the observation moves along the direction of the vector. In other words, vector $\beta_k$ is **discriminantly informative** if there exists at least one observation $\mathbf{Y}$ such that

$$(h(\mathbf{Y}) - t)(h(\hat{\mathbf{Y}}) - t) < 0$$
$$\text{where } \mathbf{Y} = \sum_{i=1}^{N} b_i\beta_i \text{ and } \hat{\mathbf{Y}} = \sum_{i=1\ i\neq k}^{N} b_i\beta_i \text{ and } \{\beta_1,\beta_2,..,\beta_N\} \text{ is a basis of } E^N$$

**Definition 2.** A feature vector is discriminantly redundant if, for every observation, the classification result remains unchanged even though the observation moves along the direction of the vector. In other words, vector $\beta_k$ is **discriminantly redundant** if for any observation $\mathbf{X}$

$$(h(\mathbf{X}) - t)(h(\hat{\mathbf{X}}) - t) > 0$$
$$\text{where } \mathbf{X} = \sum_{i=1}^{N} b_i\beta_i \quad \text{and} \quad \hat{\mathbf{X}} = \sum_{i=1\ i\neq k}^{N} b_i\beta_i$$

Fig. 1 shows examples of the discriminant informative feature and the discriminant redundant feature. It was shown that discriminantly informative features and discriminantly redundant features are related to the decision boundary and can be extracted from the decision boundary [14, 15]. It was also shown that discriminantly informative feature vectors have a component which is normal to the decision boundary at at least one point on the decision boundary and discriminantly redundant feature vectors are orthogonal to the vector normal to decision boundary at every point on decision boundary.

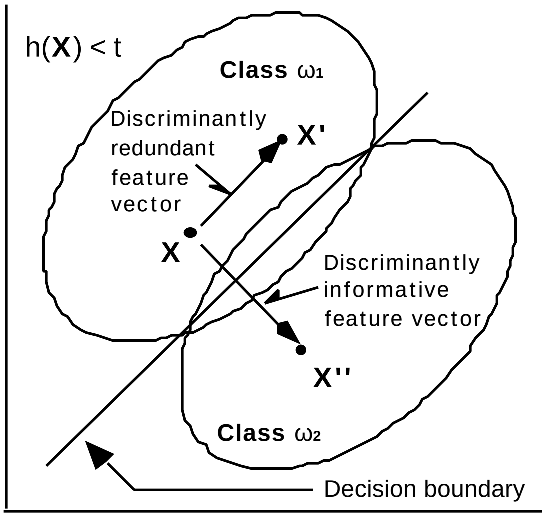
h(**X**) < t
Class ω1
Discriminantly redundant feature vector
X'
X
Discriminantly informative feature vector
X''
Class ω2
Decision boundary

Fig. 1 Examples of the discriminant informative feature and the discriminant redundant feature.

In order to extract discriminantly informative features and discriminantly redundant features from the decision boundary, in [15] the decision boundary feature matrix was defined as follows:

**Definition 3. The decision boundary feature matrix (DBFM):** Let **N(X)** be the unit vector normal to the decision boundary at a point **X** on the decision boundary for a given pattern classification problem. Then the decision boundary feature matrix $\Sigma_{DBFM}$ is defined as

$$\Sigma_{DBFM} = \frac{1}{K} \int_S \mathbf{N}(\mathbf{X})\mathbf{N}^t(\mathbf{X})p(\mathbf{X})d\mathbf{X} \tag{4}$$

where p(**X**) is a probability density function, K= $\int_S p(\mathbf{X})d\mathbf{X}$, and S is the decision boundary, and the integral is performed over the decision boundary.

It was shown that the rank of the decision boundary feature matrix is equal to the smallest dimension where the same classification could be obtained as in the original space, and the eigenvectors of the decision boundary feature matrix of a pattern recognition problem corresponding to non-zero eigenvalues are the necessary feature vectors to achieve the same classification accuracy as in the original space for the pattern recognition problem [14]. In these regards, the following two theorems were derived.

**Theorem 1.** The rank of the decision boundary feature matrix $\Sigma_{DBFM}$ of a pattern classification problem will be the smallest dimension where the same classification could be obtained as in the original space.

**Theorem 2.** The eigenvectors of the decision boundary feature matrix of a pattern recognition problem corresponding to non-zero eigenvalues are the necessary feature vectors to achieve the same classification accuracy as in the original space for the pattern recognition problem.

The decision boundary feature extraction method has been successfully applied to a parametric classifier [14, 15]. The method was shown to be robust and effective even when there are no class mean differences or no class covariance differences [15]. It was also shown that by considering the effective decision boundary only, the number of features can be significantly reduced while achieving almost the same classification accuracy [15]. However, in non-parametric classifiers, when outliers exist, the decision boundary feature matrix defined as in Definition 3 may be calculated from insignificant portions of the decision boundary. Such outlier problems are discussed in [19].

## III. DECISION BOUNDARY FEATURE EXTRACTION FOR NON-PARAMETRIC CLASSIFICATION

### A. Parzen Density Estimation and Selection of Kernel Size

A non-parametric classifier with Parzen density estimation will be used to test the proposed feature extraction algorithm for non-parametric classification; thus we will briefly discuss Parzen density estimation. Parzen density estimation with kernel ϕ is defined as [2]

$$p(X) = \frac{1}{Nh^n}\sum_{i=1}^{N}\phi\left(\frac{X - X_i}{h}\right)$$

where n is the dimensionality of the data, and h is the window size, and N is the number of training samples. The kernel ϕ must be non-negative and satisfy the following condition:

$$\frac{1}{h^n}\int_{R^n}\phi(X)\,dX = 1$$

Although many authors have studied the problem of determining the value of the Parzen scale parameter *h* , no theoretical value of *h* gives consistently optimum results [16]. As a result, we determined the best *h* experimentally in our experiments. Fig. 2 shows the classification results for various *h*. The peak performance occurs when *h* is between 0.5 and 0.7 in this case.

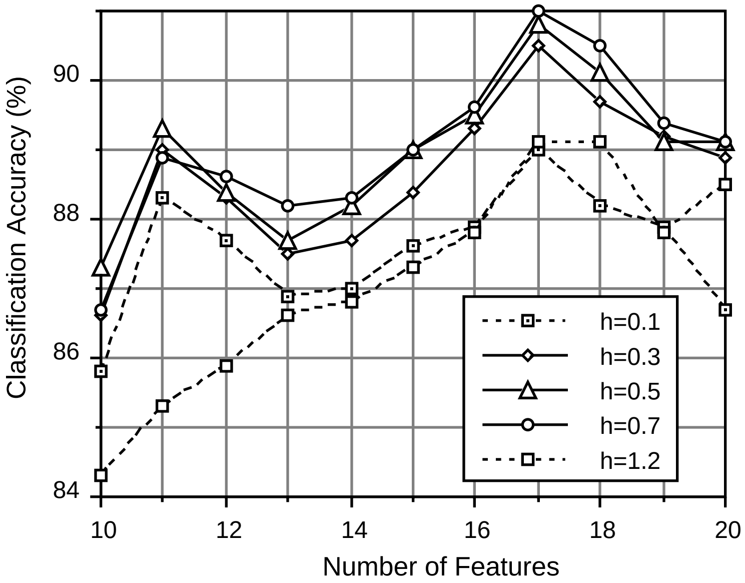

Fig. 2 Determining the best *h* experimentally.

## B. Determining the Decision Boundary and Finding Normal Vectors to the Decision Boundary for Non-Parametric Classifiers

In order to extract feature vectors from the decision boundary of a given classifier, we need to calculate the decision boundary feature matrix $\Sigma_{DBFM}$ as given in Definition 3. Then Theorem 1 and Theorem 2 tell us that the eigenvectors of $\Sigma_{DBFM}$ corresponding to non-zero eigenvalues of $\Sigma_{DBFM}$ are all the feature vectors needed for discriminating between the classes for the given classifier [14, 15]. In order to calculate the decision boundary feature matrix $\Sigma_{DBFM}$, the decision boundary must be found. However, in general, a non-parametric classifier defines an arbitrary decision boundary which may not be expressed in analytic form. Therefore the decision boundary for non-parametric classifiers must be calculated numerically.

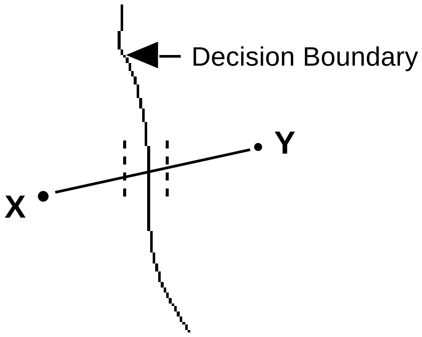

Fig. 3 Finding decision boundary numerically for non-parametric classifiers.

Consider an example in Fig. 3. Assuming **X** and **Y** are classified differently, the line connecting **X** and **Y** must pass through decision boundary. Although, by moving along the line, we can find a point **Z** on the decision boundary, there is no guarantee that the point **Z** is exactly on the true decision boundary, even though h(**Z**)=t. Fig. 4 shows an example. In the example, data are generated for the following statistics.

$$\mathbf{M}_1 = \begin{bmatrix} -0.6 \\ 0.6 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \mathbf{M}_2 = \begin{bmatrix} 0.6 \\ -0.6 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

The points of the decision boundary found numerically are shown along with the true decision boundary plotted as a straight line in Fig. 4. As can be seen, the points of the numerically found decision boundary are distributed along the true decision boundary. However, the points are not exactly on the true decision boundary. The problem that the numerically found decision boundary does not match exactly the true decision boundary becomes more apparent when training samples are limited or the Parzen scale parameter *h* is small. However, in our experiments, we found that inaccurate estimation of the decision boundary has relatively little impact on the performance of the decision boundary feature extraction method for non-parametric classifiers. We will discuss this problem more in the experiments.

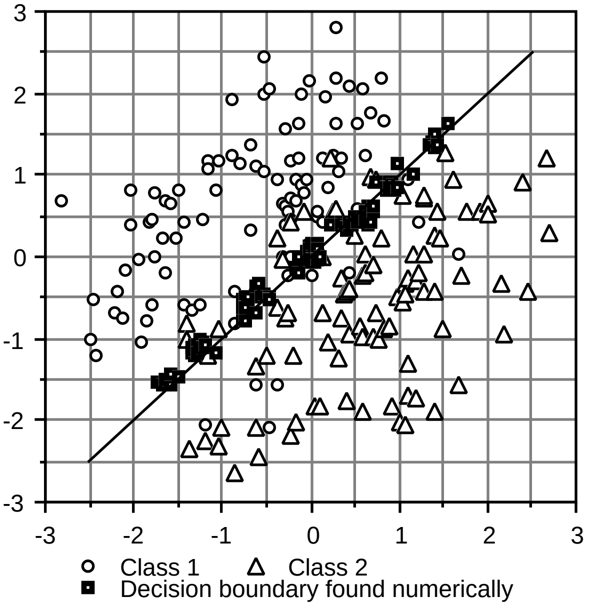

Fig. 4 Finding decision boundary numerically.

A normal vector to the decision boundary at **X** is given by

$$h(\mathbf{X}) = \frac{\partial h}{\partial x_1}\mathbf{x}_1 + \frac{\partial h}{\partial x_2}\mathbf{x}_2 + \cdots\cdots + \frac{\partial h}{\partial x_n}\mathbf{x}_n \qquad (5)$$

However, in non-parametric classifiers, the decision boundary can not be expressed analytically and Eq. (5) can not be used. Instead, we may estimate the normal vector as follows:

$$h(\mathbf{X}) \approx \frac{\Delta h}{\Delta x_1}\mathbf{x}_1 + \frac{\Delta h}{\Delta x_2}\mathbf{x}_2 + \cdots\cdots + \frac{\Delta h}{\Delta x_n}\mathbf{x}_n \tag{6}$$

A problem of estimating a normal vector numerically is that the nearest samples have often much influence on the estimation of normal vectors. This problem becomes more apparent when training samples are limited or the Parzen scale parameter *h* is small. As a result, care must be taken in selecting the Parzen scale parameter *h*, particularly in a high dimensional space. We will discuss this problem more in the experiments.

### C. Decision Boundary Feature Extraction Procedure for Non-Parametric Classification

Now we propose the following procedure to find decision boundary numerically and calculate the decision boundary feature matrix for non-parametric classifiers.

**Procedure for Feature Extraction for Non-Parametric Classifier**
**Utilizing the Decision Boundary**
( 2 pattern class case)

STEP 1: Classify the training data.

STEP 2: For each sample correctly classified as class $\omega_1$, find the nearest sample correctly classified as class $\omega_2$. Repeat the same procedure for the samples correctly classified as class $\omega_2$.

STEP 3: Connect the pairs of samples found in STEP 2. Since a pair of samples are classified differently, the line connecting the pair of samples must pass through the decision boundary. By moving along the line, find the point on the decision boundary or near the decision boundary within a threshold.

STEP 4: At each point found in STEP 3, estimate the unit normal vector $\mathbf{N}_i$ by

$\mathbf{N}_i = \ h(\mathbf{X}) / | \ h(\mathbf{X})|$

where $h(\mathbf{X}) \approx \frac{\Delta h}{\Delta x_1}\mathbf{x}_1 + \frac{\Delta h}{\Delta x_2}\mathbf{x}_2 + \cdots\cdots + \frac{\Delta h}{\Delta x_n}\mathbf{x}_n$

$h(\mathbf{X}) = -\ln\frac{P(\mathbf{X}|\omega_1)}{P(\mathbf{X}|\omega_2)}$ assuming Bayes' decision rule for minimum error is used.

STEP 5: Estimate the decision boundary feature matrix using the normal vectors found in STEP 4.

$$\Sigma_{EDBFM} = \sum_{i} \mathbf{N}_i \mathbf{N}_i^t$$

STEP 6: Select the eigenvectors of the decision boundary feature matrix as new feature vectors according to the magnitude of corresponding eigenvalues.

Euclidean distance is used to find the nearest sample in STEP 2 in our experiments. Fig. 5 shows an illustration of the proposed procedure. Although the proposed procedure does not find the decision boundary where data are sparsely distributed, this is an advantage not a disadvantage of the procedure. By concentrating on the decision boundary where most of data are distributed, the feature extraction can be more efficient [15]. The classification error increase resulting from not considering the decision boundary in the region where data are sparsely distributed will be minimal since there will be very little data in that region.

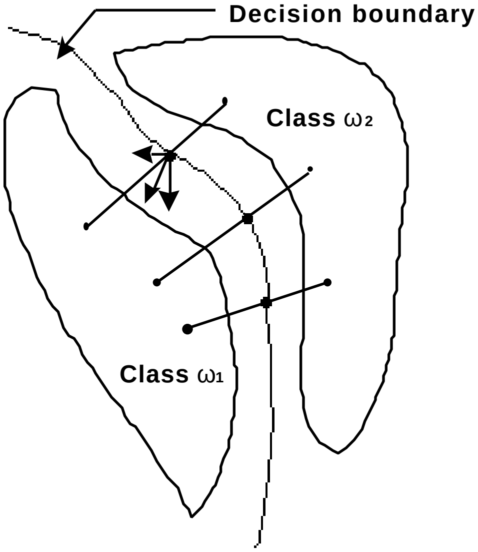

Fig. 5 Illustration of the procedure feature extraction for a non-parametric classifier utilizing decision boundary

Some non-parametric classifiers such as the kNN classifier do not define class probability densities. If the employed non-parametric classifier does not define class probability densities, h(**X**) in (3) can not be calculated. In such a case, normal vectors can not be estimated. In that case, one might find a vector along which the classification result changes most rapidly. For example, let **X** be a point on the decision boundary. Then find the smallest $\Delta x_i$ such that the classification result of $\mathbf{X}+\Delta x_i \mathbf{x}_i$ is different from that of **X**. And we may estimate a unit vector **N** along which the classification result changes most rapidly as follows:

$$\mathbf{N} = \mathbf{V} / |\mathbf{V}| \text{ where } \mathbf{V} \approx \frac{1}{\Delta x_1}\mathbf{x}_1 + \frac{1}{\Delta x_2}\mathbf{x}_2 + \cdots + \frac{1}{\Delta x_n}\mathbf{x}_n$$

If there are more than two classes, the procedure can be repeated for each pair of classes and the total effective decision boundary feature matrix can be calculated by averaging the effective decision boundary feature matrices which are calculated for each pair of classes. If prior probabilities are available, the summation can be weighted. In other words, if there are M classes, the decision boundary feature matrix can be calculated as

$$\Sigma_{DBFM} = \sum_{i}^{M} \sum_{j,\, j\neq i}^{M} P(\omega_i)P(\omega_j)\Sigma_{DBFM}^{ij}$$

where $\Sigma_{DBFM}^{ij}$ is the decision boundary feature matrix between class $\omega_i$ and class $\omega_j$ and $P(\omega_i)$ is the prior probability of class $\omega_i$ if available. Otherwise let $P(\omega_i)$=1/M.

## IV. DECISION BOUNDARY FEATURE EXTRACTION AND PROBLEM LOCALIZATION

By problem localization, Short and Fukunaga showed that most pattern recognition problems can be solved using simple parametric forms [10]. In [11] Short and Fukunaga proposed a feature extraction method using problem localization. In their method, the original space is subdivided into a number of subregions and a linear estimation is performed in each subregion. In their method, a modified clustering algorithm is used to find the subregions. To a certain extent, the decision boundary feature extraction method parallels the problem localization approach. In the problem localization, Short and Fukunaga recognized that a parametric discriminant function can be used in each subregion [11]. In the Decision Boundary Feature Extraction method, we recognized that only a small portion of the decision boundary plays a significant role in discriminating between classes. Consider the case of Fig. 6. The effective decision boundary which is plotted in bold, plays a significant role in discriminating between classes. Even if the effective decision boundary is used, the data still can be classified in the almost same manner as the whole decision boundary is used as shown in Fig. 7. On the other hand, a part of the decision boundary, which is plotted as a plain line, play relatively little role in discriminating between classes while some part of the decision boundary, plotted as a dotted line, are rarely used. Therefore, we recognized that by concentrating on the effective decision boundary, the feature extraction can be more efficient. It is noted that the effective decision boundary need not to be linear or be represented by a parametric form.

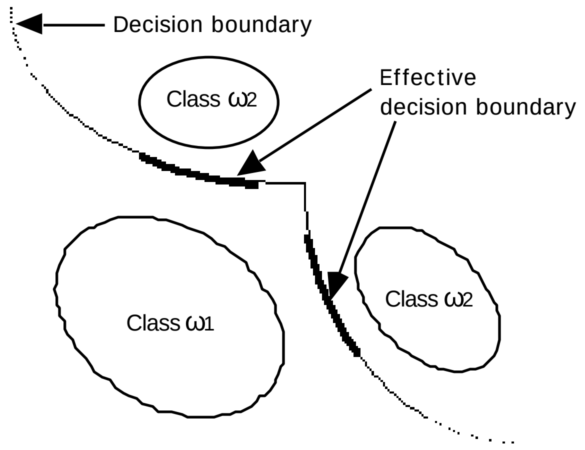

Fig. 6 Decision boundary and effective decision boundary

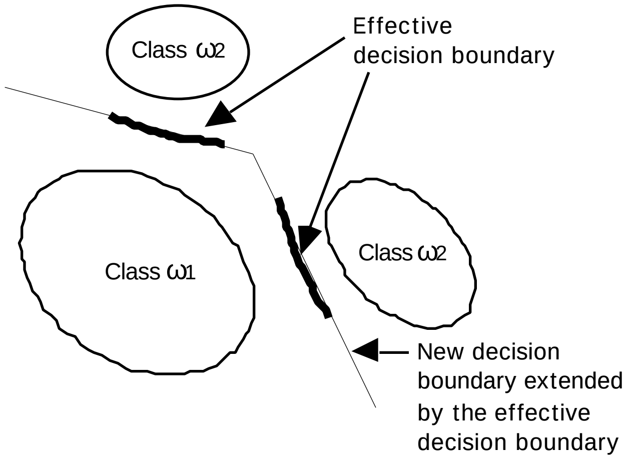
Class ω2
Effective decision boundary
Class ω1
Class ω2
New decision boundary extended by the effective decision boundary

Fig. 7 Effective decision boundary and new decision boundary extended by the effective decision boundary.

However, the Decision Boundary Feature Extraction method differs from the problem localization in several ways. First, the Decision Boundary Feature Extraction method does not divide the pattern space into subregions. Dividing the pattern space into subregions is not an easy task when the number of subregions is unknown. This problem becomes apparent particularly in a multiclass problem with real, high dimensional data. Secondly, the Decision Boundary Feature Extraction method finds a global feature set while a local feature set is found in the problem localization. Thirdly, in the problem localization, Short and Fukunaga take advantage of the fact that class boundaries are likely to be more nearly linear in each subregions while the Decision Boundary Feature Extraction method does not assume that the effective decision boundary is nearly linear or can be represented in a parametric form. In the Decision Boundary Feature Extraction method, the effective decision boundary can be of any shape. Finally the Decision Boundary Feature Extraction method has the capability to predict the minimum number of features needed to achieve the same classification accuracy as in the original space.

## V. EXPERIMENTS AND RESULTS

### A. Experiments with generated data

In order to evaluate closely how the proposed algorithm performs under various circumstances, tests are conducted on generated data with given statistics. The non-parametric classifier was implemented by Parzen density estimation using a Gaussian kernel function. In each example, classification accuracies of the Decision Boundary Feature Extraction method and the discriminant analysis using Eq. (1) as a criterion function are compared. We will refer the Decision Boundary Feature Extraction method as DECISION BOUNDARY FEATURE EXTRACTION, and the discriminant analysis using Eq. (1) as DISCRIMINANT ANALYSIS.

**Example 1.** In this example, class $\omega_1$ is normal with the following statistics:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 3 & 0.5 \\ 0.5 & 3 \end{bmatrix}$$

And class $\omega_2$ is equally divided between two normal distributions with the following statistics:

$$\mathbf{M}_2^1 = \begin{bmatrix} -3 \\ 3 \end{bmatrix} \quad \Sigma_2^1 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix} \text{ and } \mathbf{M}_2^2 = \begin{bmatrix} 3 \\ -3 \end{bmatrix} \quad \Sigma_2^2 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 2 \end{bmatrix}$$

200 samples are generated for each class. Fig. 8 shows the distribution of the data along with the decision boundary found by the proposed procedure numerically. Eigenvalues $\lambda_i$ and eigenvectors $\varphi_i$ of $\Sigma_{EDBFM}$ are calculated as follows:

$$\lambda_1 = 355.0, \quad \lambda_2 = 6.0 \qquad \varphi_1 = \begin{bmatrix} 0.69 \\ -0.72 \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} 0.72 \\ 0.69 \end{bmatrix}$$

Since one eigenvalue is significantly larger than the other, it can be said that the rank of $\Sigma_{EDBFM}$ is 1. That indicates only one feature is needed to achieve the same classification accuracy as in the original space. Considering the statistics of the two classes, the rank of $\Sigma_{EDBFM}$ gives the correct number of features needed to achieve the same classification accuracy as in the original space. Table I shows the classification accuracies of DECISION BOUNDARY FEATURE EXTRACTION and DISCRIMINANT ANALYSIS. DECISION BOUNDARY FEATURE EXTRACTION finds the right feature achieving about the same classification accuracy with one feature while DISCRIMINANT ANALYSIS performs significantly less well in this example since class means are the same.

Table I. Classification accuracies of DECISION BOUNDARY FEATURE EXTRACTION and DISCRIMINANT ANALYSIS in Example 1.

| Number of Features | Discriminant Analysis | Decision Boundary Feature Extraction |
|---|---|---|
| 1 | 54.5 (%) | 92.8 (%) |
| 2 | 91.8 (%) | 92.0 (%) |

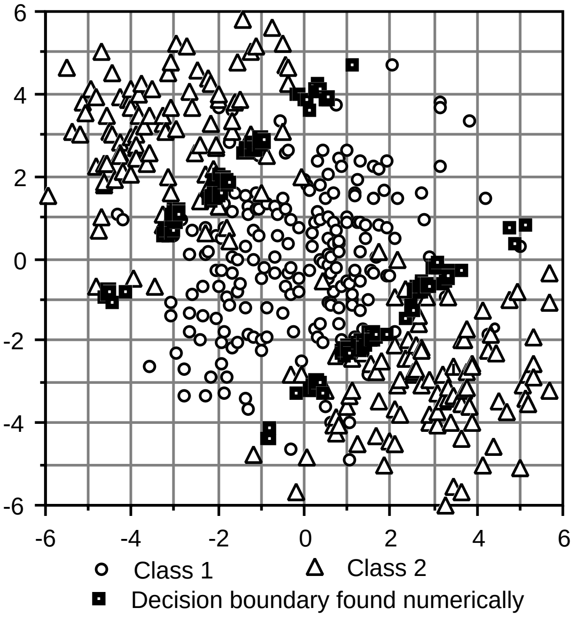

Fig. 8 Data distribution of Example 1. The decision boundary found by the proposed procedure is also shown.

**Example 2.** In this example, class $\omega_1$ is normal with the following statistics:

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

And class $\omega_2$ is equally divided between two normal distributions with the following statistics:

$$\mathbf{M}_2^1 = \begin{bmatrix} 3 \\ 0 \\ 0.1 \end{bmatrix} \quad \Sigma_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{bmatrix} \text{ and } \mathbf{M}_2^2 = \begin{bmatrix} -3 \\ 0 \\ 0.1 \end{bmatrix} \quad \Sigma_2^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

200 samples are generated for each class. From the statistics, it can be said that the decision boundary approximately consists of two hollow cylinders. Fig. 9 shows the distribution of the data in the x1-x2 plane. The decision boundary found by the proposed procedure numerically is also shown. Eigenvalues $\lambda_i$ and eigenvectors $\varphi_i$ of $\Sigma_{EDBFM}$ are calculated as follows:

$$\lambda_1 = 184.0, \quad \lambda_2 = 106.0, \quad \lambda_3 = 21.9$$

$$\varphi_1 = \begin{bmatrix} 0.25 \\ -0.97 \\ 0.00 \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} 0.95 \\ 0.24 \\ 0.17 \end{bmatrix}, \quad \varphi_3 = \begin{bmatrix} -0.17 \\ -0.04 \\ 0.98 \end{bmatrix}$$

$$\text{Rank}(\Sigma_{EDBFM}) \approx 2$$

It can be said that the rank of $\Sigma_{EDBFM}$ is approximately 2. It means two features are needed to achieve the same classification accuracy as in the original space, which agrees with the data. Table II shows the classification accuracies of DECISION BOUNDARY FEATURE EXTRACTION and DISCRIMINANT ANALYSIS. DECISION BOUNDARY FEATURE EXTRACTION find the correct features achieving about the same classification accuracy with two features while DISCRIMINANT ANALYSIS performs significantly less well, since there is no class mean difference.

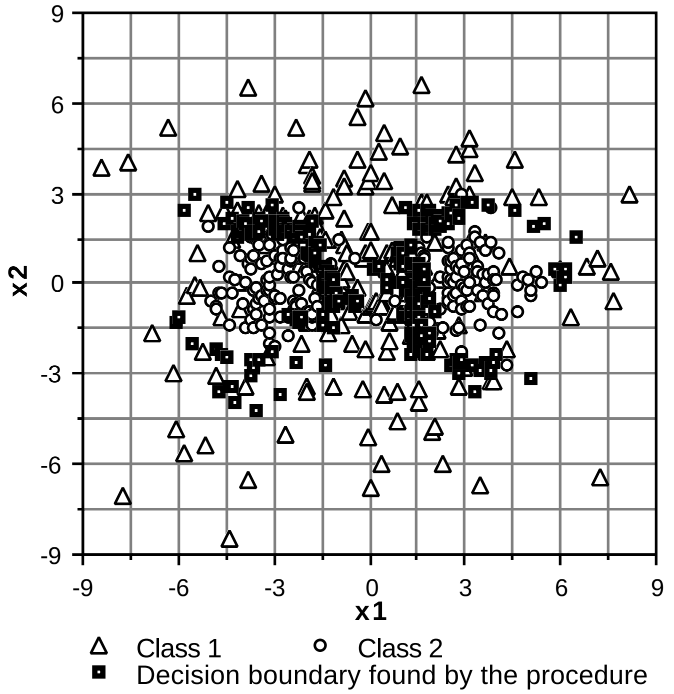

Fig. 9 Data distribution of Example 2. The decision boundary found by the proposed procedure is also shown.

Table II. Classification accuracies of DECISION BOUNDARY FEATURE EXTRACTION and DISCRIMINANT ANALYSIS in Example 2.

| Number of Features | Discriminant Analysis | Decision Boundary Feature Extraction |
|---|---|---|
| 1 | 61.5 (%) | 68.8 (%) |
| 2 | 67.8 (%) | 76.3 (%) |
| 3 | 76.0 (%) | 76.5 (%) |

## B. Experiments with real data

Real data sets were selected from a high dimensional multispectral remote sensing data base of agricultural areas. The data were collected by the Field Spectrometer System (FSS), a helicopter-mounted field spectrometer, as a part of the LACIE program [17]. Table III shows major parameters of FSS.

TABLE III. Parameters of Field Spectrometer System (FSS)

| | |
|---|---|
| Number of Bands | 60 |
| Spectral Coverage | 0.4 - 2.4 μm |
| Altitude | 60 m |
| IFOV(ground) | 25 m |

Along with the proposed algorithm, three other feature extraction algorithms, UNIFORM FEATURE DESIGN, the Karhunen-Loeve transformation (Principal Component Analysis) [2], and the discriminant analysis using Eq. (1) as a criterion function [1] are tested to evaluate and compare the performance of the proposed algorithm. UNIFORM FEATURE DESIGN is a simple band combination procedure. For example, if the number of features is to be reduced to 30, every two consecutive bands are combined to form a new feature. Where the number of features desired is not evenly divisible into 60, the nearest integer number of bands is used. For example, for 9 features, the first 6 original bands were combined to create the first feature, then the next 7 bands were combined to create the next feature, and so on. UNIFORM FEATURE DESIGN is used as a baseline means to evaluate efficiencies of the other feature extraction methods. The discriminant analysis using Eq. (1) is referred as DISCRIMINANT ANALYSIS.

In the first test, 4 classes are chosen from the FSS data. Table IV provides descriptive information on the 4 classes. Fig. 10 shows the mean graph of the 4 classes. As can be seen, there are significant mean differences among classes. In this test, 400 randomly selected samples are used for training and the rest are used for test.

TABLE IV. Class Description

| SPECIES | DATE | No. of Samples |
|---|---|---|
| Winter Wheat | May 3, 1977 | 657 |
| Unknown Crops | May 3, 1977 | 678 |
| Winter Wheat | March 8, 1977 | 691 |
| Unknown Crops | March 8, 1977 | 619 |

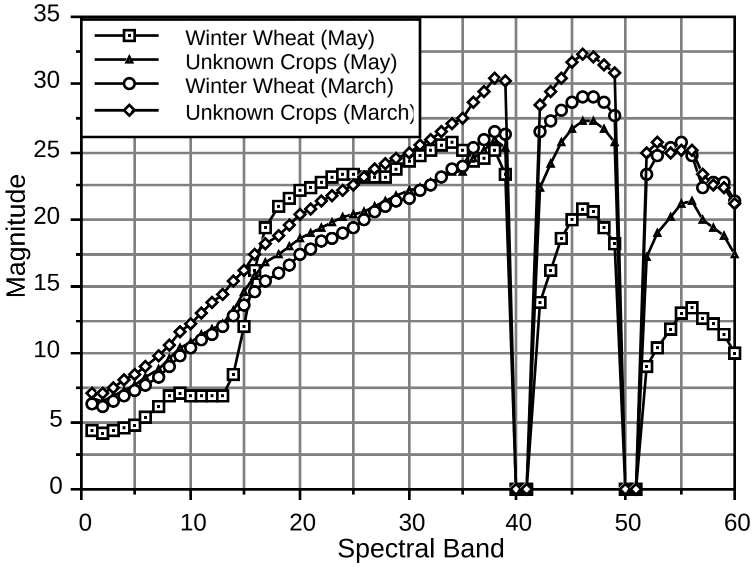

Fig. 10 Mean graph of the four classes in Table IV.

Fig. 11 shows a performance comparison. First the original 60 dimensional data is reduced to 17 dimensional data using UNIFORM FEATURE DESIGN. And then DECISION BOUNDARY FEATURE EXTRACTION, DISCRIMINANT ANALYSIS, and Principal Component Analysis are applied to the 17 dimensional data. With 17 features, the classification accuracy is about 90.0%. At low dimensionality (number of feature ≤3 ), DISCRIMINANT ANALYSIS performs better than the other methods. However, when more than 3 features are used, DECISION BOUNDARY FEATURE EXTRACTION begins to perform better than the other methods.

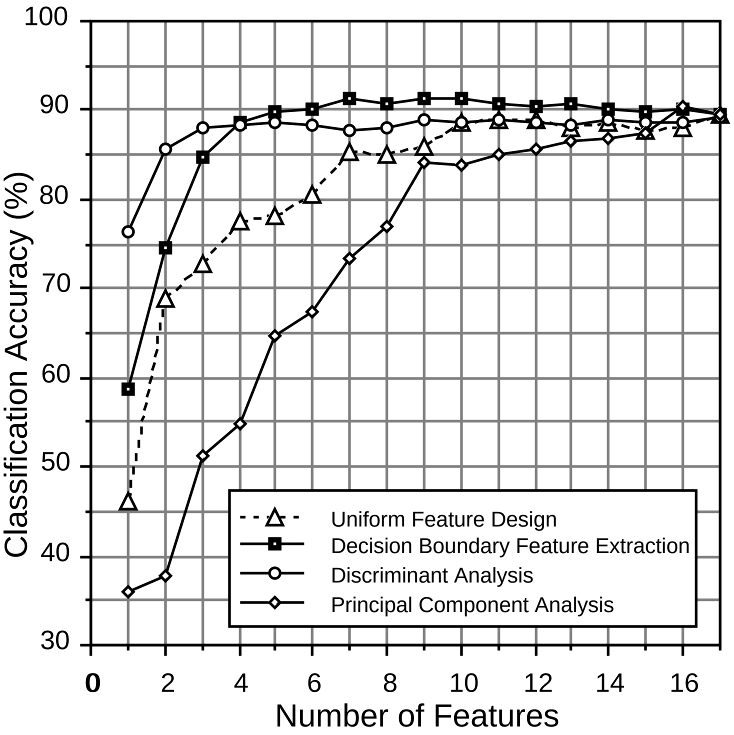

Fig. 11 Performance comparison of UNIFORM FEATURE DESIGN, DECISION BOUNDARY FEATURE EXTRACTION, DISCRIMINANT ANALYSIS, and Principal Component Analysis of the data in Table IV.

In order to test the performance in a multimodal situation, 3 classes with 2 subclasses were chosen. In other words, 2 subclasses were combined to form a new class, thus the data are purposely made multimodal. TABLE V provides information on the classes. Fig. 12 shows a graph of the mean values for the 6 subclasses, and Fig. 13 shows a graph of the mean values of the 3 classes each of which has 2 subclasses. 500 randomly selected samples from each class are used as training data and the rest are used for test.

TABLE V. Class Description

| Class | Subclass | No. of Samples | Total No. of Sample |
|---|---|---|---|
| Class $\omega_1$ | Winter Wheat March 8, 1977 | 691 | 1209 |
| | Spring Wheat July 26, 1978 | 518 | |
| Class $\omega_2$ | Winter Wheat June 26, 1977 | 677 | 1146 |
| | Spring Wheat Sep. 21, 1978 | 469 | |
| Class $\omega_3$ | Winter Wheat Oct. 18, 1977 | 662 | 1103 |
| | Spring Wheat Oct. 26, 1978 | 441 | |

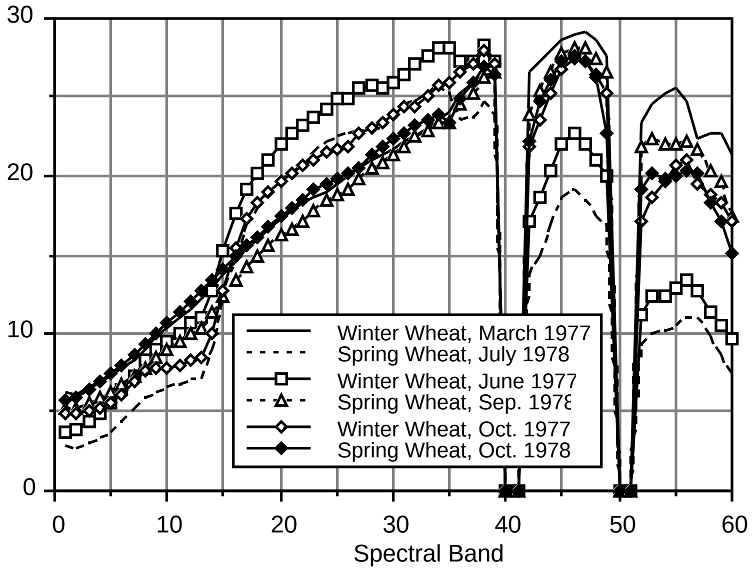

Fig. 12 Mean graph of the 6 subclasses in Table V.

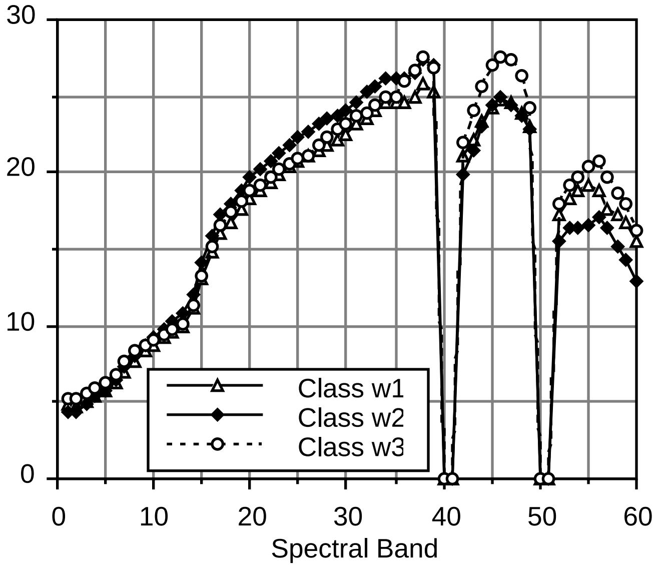

Fig. 13 Mean graph of the 3 classes in Table V.

Fig. 14 shows a performance comparison. With 17 features, the classification accuracy is about 94%. DISCRIMINANT ANALYSIS shows the best performances until 2 features are used. However, the classification accuracies are much lower than the maximum possible classification accuracy, and the comparison seems irrelevant. DECISION BOUNDARY FEATURE EXTRACTION shows consistently better performances when more than 3 features are used. DECISION BOUNDARY FEATURE EXTRACTION achieves about 94% classification accuracy with 7 features while all other methods needs 13-17 features to achieve about the same classification accuracy.

It appears that when class mean differences are reasonably large and classes are unimodal, DISCRIMINANT ANALYSIS finds a good feature set. However, when classes are multi-modal, DISCRIMINANT ANALYSIS does not often find a good feature set. On the other hand, DECISION BOUNDARY FEATURE EXTRACTION finds a good feature set even when classes are multi-modal.

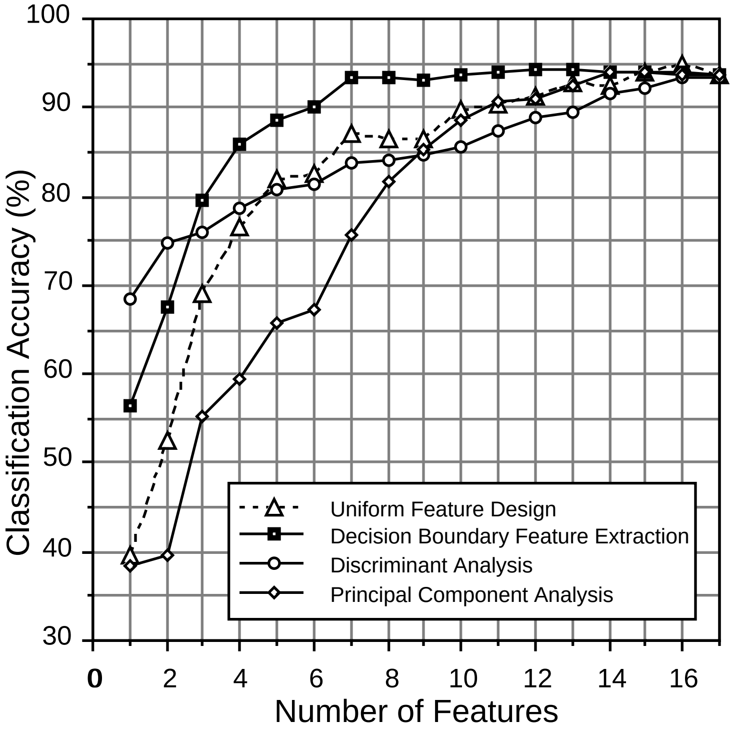

Fig. 14 Performance comparison of UNIFORM FEATURE DESIGN, DECISION BOUNDARY FEATURE EXTRACTION, DISCRIMINANT ANALYSIS, and Principal Component Analysis of the data in Table V.

Table VI lists the eigenvalues of a decision boundary feature matrix of the 17 dimensional data, along with proportions and accumulations. It also shows classification accuracies and normalized classification accuracies obtained by dividing classification accuracies with the classification accuracy obtained using the whole feature set.

The rank of the decision boundary feature matrix ($\Sigma_{DBFM}$) must be decided upon, and in this case, somewhat arbitrarily so. Theoretically, the classification result obtained using all the eigenvectors of the decision boundary feature matrix corresponding to non-zero eigenvalues are the same as the classification result obtained using the whole feature set. However, for real data, eigenvalues of the decision boundary feature matrix are seldom zero, even though some eigenvalues are very close to zero and there are large differences among the eigenvalues. As a result, although it is relatively easy to decide the rank of the decision boundary feature matrix for low dimensional generated data, it becomes less obvious for high dimensional real data. In non-parametric classification, it would be more difficult since decision boundary and normal vectors are estimated. One may add eigenvalues until the accumulation exceeds 95% of the total sum and set that number of the eigenvalues as the rank of the $\Sigma_{DBFM}$. Defined in this way, the rank of the $\Sigma_{DBFM}$ would be 9. Alternatively, one may retain the eigenvalues greater than one tenth of the largest eigenvalue. In this way, the rank of the $\Sigma_{DBFM}$ would be 6. As can be seen in Table VI, the normalized classification accuracy increases monotonically as the accumulation of eigenvalues increases up to 5 features. After 5 features, the classification accuracy is almost saturated and adding more features does not improve classification accuracy. Fig. 15 shows the relationship between the accumulations of eigenvalues and the normalized classification accuracies. More experiments are needed to obtain a better understanding on the relationship between the normalized classification accuracy and the accumulation of eigenvalues.

Table VI. Relationship between eigenvalues of decision boundary feature matrix and classification accuracy. (Ev: Eigenvalues, Pro: Proportion, Accu: Accumulation, Cl. Ac: Classification Accuracy, N. Cl. Ac: Normalized Classification Accuracy.

| | Ev | Pro (%) | Accu (%) | Cl. Ac (%) | N.Cl.Ac (%) |
|---|---|---|---|---|---|
| 1 | 995.2 | 34.5 | 34.5 | 57.1 | 63.2 |
| 2 | 556.4 | 19.3 | 53.8 | 84.7 | 93.7 |
| 3 | 446.4 | 15.5 | 69.3 | 87.9 | 97.2 |
| 4 | 293.3 | 10.2 | 79.4 | 88.5 | 97.9 |
| 5 | 138.5 | 4.8 | 84.2 | 89.8 | 99.3 |
| 6 | 120.5 | 4.2 | 88.4 | 89.8 | 99.3 |
| 7 | 88.6 | 3.1 | 91.5 | 90.1 | 99.7 |
| 8 | 55.8 | 1.9 | 93.4 | 90.1 | 99.7 |
| 9 | 50.8 | 1.8 | 95.2 | 90.5 | 100.1 |
| 10 | 46.2 | 1.6 | 96.8 | 90.2 | 99.8 |
| 11 | 34.0 | 1.2 | 97.9 | 90.1 | 99.7 |
| 12 | 21.4 | 0.7 | 98.7 | 90.2 | 99.8 |
| 13 | 14.1 | 0.5 | 99.2 | 90.3 | 99.9 |
| 14 | 11.3 | 0.4 | 99.6 | 90.4 | 100.0 |
| 15 | 5.8 | 0.2 | 99.8 | 90.4 | 100.0 |
| 16 | 4.5 | 0.2 | 99.9 | 90.4 | 100.0 |
| 17 | 2.3 | 0.1 | 100.0 | 90.4 | 100.0 |

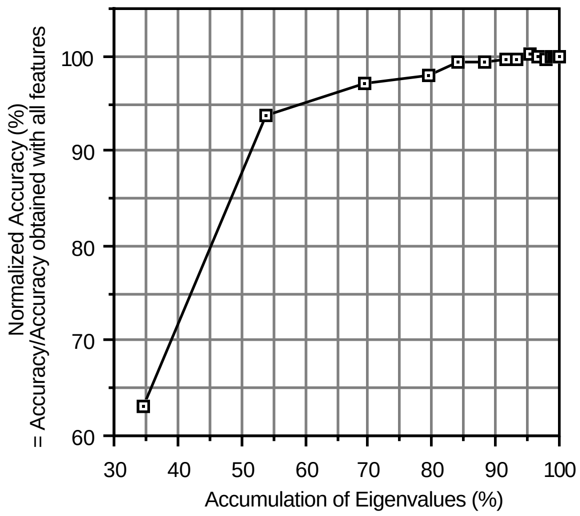

Fig. 15 Relationship between Accumulations of Eigenvalues and Normalized Classification Accuracies.

## VI. ESTIMATION OF DECISION BOUNDARY AND NORMAL VECTOR

Since non-parametric classifiers do not define the decision boundary in analytic form, it must be estimated numerically. And then, from the estimated decision boundary, normal vectors are estimated as follows:

$$h(\mathbf{X}) \approx \frac{\Delta h}{\Delta x_1}\mathbf{x}_1 + \frac{\Delta h}{\Delta x_2}\mathbf{x}_2 + \cdots\cdots + \frac{\Delta h}{\Delta x_n}\mathbf{x}_n$$

Next we will investigate the effect of inaccurate estimation of the decision boundary and normal vectors on the performance of the proposed decision boundary feature extraction.

### A. Effect of Inaccurate Estimation of Decision Boundary

In the proposed procedure, we found a point on the decision boundary by moving along the line connecting two differently classified samples. In other words, by moving along the line, we try to find a point **X** such that

$$h(\mathbf{X})=t$$

When the difference between the decision boundary and an estimated decision boundary is smaller than a threshold, the searching procedure stopped. In other words, if

$$(h(\mathbf{X}) - t)(h(\mathbf{X'}) - t) < 0 \text{ and } |\mathbf{X} - \mathbf{X'}| < \varepsilon$$

we take either **X** or **X'** as a point on the decision boundary. To investigate the sensitivity of the Decision Boundary Feature Extraction method, it was applied to the 17 dimensional data with various thresholds, ε=0.01σ, 0.05σ, 0.1σ, 0.5σ, 1σ and 2σ, where σ is the average standard deviation, i.e.,

$$\sigma = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sigma_j^i$$

where N is the number of features, M is the number of classes, and $\sigma_j^i$ is j-th feature standard deviation of class $\omega_i$.

With 17 features, the classification accuracy is 90.4%. Fig. 16 shows the performance comparison for the first 5 features. For 1 feature, there is not much difference. For 2 features, the classification accuracy decreases as the threshold increases. If more than 2 features are considered, the performances are essentially the same. When 3 features are used, all thresholds achieve about 89% classification accuracy. From the experiments, it appears that the threshold between 0.05σ and 0.5σ would be reasonable and the performance of the Decision Boundary Feature Extraction method does not appear to be very sensitive to inaccurate estimation of the decision boundary.

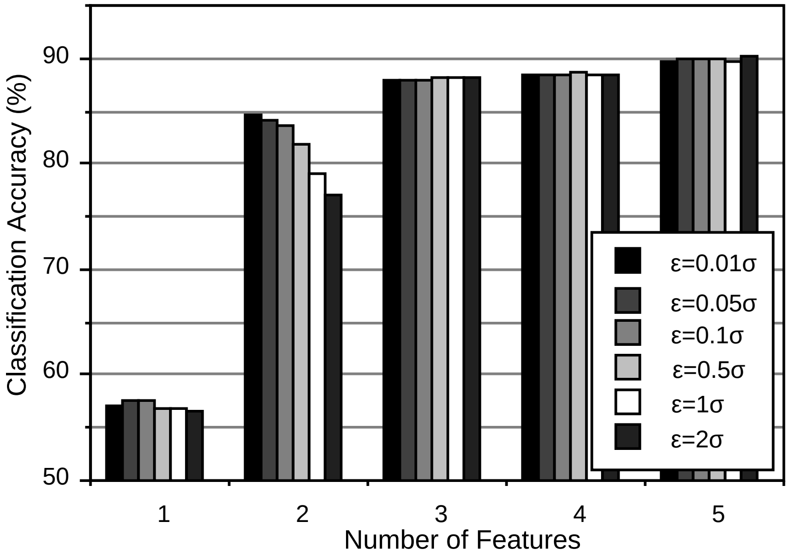

Fig. 16 Effect of inaccurate estimation of decision boundary on the performance of the Decision Boundary Feature Extraction method.

### B. Effect of the Parzen Scale Parameter *h* in Estimating Normal Vectors

Since normal vectors are estimated using (6), the Parzen scale parameter *h* will affect the estimation of normal vectors. Since normal vectors are used to estimate the decision boundary feature matrix, the Parzen scale parameter will affect the performance of the Decision Boundary Feature Extraction method. In the following test, we estimated the normal vectors using various Parzen scale parameters and investigate the effect of the Parzen scale parameter on the performance of the Decision Boundary Feature Extraction method. The Decision Boundary Feature Extraction method is applied to 18 dimensional data. With 18 features the classification accuracy is 92.9%. Fig. 17 shows the performance comparison for various Parzen scale parameters in estimating normal vectors. When h=0.3, 0.5, 0.7, and 1.0, the classification accuracies with 3 features are 92.6%, 92.3%, 92.2%, and 92.1%, respectively. As larger Parzen scale parameters are used (h ≥ 2), classification accuracies decrease, though the decreasing rate is relatively small. However, if the Parzen scale parameter is too small (h=0.1), the classification accuracy decreases considerably. Overall, the Parzen scale parameters between 0.5 and 1.0 give best results in this case. Although the performance of the Decision Boundary Feature Extraction method does not seem to be very sensitive to the variation of the Parzen scale parameter, care must be taken that the Parzen scale parameter should not be too small or too large for a given data.

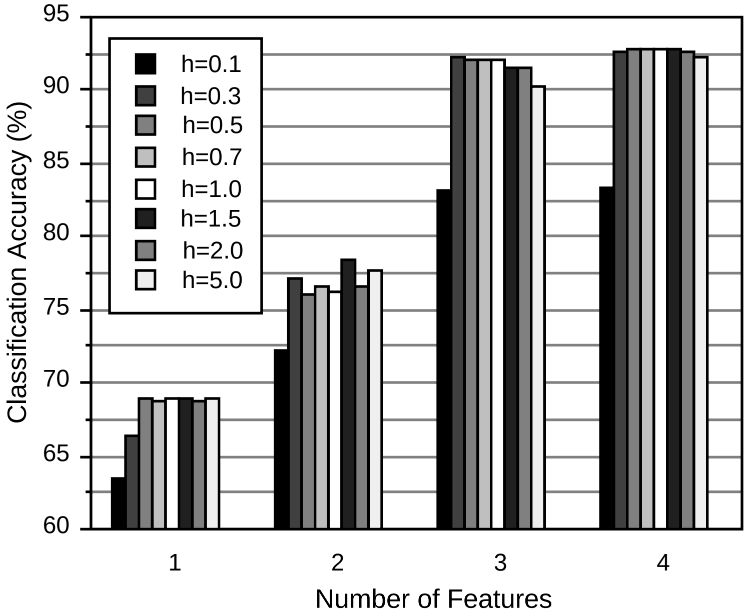

Fig. 17 Performance comparison for various Parzen scale parameters in estimating normal vectors.

## VII. CONCLUSION

Decision boundary feature selection is a new feature selection technique which is derived from the fact that all the feature vectors needed in discriminating between classes for a given classifier can be obtained from the decision boundary defined by the given classifier. Instead of utilizing class mean differences or class covariance differences, the method utilizes the decision boundary directly. As a result, the method does not deteriorate under the circumstances of equal means or equal covariances, and can be used for both parametric and non-parametric classifiers. In this paper we proposed a decision boundary feature selection algorithm for non-parametric classifiers. By directly utilizing the decision boundary defined by an employed non-parametric classifier without any assumption about the distribution of data, the proposed feature selection algorithm can take advantage of the generality of the non-parametric classifier, which can define a complex decision boundary. The experiments show that the performance of the proposed algorithm is very promising. The importance of such algorithms is enhanced as the use of non-parametric classifiers such as neural networks continues to grow [18].

Compared with the conventional feature selection algorithms, the proposed algorithm predicts the minimum number of features to achieve the same classification accuracy as in the original space and at the same time finds the needed feature vectors which have a direct relationship with classification accuracy. Unlike some of the conventional extraction algorithms using the lumped covariance, the proposed algorithm takes a full advantage of the information contained in class covariance differences by extracting new features directly from the decision boundary. Since the information contained in the second order statistics increases its importance in discriminating classes in high dimensional data, the proposed algorithm also has potential for feature extraction for high dimensional data and multi-source data.

## REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* Academic Press, 1972.

[2] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*", John Wiley & Sons, 1973.

[3] P. H. Swain & S.M. Davis, *Remote Sensing: The Quantitative Approach*, McGraw–Hill, 1978.

[4] M. J. Muasher and D. A. Landgrebe, "The K–L expansion as an effective feature ordering technique for limited training sample size," IEEE Trans. Geosci. Remote Sensing, vol. GE-21, pp. 438-441, 1983

[5] Chin-Chien Thomas Chen and D. A. Landgrebe, "A Spectral Feature Design System for the HIRIS/MODIS Era," IEEE Trans. Geosci. Remote Sensing, vol. GE-17, pp 681-686, 1989.

[6] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," IEEE Trans. Communication Technology, vol. COM-15, No. 1, pp. 52-60, 1967.

[7] P. H. Swain and R. C. King, " Two Effective Feature Selection Criteria for Multispectral Remote Sensing," Proc. First Int. Joint Conf. On Pattern Recognition, 536-540, 1973.

[8] K. Fukunaga and J. M. Mantock, "Nonparametric Discriminant Analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 6, pp. 671-678, November 1983.

[9] E. A. Patrick and F. P. Fischer II, "Nonparametric Feature Selection," IEEE Trans. on Information Theory, Vol. IT-15, No. 5, pp.577-584, 1969.

[10] K. Fukunaga and R. D. Short, "Generalized Clustering for Problem Localization," IEEE Transactions on Computers, Vol. C-27, No. 2, pp. 176-181, Feb. 1978.

[11] R. D. Short and K. Fukunaga, "Feature Extraction Using Problem Localization," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 3, pp. 323-326, May 1982.

[12] K. Fukunaga and W. L. G. Koontz, "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering", IEEE Trans. Computer, Vol. C-19, No. 4, pp 311-318, April 1970.

[13] D. H. Foley and J. W. Sammon, "An Optimal Set of Discriminant Vectors," IEEE Trans. Computer, vol. C-24, No. 3, pp.281-289, Mar. 1975.

[14] C. Lee and D. A. Landgrebe, "Feature Selection Based on Decision Boundaries," in Proc. IEEE International Geoscience & Remote Sensing Symposium, pp. 1471-1474, 1991.

[15] Chulhee Lee and David A. Landgrebe, "Feature Extraction Based on Decision Boundaries," to be published in IEEE Transactions on Pattern Analysis and Machine Intelligence.

[16] K. Fukunaga and D. M. Hummels, "Bayes Error Estimation Using Parzen and k-NN Procedures," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 5, pp. 634-643, September 1987.

[17] L. L. Biehl, *et al.*, "A Crops and Soils Data Base For Scene Radiation Research," in Proc. Machine Process. of Remotely Sensed Data Symp., West Lafayette, IN 1982, pp 169-177.

[18] Chulhee Lee and David A. Landgrebe, "Feature Selection for Neural Networks Using Parzen Density Estimator," to be published in Proc. IEEE International Geoscience & Remote Sensing Symposium, 1992.

[19] Chulhee Lee, “Feature Extraction And Classification Algorithms for High Dimensional Data, PhD Thesis, Purdue University School of Electrical Engineering, December 1992.

## Proof of the theorems, extracted from [14, 15]

**Theorem 1.** If a vector is parallel to the tangent hyper-plane to the decision boundary at every point on the decision boundary for a pattern classification problem, the vector contains no information useful in discriminating classes for the pattern classification problem, i.e., the vector is discriminantly redundant.

*Proof*. Let $\{\beta_1,\beta_2,..,\beta_N\}$ be a basis of the N-dimensional Euclidean space $E^N$, and let $\beta_N$ be a vector that is parallel to the tangent hyper-plane to the decision boundary at every point on the decision boundary. Let **W** be a subspace spanned by N-1 spanning vectors, $\beta_1,\beta_2,..,\beta_{N-1}$, i.e.,

$$\mathbf{W} = \mathrm{Span}\{\beta_1,\beta_2,..,\beta_{N-1}\} \text{ and } \dim(\mathbf{W}) = N-1$$

If $\beta_N$ is not a discriminantly redundant feature, there must exist an observation **X** such that

$$(h(\mathbf{X}) - t)(h(\hat{\mathbf{X}}_W) - t) < 0$$

$$\text{where } \mathbf{X} = \sum_{i=1}^{N} b_i\beta_i \text{ and } \hat{\mathbf{X}}_W = \sum_{i=1}^{N-1} c_i\beta_i$$

Without loss of generality, we can assume that the set of vectors $\beta_1,\beta_2,..,\beta_N$ is an orthonormal set. Then $b_i = c_i$ for i=1,N-1. Assume that there is an observation **X** such that

$$(h(\mathbf{X}) - t)(h(\hat{\mathbf{X}}_W) - t) < 0$$

This means **X** and $\hat{\mathbf{X}}_W$ are on different sides of the decision boundary. Then the vector

$$\mathbf{X}_d = \mathbf{X} - \hat{\mathbf{X}}_W = b_N\beta_N$$

where $b_N$ is a coefficient, must pass through the decision boundary. But this contradicts the assumption that $\beta_N$ is parallel to the tangent hyper-plane to the decision boundary at every point on the decision boundary. Therefore if $\beta_N$ is a vector parallel to the tangent hyper-plane to the decision boundary at every point on the decision boundary, then for all observations **X**

$$(h(\mathbf{X}) - t)(h(\hat{\mathbf{X}}_W) - t) > 0$$

Therefore $\beta_N$ is discriminantly redundant. Fig. A1 shows an illustration of the proof.

**Q.E.D.**

From the theorem, we can easily derive the following lemmas which are very useful in finding discriminantly informative features.

**Lemma 1.** If vector **V** is orthogonal to the vector normal to decision boundary at every point on decision boundary, vector **V** contains no information useful in discriminating classes, i.e., vector **V** is discriminantly redundant.

**Lemma 2.** If a vector is normal to the decision boundary at at least one point on the decision boundary, the vector contains information useful in discriminating classes, i.e., the vector is discriminantly informative.

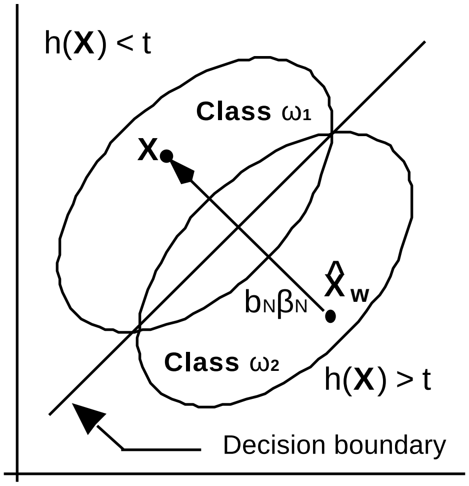

Fig. A1 If two observations are on the different sides of the decision boundary, the line connecting the two observations will pass through the decision boundary.

**Theorem 2.** The rank of the decision boundary feature matrix $\Sigma_{DBFM}$ of a pattern classification problem will be the smallest dimension where the same classification could be obtained as in the original space.

*Proof:* Let **X** be an observation in the N-dimension Euclidean space $E^N$ under the hypothesis $H_i$: $\mathbf{X}$ $\omega_i$ i = 1, 2. Let $\Sigma_{DBFM}$ be the decision boundary feature matrix as defined in Definition 3. Suppose that

$$\mathrm{rank}(\Sigma_{DBFM}) = M \leq N.$$

Let $\{\varphi_1, \varphi_2,.., \varphi_M\}$ be the eigenvectors of $\Sigma_{DBFM}$ corresponding to non-zero eigenvalues. Then a vector normal to the decision boundary at any point on decision boundary can be represented by a linear combination of $\varphi_i$, i=1,M. In other words, for any vector **N** normal to the decision boundary

$$\mathbf{N} = \sum_{i=1}^{M} a_i \varphi_i$$

Since any linearly independent set of vectors from a finite dimensional vector space can be extended to a basis for the vector space, we can expand $\{\varphi_1, \varphi_2,.., \varphi_M\}$ to form a basis for the N-dimension Euclidean space. Let $\{\varphi_1, \varphi_2,.., \varphi_M, \varphi_{M+1},.., \varphi_N\}$ be such a basis. Without loss of generality, we can assume $\{\varphi_1, \varphi_2,.., \varphi_M, \varphi_{M+1},.., \varphi_N\}$ is an orthonormal basis. One can always find an orthonormal basis for a vector space using the *Gram-Schmidt* procedure. Since the basis is assumed to be orthonormal, it can be easily seen that the vectors $\{\varphi_{M+1}, \varphi_{M+2},.., \varphi_N\}$, are orthogonal to any vector **N** normal to the decision boundary. This is because for i = M+1,N

$$\varphi_i^t \mathbf{N} = \varphi_i^t \sum_{k=1}^{M} a_k \varphi_k$$

$$= \sum_{k=1}^{M} a_k \varphi_i^t \varphi_k = 0 \qquad \text{since } \varphi_i^t \varphi_k = 0 \text{ if } i \neq k$$

Therefore since the vectors $\{\varphi_{M+1}, \varphi_{M+2},.., \varphi_N\}$ are orthogonal to any vector normal to the decision boundary, according to Lemma.1, the vectors $\{\varphi_{M+1}, \varphi_{M+2},.., \varphi_N\}$ are discriminantly redundant. Therefore the number of discriminantly redundant features is N – M and only M features are needed to achieve the same classification could be obtained as in the original space.

**Q.E.D.**

From Theorem 2 we can easily derive the following theorem which is useful to find the necessary feature vectors.

**Theorem 3.** The eigenvectors of the decision boundary feature matrix of a pattern recognition problem corresponding to non-zero eigenvalues are the necessary feature vectors to achieve the same classification accuracy as in the original space for the pattern recognition problem.