Journal of Econometrics and Statistics
Vol. 5, Issue 1, 2025, pp. 37-56

A data-centric study on relation between n and p in PCA based parametric and nonparametric classification

Bodhoditya Barma[a] and Saran Ishika Maiti[a]

[a] *Department of Statistics, Visva-Bharati, Santiniketan*

ABSTRACT
In modern statistics, the big data issue is increasingly widespread. Long since principal component analysis (PCA) is a technique for reducing the dimensionality of such data sets. Principal component regression further reduces this large number of explanatory variables to a more handy model. This article explains the relationship of no of variables(p) and no of observations(n) in principal component-based statistical classification techniques both in the parametric and non-parametric ways. It discusses on the amount of misclassification error through the adaptive data analysis technique. In reality, we established that reducing a large number of candidate explanatory variables does not make principal component-based classification more worthy. In fact for non-Gaussian populations, variable-based non-parametric classification comes out more convincing.

KEYWORDS
Statistical classification, linear discriminant analysis, nonparametric classification, kernel, big data analytics

1. Introduction

In recent years, higher dimensional data has become increasingly common across various fields. To tackle this type of data, statistical analysis demands more development and efficiency. In this issue, Principal Component Analysis (PCA) has long been recognized as a powerful technique for reducing the dimensionality of large datasets, thereby simplifying the complexity of data and making it more manageable. However, the relationship between the number of variables (p) and the number of observations (n) in statistical classification techniques is not direct and universal. In parametric as well as non-parametric classification, there lies a different effect between the number of variables and the number of observations in the construction of the classifying rule. While PCA is widely utilized in both parametric and non-parametric classification methods, its effectiveness for higher dimensional datasets is often contingent on the underlying data distribution and the balance between p and n. The common conception that a large number of explanatory variables enhances the accuracy of classification models is sometimes misleading, especially for non-Gaussian populations where PC-based techniques may not perform optimally.

CONTACT Author[c]. Email: saranishika.maiti@visva-bharati.ac.in

Article History
Received : 28 October 2024; Revised : 29 November 2024; Accepted : 01 December 2024; Published : 08 January 2025

To cite this paper

Bodhoditya Barma & Saran Ishika Maiti (2025). A data-centric study on relation between n and p in PCA based parametric and nonparametric classification. *Journal of Econometrics and Statistics.* 5(1), 37-56.

Vapnik ([17]) first introduces kernel-based classification. This method easily separates the non-linearly separable classes after mapping into feature space. However, this method fails to deal with the high dimensionality and multicollinearity of the datasets. Scholkopf *et al* ([14]) improves kernel-based classification by applying principal component analysis to deal with high dimensionality and multicollinearity. Later, taking into account of kernel-based principal component analysis and also of incorporating kernel trick, which is eigenvalue decomposition of kernel matrix, Baudat ([5]) provides an improved kernel-based classification technique which also deals with the same. These methods never disclosed how much the high dimensionality affects the classification techniques or how classification techniques react with high dimensional data for low-class size. In fact for "poorly posed" situation when sample size (n) is relatively smaller with respect to number of feature variable (p) but $n > p$, inverse of sample covariance matrix S^{-1} gets unstable which subsequently turns linear discriminant analysis and kernel Fisher discriminant analysis based on principal components a less performative. Bai[3] showed that when $\frac{p}{n} \to y \in (0, \infty)$ limiting spectral distribution of sample covariance matrix S is the Marchenko-Pastur distribution with ratio index y. Hence, consideration of Gaussian population should not be viable.

In this article through an extensive simulation study, we explore the intricate relationship between the number of variables(p) and observations(n) in variable-based as well as principal component-based classification, under both parametric and non-parametric frameworks. The impact of number of variables with respect to number of observations on the misclassification error rate is elucidated, unfurling insights into when and how variable-based and PC-based classification techniques are more effective corresponding to different parametric and non-parametric methods of classifications. Here we adopt linear discriminant analysis (LDA) as a parametric classification tool and its nonparametric analogous Kernel Fisher Discriminant Analysis(KFDA) as non-parametric tools. Support vector machine is chosen as a potential n We also endeavour to explore the relationship in case of various parametric and non-parametric classification methods, applied on different real-life data set.

This short article is organized as follows. Brief descriptions of different applied methods are discussed in section 2. Section 3 unravels the detailed analysis and exploration of simulated datasets for binary and multivariate classes with the corresponding misclassification errors. In section 4 experiments are performed on two different types of real-life datasets. Finally, section 5 concludes this article with a few directions to future studies.

2. Description of methods applied

In parametric classification techniques, we have to assume that the data follows a specific distribution, such as Gaussian distribution. However, real-world data often doesn't follow these assumptions. Non-parametric methods don't assume a predefined form for the data's distribution, making them suitable when the true distribution is unknown or non-normal. Taking into account of the advantages of non-parametric classification techniques, in our study, we mainly focused on non-parametric classification methods, specifically kernel-based classification techniques.

Definition 2.1. *A kernel is a function used for implicitly mapping data on input space to a higher-dimensional space, enabling the application of linear methods to solve non-*

linear problems.

A pair of examples of kernel mapping functions are given below.

- Polynomial kernel degree 3 is

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}'\mathbf{y} + c)^3.$$

  Here $\mathbf{x}$ and $\mathbf{y}$ are the input vectors. c is a constant that trades off the influence of higher-order versus lower-order terms.
- Gaussian kernel is

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{||\mathbf{x}\text{-}\mathbf{y}||^2}{2\sigma^2})$$

  with $\sigma > 0$ being the scale parameter.

This study uses a pair of kernel-based classification techniques: Support vector Machine (SVM) and Kernel Fishers Discriminant analysis (KFDA) as the nonparametric classification way-out. In SVM, we choose **Gaussian** and **Polynomial of degree 3** kernels as the mapping function. In our study, SVM using Gaussian kernel is denoted as SVM 1 and SVM using polynomial of degree 3 kernel is denoted as SVM 2. For the comparison purpose between nonparametric classification and parametric classification, we furnish the linear discriminant analysis result too.
Next, we briefly furnish the technicalities of Linear Discriminant Analysis, Support Vector Machine and Kernel Fisher discriminant analysis and Principal component-based classification.

2.1. Linear Discriminant Analysis(LDA)

Suppose there be k classes and i-th class denoted by the class level π_i where $i = 1, 2, ..., k$. The objects are classified based on the vector constructed by p associated random variables, i.e. $\mathbf{X}' = [X_1, X_2 \cdots, X_p]$. The observed values of $\mathbf{X}$ differ from one class to another. f_i and p_i, $i = 1, 2, .., k$ denote the probability density function and the prior probability respectively of i^{th} class. In case of non-availability of prior probabilities, p_i's are assumed to be equiprobable for each class. Let $\mathbf{x}$ be the observation to be assigned among any of the two classes.
In LDA, the parametric assumption to be considered is that the probability density function for any i^{th} family is multivariate Gaussian,i.e. $f_i(x) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$, $i = 1, 2, .., k$, $\boldsymbol{\mu}_i$ and Σ_i being the mean vector and covariance matrix correspond to i^{th} class. Consider that, the covariance matrices are equal for all classes which are Σ, $\Sigma = \Sigma_1 = \Sigma_2 = ... = \Sigma_k$. The linear discriminant scores are thereby calculated through the following formula

$$d_i(\mathbf{x}) = \boldsymbol{\mu}_i'\Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_i'\Sigma^{-1}\boldsymbol{\mu}_i + \ln p_i, \qquad i = 1, 2, ..., k.$$

$\mathbf{x}$ will be allocated to the π_g if the linear score $d_g(\mathbf{x})$= Largest($d_1(\mathbf{x})$,$d_2(\mathbf{x})$,....,$d_k(\mathbf{x})$). In general, the mean vector $\boldsymbol{\mu}_i$ and covariance matrix Σ_i are unknown. For the classification of the training data set, the mean vectors and covariance matrices for each class are estimated from training sample data as $\hat{\mu}_i$ and $\hat{\Sigma}_i$. Σ is estimated by the pooled

estimate, Σ_{pooled} where

$$\Sigma_{pooled} = \frac{1}{n_1 + n_2 + .. + n_k}[(n_1 - 1)\hat{\Sigma}_1 + (n_2 - 1)\hat{\Sigma}_2 + ... + (n_k - 1)\hat{\Sigma}_k].$$

p_i is replaced by the sample proportion $\hat{p}_i = \frac{n_i}{n}$, where $n = n_1 + n_2 + ... + n_k$. The estimated linear discriminant score, then $\hat{d}_i(\mathbf{x})$ is given by

$$\hat{d}_i(\mathbf{x}) = \hat{\boldsymbol{\mu}}_i{}'\Sigma^{-1}\mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\mu}}_i{}'\Sigma^{-1}_{pooled}\hat{\boldsymbol{\mu}}_i + \ln \hat{p}_i, \qquad i = 1, 2, ..., k.$$

$\mathbf{x}$ is allocated to π_g if the linear score $\hat{d}_g(\mathbf{x})$= Largest($\hat{d}_1(\mathbf{x})$,$\hat{d}_2(\mathbf{x})$,.....,$\hat{d}_k(\mathbf{x})$).
R.A. Fisher([8]) proposed an extension of LDA mentioned above, where discrimination between the populations can be done by taking a linear combination of the observed variables, viz., $\boldsymbol{\alpha}'\mathbf{X}$.
Fisher's criterion is the maximization of the following ratio concerning coefficient vector $\boldsymbol{\alpha}$,

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}' B \boldsymbol{\alpha}}{\boldsymbol{\alpha}' V \boldsymbol{\alpha}},$$

where B is the between class sum of square, $B = \sum_{i=1}^{k}(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_i - \bar{\boldsymbol{\mu}})'$ where, $\bar{\boldsymbol{\mu}} = \frac{1}{k}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 + ... + \boldsymbol{\mu}_k)$ and V is the withing class sum of square, $V = \sum_{i=1}^{k}(n_i - 1)\Sigma_i$. Maximum of $J(\boldsymbol{\alpha})$ occurs when $\boldsymbol{\alpha}$ is chosen as the eigen vector of $V^{-1}B$ correspond to largest eigen value λ is the eigenvalue of $V^{-1}B$. This linear function $\boldsymbol{\alpha}'\boldsymbol{x}$ is termed Fisher's discriminant function. This $\boldsymbol{\alpha}'\boldsymbol{x}$ for the choice of eigenvector would coincide with the same linear discriminant we discussed before.

2.2. Support Vector Machine

Vapnik ([17]) first introduced the support vector machine (SVM) which is the stepping stone of the kernel-based classification method. The SVM method is a supervised non-parametric statistical classification technique.
SVM is an optimization technique where the distance between two said populations is modeled to be maximum concerning a reference plain.
Suppose we have n training samples $(\mathbf{x}_i, y_i), i = 1, 2, .., n$ where $\mathbf{x}_i$ is a d-dimensional vector and y_i denotes the class labels which takes values 1 and -1. all d dimensional hyperplane are parameterized by a vector($\mathbf{w}$) and a constant (b), expressed in the equation

$$\mathbf{w}'\mathbf{x} + b = 0,$$

where $\mathbf{w}$ is the (not necessarily normalized) normal vector to the hyperplane. The reference hyperplane separating the data points is

$$f(\mathbf{x}) = sign(\mathbf{w}'\mathbf{x} + b).$$

The decision rules are

- if $f(\mathbf{x}) > 0$, then the observation $\mathbf{x}$ belongs to first class,

- if $f(\mathbf{x}) < 0$, then observation $\mathbf{x}$ belongs to second class.

This algorithm is also generalized for multi-class datasets which are linearly separable by hyperplane. The objective of this algorithm is to separate the different class data points as much as possible and separate them by the hyper-planes. For non-linearly separable classes, the kernel function is used to map the datasets into the higher dimensional feature space to get maximum separation between the classes. In the feature space, hyper-planes are used for separating the classes.

2.3. Kernel Fisher's Discriminant Analysis(KFDA)

Kernel Fisher's Discriminant Analysis [18] is a kernelized version of Linear Discriminant Analysis where kernel function is taken as Gaussian kernel to perform nonlinear mapping on input data set to the high dimensional feature space with linear properties, i.e.,

$$\phi : R^2 \longrightarrow F \Rightarrow x \longrightarrow \phi(x) \ \ \forall x,$$

where ϕ being the mapping function. In the feature space classes are linearly separable classes [Baudat *et al.*, [5]]. Note that the mapped observations are **centered** in the feature space [Schölkopf *et al.*]. As per the logic of Fisher's classification criterion, maximizing the intra-classes inertia and minimizing the within-classes inertia, the following ratio measures the variability within-group values to between-group values variability,

$$[\frac{\boldsymbol{v}'B\boldsymbol{v}}{\boldsymbol{v}'V\boldsymbol{v}}], \tag{1}$$

where V and B are the following intra-classes inertia and inter-classes inertia in the feature space. We have to select $\boldsymbol{v}$ in a way such that the ratio will be maximized. The eigenvector of the largest eigenvalue of $V^{-1}B$ gives the maximum of the above ratio. Because the eigenvectors are linear combinations of feature elements, there exist coefficients α_{pq}, $(p = 1, 2, q = 1, 2, ...n_p)$ for which the coefficient vector $\boldsymbol{v}$ comes as follows.

$$v = \sum_{p=1}^{2} \sum_{q=1}^{n_p} \alpha_{pq} \phi(x_{pq}). \tag{2}$$

Due to the high dimensional structure direct solving of equation (1) is difficult. To address this, dot product kernel $k(x_i, x_j)$ on Hilbert space is employed [Aizerman *et al.*[2], Boser*et al.* ([4])]. For the dot product kernel $k(x_i, x_j)$,

$$k(x_i, x_j) = k_{ij} = \Phi'(x_i)\Phi(x_j), \tag{3}$$

where ϕ is the mapping function. In terms of the dot product kernel, equation (1) can be written as,

$$\lambda = \frac{\boldsymbol{\alpha}'KWK\boldsymbol{\alpha}}{\boldsymbol{\alpha}'KK\boldsymbol{\alpha}}. \tag{4}$$

where K is the kernel matrix, $K = (K_{pq})_{p=1,2,q=1,2}$ where $K_{pq} = (K_{ij})_{i=1,2,...n_p,j=1,2,...n_q}$, W is the block diagonal matrix,$W = (W_l)_{l=1,2}$ where W_l the $(n_l \times n_l)$ matrix whose all terms are equal to $\frac{1}{n_l}$. Applying eigenvalue decomposition on the kernel matrix, the quantity which maximizes (4) will be chosen as $\boldsymbol{\alpha}$. For a detailed study of the process, the readers are recommended to see Boudat *et al.*,([5]). For an arbitrary test point, say $\boldsymbol{z}$, the projection on feature space can be deduced using the following.

$$\boldsymbol{v}'\phi(\boldsymbol{z}) = \sum_{p=1}^{2}\sum_{q=1}^{n_p} \alpha_{pq} k(x_{pq}, \boldsymbol{z}) \tag{5}$$

2.4. Principal Component Analysis

Mathematically speaking, principal components are the linear combination of the p random variables $X_1, X_2, ...X_p$ where each linear combination represents a new coordinate system obtained by rotating the original coordinate system where $X_1, X_2, ...X_p$ represents the coordinate axes. The new coordinate axes represent the direction where the variables exhibit maximum variability. The principal components solely depends on the covariance matrix $\boldsymbol{\Sigma}$ of the variables $X_1, X_2, ...X_p$.
Let $\mathbf{X}' = (X_1, X_2, ...X_p)$ be the random vector and $\boldsymbol{\Sigma}$ is the covariance matrix of the random vector $\boldsymbol{X}$. $\lambda_1 \geq \lambda_2 \geq ..... \geq \lambda_p \geq 0$ are the eigen values of the covariance matrix $\boldsymbol{\Sigma}$.
Consider the linear combinations

$$\begin{aligned}
Y_1 &= \mathbf{b_1'X} = b_{11}X_1 + b_{12}X_2 + ... + b_{1p}X_p \\
Y_2 &= \mathbf{b_2'X} = b_{21}X_1 + b_{22}X_2 + ... + b_{2p}X_p \\
&\quad ..... \\
&\quad ..... \\
&\quad ..... \\
Y_p &= \mathbf{b_p'X} = b_{p1}X_1 + b_{p2}X_2 + ... + b_{pp}X_p
\end{aligned}$$

Then the variance of Y_i's will be $\mathbf{b_i'\Sigma b_i}$ and covariance between Y_i and Y_j will be $\mathbf{b_i'\Sigma b_j}$ where $i = 1(1)p$ and $j = 1(1)p$. The principal components are the uncorrelated linear combinations, namely, $Y_1, Y_2, ..., Y_p$ with $V(Y_i) < \infty$. The choice of the $\boldsymbol{b}_i$'s can be made, subject to the condition $\mathbf{b_i'b_i} = 1$ and $Cov(\mathbf{b_i'X}, \mathbf{b_j'X}) = 0$. Then the first principal component is $Y_1 = \mathbf{b_1'X}$ where $\boldsymbol{b}_1$ is the eigenvector corresponding to the first eigenvalue λ_1. The second principal component is $Y_2 = \mathbf{b_2'X}$ where $\boldsymbol{b}_2$ is the eigenvector of the second eigenvalue λ_2. Similar way, the k^{th} principal component is $Y_k = \mathbf{b_k'X}$ where $\boldsymbol{b}_k$ is the eigen vector of the k^{th} eigen value λ_k.
In our study, we use these k principal components as the substitute of observed variables for the construction of classifiers under different methods of classification.

3. Experiments on Simulated Data

This section deals with two types of simulated experiments. In the first part, we simulate a data set coming from binary classes and thereon we perform linear discriminant analysis, Kernel Fisher's discriminant analysis(KFDA) and support vector machine(SVM) based on original p variables and also on the k principal components,

constructed out of those p variables and explaining 90% data variability. In the second phase of the simulation experiments, we simulate multi-class data sets and thereafter we perform aforementioned classification methods in similar strategy as adopted in binary class classification.

We design the data generation method in a novel way of considering a multiple regression scheme. As already mentioned, the number of principal components k is selected in such a way that at most 90 percent of the total data variability is explained.

The step-by-step simulation method is presented below.

3.1. Simulation Strategy

Primarily, we generate 10000 replications of $\{y_i, x_{1i}, x_{2i}, .., x_{pi}, \epsilon_i\}$ of size n where the choices of n are taken as 50, 100, 300, 500. Then, We deduce the ordinary least square method(OLS) for estimating the coefficients of conventional, multiple regression model having p regressors.

$$Y^* = \beta_1 + \beta_2 X_1 + \beta_2 X_2 + ... + \beta_{p+1} X_p + \epsilon.$$

$\hat{\beta}_{OLS} = (\hat{\beta}_{1_{OLS}}, \hat{\beta}_{2_{OLS}}, .., \hat{\beta}_{p+1_{OLS}})$ is the OLS estimates obtained from the 10000 such simulations. For the entire simulation framework, we consider the choices of variables $p = 4(1)10$, 15, 20. The structure of the simulation framework is

(1) Fix n. For two variables, says X_1 and X_2, we collect data from Bivariate normal distribution, i.e. we choose, $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim BN(0, 0, 1, 1, 0.3)$ while $\epsilon \sim N(0, 1)$.

(2) Here the linear regression model is $Y^* = \beta_1 + \beta_2 X_1 + \beta_2 X_2 + \epsilon$. The initial values of the parameters are takes as $\beta_1 = 1, \beta_2 = 2$ and $\beta_3 = 3$. Then we use Monte Carlo simulation process for the estimation of parameters.

(3) Next we again generate $2n$ number (we use this $2n$ number of samples into training samples and test samples, each with size n) of X_1, X_2 and ϵ in the same way. Using the OLS estimates of the parameters, subsequently we obtain $2n$ number of the estimated value of Y^*.

(4) For **binary-class classification setup**, The classification rule is defined as Y(a new binary variable)=1 if $Y^* > mean(Y^*)$ and $Y = 0$ otherwise. 0 and 1 denote population 1 and 2 respectively.

(5) In **multi-class classification setup**, we choose a **three-class** and **five-class** classification frame. In both three-class and five-class frameworks, the data generation process is exactly same as previously described binary classification framework. The changes are made only in defining the classification rules for three and five-class data sets. For three class classifications(denoted by 1,2, 3) the classification rules are defined as

$$\begin{aligned} Y &= 1 \quad \text{if} \quad Y* < Q[0.33] \\ &= 2 \quad \text{if} \quad Q[0.33] \leq Y* < Q[0.66] \\ &= 3 \quad \text{otherwise.} \end{aligned}$$

For five class classifications, the classification rules are defined as

$$\begin{aligned} Y = 1 \;\; & \text{if} \;\; Y* < Q[0.20] \\ = 2 \;\; & \text{if} \;\; Q[0.20] \leq Y* < Q[0.40] \\ = 3 \;\; & \text{if} \;\; Q[0.40] \leq Y* < Q[0.60] \\ = 4 \;\; & \text{if} \;\; Q[0.60] \leq Y* < Q[0.80] \\ = 5 \;\; & \text{otherwise.} \end{aligned}$$

$Q[p]$ represents the p^{th} percentile of $Y*$.

(6) Following the above rule, after separating all $2n$ number of data sets into two classes or three classes or five classes using the classification rule we split this $2n$ number of samples into training samples and test samples, each with size n for efficiency checking of the classification mechanism.

(7) Perform LDA, KFDA and SVM based on set of generated X_1 and X_2 and calculated misclassification error.

(8) Repeat steps 3 to 6 for 5000 times and calculate the mean misclassification error recorded in Table 1.

(9) Next, derive the principal components constructed by the two variables($p = 2$) and again perform LDA, KFDA and SVM using principal components and calculate the misclassification error. This process is repeated 5000 times and the average of misclassification error is calculated and recorded in Table 1. SVM1 in the table indicate SVM method performed through polynomial degree 3 kernel while SVM2 denotes SVM done by Gaussian kernel.

(10) For $p \geq 3$, the $X_1, X_2, ..., X_p$ variables are generated from multivariate normal distribution. For the construction of dispersion matrix(Σ), first, we build a matrix A whose elements are generated from $Uniform(0, 0.5)$, then construct $\Sigma = A^T A$ is used as a dispersion matrix and also $\epsilon \sim N(0, 1)$.

(11) Fix the initial starting value $\beta = (\beta_1, \beta_2, ..., \beta_p) = (1, 2, 3, ..., p)$, say.

(12) Using the initial value of $\boldsymbol{\beta}$ in regression equation, again $\boldsymbol{X}$ values can be generated, and therefore Y^*. As mentioned in step 4, the classification between two groups is done the values of Y^*.

(13) The total sample is divided into training and test data sets with equal size.

(14) LDA, KFDA and SVM are performed on the training and test data sets and misclassification errors are calculated.

(15) The steps (11), (12) and (13) repeated 5000 times, and mean classification error rates are calculated for each variable for each value of n.

(16) Next, for $p \geq 3$, we perform principal component analysis using first k($<$ p) principal components. k is the minimum number of principal components required for explaining 90 percent of the total variability in data.

(17) Again we perform LDA, KFDA as well as SVM based the first k principal component and calculate the misclassification error rate. Mean misclassification error rates is found 5000 such sets. All the misclassification error recorded on original variable based classification and PCA based classification are reported in Table 1, Table 6 , Table 7, Table 8, Table 9, among which last four possess the detailed result for $n = 50, 100, 300, 500$.

For the visual idea on how the simulated observations within classes clutter around, we present plots (Figure 1) on train data for $p = 2$ in binary, three-class and five-class respectively.

Figure 1.: Train data plot

Validation of the classification technique is to be checked on test samples too via misclassification error in order to evaluate their performance. When the probability distribution of parent population is known completely, misclassification probability can be identified through the conditional probabilities calculation.
Therefore, misclassification error for an observation X =P($X \in \pi_1$ and is misclassified) + P($X \in \pi_2$ and is misclassified).
Since in most of the cases, the parent population structure is not known, sample misclassification error is considered for such scenario. The apparent error rate is a sample estimate of misclassification errors that does not depend on the form of the parent populations and that can be calculated for any classification procedure. The apparent error rate is easily calculated from the confusion matrix. Lower the value of misclassification error better is the efficiency of the classification method.

Table 1, presented below, reports minimum classification error under original variable based classification for each sample size (n=50,100,300,500). The variable value (p) for which minimum misclassification error rate occur is mentioned within the parenthesis at the bottom of each figure. Also, misclassification error due to PC based classification along with **number of principal component(PC)** explaining 90% data variability is furnished in Table1. Detailed values are elaborated in Table 6, Table 7, Table 8 and Table 9 provided in the Appendix section.

In order to portray the difference between variable based classification and PC-based classification, Figure 2 and Figure 3 are presented below with reference to only KFDA classification methods for binary class, three and five class. The label KFLD along X axis denotes linear discriminant in kernel Fisher's classification. As expected PCA based classification roughly shows more compactness in separating the points.

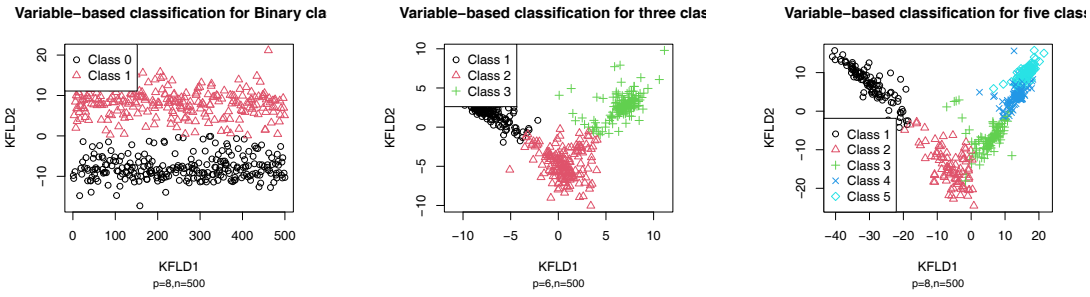

Figure 2.: Variable-based classification plot for KFDA method

		Original variables-based				PC-based			
No.of Class	n	LDA	KFDA	SVM 1	SVM 2	LDA	KFDA	SVM 1	SVM 2
Binary	50	0.08248 (p=5)	0.22581 (p=5)	0.18064 (p=4)	0.01237 (p=9)	0.06037 (PC=2)	0.08204 (PC=2)	0.17424 (PC=2)	0.11642 (PC=3)
	100	0.06163 (p=6)	0.09921 (p=5)	0.13879 (p=6)	0.05114 (p=5)	0.04087 (PC=3)	0.05805 (PC=2)	0.12712 (PC=3)	0.05051 (PC=2)
	300	0.03765 (p=7)	0.04396 (p=7)	0.07405 (p=7)	0.02373 (p=8)	0.02943 (PC=3)	0.03303 (PC=3)	0.07531 (PC=3)	0.03624 (PC=3)
	500	0.03144 (p=8)	0.02130 (p=8)	0.06417 (p=8)	0.02048 (p=8)	0.03794 (PC=3)	0.04101 (PC=3)	0.06025 (PC=3)	0.03593 (PC=3)
Three	50	0.15178 (p=5)	0.34820 (p=7)	0.24736 (p=7)	0.11200 (p=6)	0.11863 (PC=2)	0.23842 (PC=3)	0.29092 (PC=3)	0.13684 (PC=3)
	100	0.11913 (p=5)	0.12775 (p=5)	0.19710 (p=5)	0.09032 (p=5)	0.12416 (PC=2)	0.10170 (PC=2)	0.24692 (PC=2)	0.10931 (PC=2)
	300	0.09182 (p=6)	0.06798 (p=6)	0.13239 (p=7)	0.06215 (p=8)	0.09517 (PC=3)	0.08590 (PC=3)	0.20695 (PC=3)	0.07539 (PC=3)
	500	0.08673 (p=8)	0.05284 (p=6)	0.11356 (p=8)	0.04039 (p=8)	0.08280 (PC=3)	0.06042 (PC=3)	0.16085 (PC=3)	0.07721 (PC=3)
Five	50	0.24745 (p=5)	0.38606 (p=9)	0.40724 (p=8)	0.19758 (p=5)	0.17203 (PC=2)	0.50911 (PC=3)	0.47062 (PC=3)	0.37474 (PC=2)
	100	0.19286 (p=6)	0.26646 (p=5)	0.34894 (p=7)	0.15080 (p=6)	0.15657 (PC=3)	0.25158 (PC=2)	0.43768 (PC=3)	0.19102 (PC=3)
	300	0.13704 (p=7)	0.12814 (p=6)	0.26107 (p=7)	0.09168 (p=7)	0.12338 (PC=3)	0.12410 (PC=3)	0.34313 (PC=3)	0.15518 (PC=3)
	500	0.12227 (p=7)	0.10486 (p=8)	0.21886 (p=8)	0.07920 (p=8)	0.11339 (PC=3)	0.09991 (PC=3)	0.29068 (PC=3)	0.15101 (PC=3)

Table 1.: Comparative table of misclassification errors of optimum p for different methods under two techniques

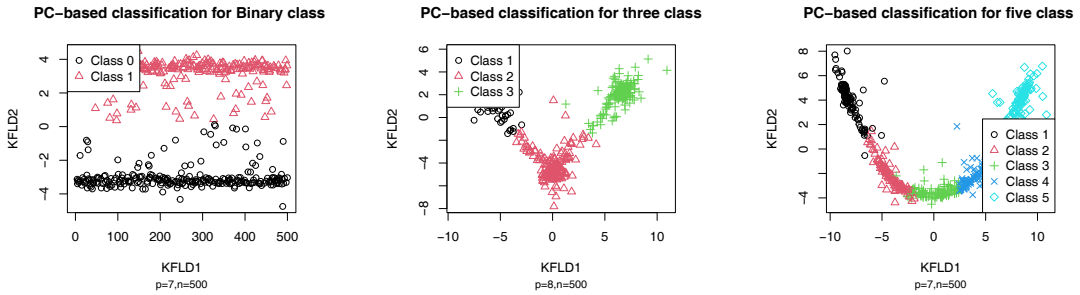

Figure 3.: PC-based classification plot for KFDA method

Comparison between classification by original variables and principal components

Table1 above reflects few important points as depicted below.

- Quite intuitively, in binary, as well as in multi-class classification, misclassification error rates, decrease when the number of observations increases.
- For fixed n, in binary and also in multi-class classification under every method of classification technique, when the number of variables(p) increases, misclassification error rates drops up to an optimum p and after that specific p it starts to increase again. However, value of that optimum p differ for binary class classification, three-class classifications and also for the five-class classification. For instance, in LDA process under $n = 300$, $p = 7$ was the optimum number of variables having the least misclassification error(0.0456) in binary classification while in three class classifications for same n, this optimum value is $p = 6$ and for five-class classifications this optimum p turns 7.
- It is observed that for all three different class classifications(LDA, SVM, KFDA) when $n = 500$, the optimum p for every method of classification is 8 except for a few stray cases.
- In binary and multi-class classifications, for all three methods of classification, misclassification error rates are minimal when the classification rule is defined using k principal components compared to the classification rule is defined using p variables.
- Not always PCA based classifications are the most effective one, particularly for the combination of higher p, higher n and higher no. of classes(see Table1 misclassification error under SVM2 for class 5, $p = 8$,$n = 500$) variable classification turns well worth. More specifically, as hinted in introduction, for poorly-posed or ill-posed condition ($n \to \infty$ as well as $p \to \infty$) no matter whether under parametric or nonparametric, PCA based classification, may not yield lower misclassification error because of inflated S^{-1}.

4. Experiment on Real-life Data-set

4.1. Small/Medium Sample Size Dataset

For checking the performance of three above mentioned classification techniques, we choose Wheat Seed dataset obtained from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets/seeds`). The dataset contains information on three distinct varieties of wheat, viz, Kama, Rosa and Canadian, based on seven agronomic morphological variables V1, V2, V3, V4, V5, V6 and V7. The total sample size is 210 consisting of 70 random observations from each variety. Each unit

is characterized using a soft X-ray technique and high-quality visualization, in favor of those seven attribute for every observation.

Figure 4.: Correlation Plot

The correlation matrix (Figure 4) indicates that multi-colinearity exists among the variables in the seed dataset (e.g. V2, V1, V4). As per the underlying distribution of three classes concerned, Henze-Zirkler multivariate normality test (package "mvnormalTest" in R 4.4.1) shows that including three classes, the first two classes fail to meet the test but for 3rd class null hypothesis is accepted as multivariate normal. The p-values for three classes are 1.987299^{-14}, 2.940114^{-08} and 0.1448198 respectively. In this experiment, we divide the dataset by random mechanism into training and test sets with a 70%:30% split ratio. This splitting process is replicated 5000 times and the mean misclassification error is recorded accordingly.
For each iteration, we conduct three different classification approaches- variable-based classification and PCA-based classification through Linear Discriminant Analysis (LDA), Kernel Fisher Discriminant Analysis (KFDA) and Support Vector Machine (SVM).

This experiment aims to evaluate the performance of aforementioned classification techniques, executed on the Wheat seed dataset. Table 2 and Table 3 display mean misclassification error rates .

Methods	LDA	KFDA	SVM1	SVM2
Variable-Based	0.03663	0.13715	0.1404921	0.07106

Table 2.: Misclassification Error Rates of by different classification rules

	90%				99%			
Methods	LDA	KFDA	SVM1	SVM2	LDA	KFDA	SVM1	SVM2
PC based	0.10644	0.13224	0.15715	0.11179	0.06583	0.12823	0.15715	0.11083

Table 3.: Misclassification Error Rates of by different classification rules

Under the variable-based classification approach, both LDA and SVM2 techniques exhibit lower misclassification error rates, outperforming the KFDA and SVM1 methods.
In contrast, under principal component-based classification, when a benchmark of 90% of total explained variability is fixed, two Principal Components (PCs) are sufficient to capture the variability. However, when the cutoff is further increased to 99%, classifications through three PCs show the improved misclassification error rates in every method of classification. It is observed that increasing number of PCs does not harbinger to attain significantly lower misclassification error rates compared with variable-based classification.
In summary, the experimental results reveal that LDA and SVM2 perform well in the variable-based classification approach. Also, under the need of 99% total explained variability, five PCs are sufficient in constructing classification rule.

4.2. Large Sample Size Data-set

For the large sample size dataset, we choose dry beans dataset which is obtained from UCI Machine Learning Repository (`https://doi.org/10.24432/C50S4B`). In this dataset, seven distinct types of dry beans, namely, Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira, are selected for a comprehensive examination. An advanced computer vision system was devised to investigate **sixteen** various attributes on these beans such as Area (A), Perimeter (P), Major axis length (L), Minor axis length (l), Aspect ratio (K), Eccentricity (Ec), Convex area (C), Equivalent diameter (Ed), Extent (Ex), Solidity (S), Roundness (R), Compactness (CO), Shape Factor1 (SF1), Shape Factor2 (SF2), Shape Factor3 (SF3), Shape Factor4 (SF4) of seven different varieties of dry beans. The computer vision system enables in deriving meaningful information from digital images and extract various feature details about the image. A total image of 13,611 individual grains(Seker: # 2023, Barbunya: # 1323, Bombay: # 523, Cali: # 1631, # Hozor: # 1929, Sira: # 2637 and Dermason: # 3547), collected from the seven registered dry bean varieties was captured using a high-resolution camera. The notation # inside the parenthesis indicates the total number of observations under a particular class. At first we standardize the whole dataset. Then we present the correlation matrix (Figure 5 constructed for those sixteen variables.

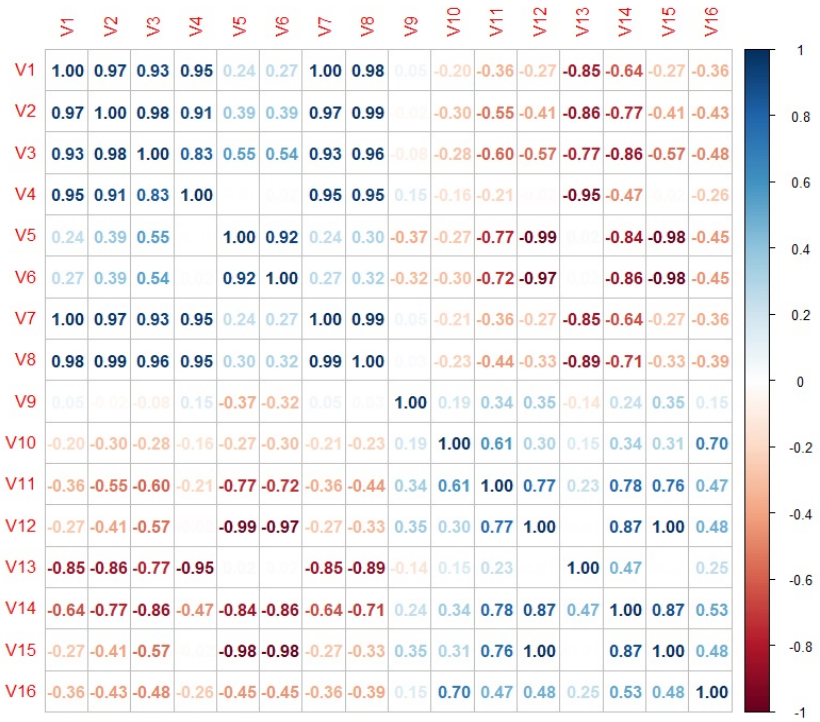

Figure 5.: Correlation Plot

The correlation matrix indicates that strong multicollinearity exists among few variables in the dry beans dataset. Henze-Zirkler multivariate normality test establishes that the seven populations indicating seven varieties of dry beans are not distributed as multivariate normal (p-value $<10^{-11}$) for all classes). Parametric methods of classification, for example, LDA method may not be a good choice for classification in such non-Gaussian case. However, we still choose to use parametric classification LDA for the comparative purpose along with the non-parametric method of classification (KFDA, SVM) under two different approaches.
The sampling scheme is designed as follows. From each variety we select randomly 500 observations and thereby merge them into a single dataset containing atotal of 3500 randomly selected observations ready for the experiment. Then, we split the dataset randomly into training and test datasets with a 70%-30% split ratio and thereafter we perform LDA, QDA and KFDA methods of classification under two different approaches: variable-based and PCA-based. These methods are replicated over 5000 times and the misclassification error rate are calculated and recorded in the following tables.

Methods	LDA	KFDA	SVM1	SVM2
Variable-Based	0.08385	0.06509	0.09023	0.07531

Table 4.: Misclassification Error Rates of by different classification rules

By the merit of mean misclassification error rates of the three different approaches, the variable-based classification approach performed better. In both cases, KFDA and SVM2 methods of classification show comparatively lower misclassification error rates than other methods of classification.

	90%				99%			
Methods	LDA	KFDA	SVM1	SVM2	LDA	KFDA	SVM1	SVM2
PC based	0.12902	0.11528	0.14975	0.11927	0.08342	0.06569	0.08348	0.08173

Table 5.: Misclassification Error Rates of by different classification rules

In PCA-based classification, when we set the benchmark at 90% of data variability, three PCs are involved in the classification and for 99% of explained data variability seven PCs are involved. When we consider the data variability 99%, instead of 90%, the misclassification error rates decrease as well as the performance of PCA-based classification comes close to variable-based classifications.
KFDA method of classification turned out the best well under variable-based classification approach but under PC-based classification KFDA performs best only when the benchmark of total explained variability is set to ($\geq$ 99%). This implies that in presence of large number of observations (n) and large number of variables (p) PCA based classification may result unsatisfactory.

5. Conclusion

This article unravels some important highlights regarding the connection between the number of total observations (n) with respect to the variables under study (p) in study of statistical classification. For example, in both binary and multiclass classification when the total number of observations is large, say, 500, the optimal number of variables (p) giving the best separation under LDA, SVM and KFDA comes up as 8 or 7. Similarly, for $p < 5$, n could be roughly 100 to 150 to capture a sufficiently distinct classification, both in binary and multi-class problems. Again, for $p \geq 5$, the minimum number of observations required is 200 to achieve a better classification. Generally speaking, for $p > 7$, the minimum requirement of observations should be 400. In comparison to parametric classification techniques, non-parametric KFDA demands a larger number of observations in case of a higher value of p for better clarification among the classes. Loosely speaking, in parametric as well as nonparametric classification for binary and multi-class classifications no of principal components used in classification should be at least half of variables to obtain least misclassification error.

Secondly, the intuitive belief on the supremacy of classification by PC-based classification technique does not hold good in presence of large p. In such "poorly-posed" or "ill-posed" situation variable based classification,in general is recommended. More superior result in PCA based classification may be achieved by regularizing the sample covariance matrix S , prior to investigation. Instead of using simply S in the classifier, ridge like estimate $(S + \gamma I_p)$ may be incorporated so that the inverse, i.e., $(S + \gamma I_p)^{-1}$ would give smaller mean square error than S^{-1}([16],[10]). This shrinkage technique could also handle the multicollinearity issue. Incorporation of shrunken centroid similar to ridge regression strategy in kernel Fisher discriminant analysis or in support vector machine may be of worth investigating in future.

Conflict of Interest

The authors confirm that this article content has no conflict of interest.

References

[1] Anderson, T.W.(1984). An Introduction to Multivariate Statistical Analysis, New York, Wiley, Third Edition.

[2] Aizerman M. A., Braverman E. M., Rozonoér L. I.(1964). Theoretical foundations of the potential function method in pattern recognition learning, Automation and Remote Control, 25, 821-837.

[3] Bai, Z.(1999). Methodologies in spectral analysis of large dimensional random matrices. Statistica Sinica, 9, 611-677.

[4] Boser B. E., Guyon I. M., Vapnik V. N.(1992). A training algorithm for optimal margin classifiers. In D.Haussler edited vol on 5th Annual ACM Workshop on COLT, 144-152, Pittsburgh, PA.

[5] Baudat, G, and Anouar, F.(2000). Generalized discriminant analysis using a kernel approach, Neural Computation, 12.10, 2385-2404.

[6] Cortes C, Vapnik V. (1995). Support vector machine. Machine learning, Sep; 20(3), 273-297.

[7] Donghwan Kim (2017). **kfda**: *Kernel Fisher Discriminant Analysis*, R package version >= 3.0.0, URL: https://github.com/ainsuotain/kfda.

[8] Fisher, Ronald A. (1936). The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2), 179–188.

[9] Fletcher, T., (2009). Support vector machines explained. Tutorial paper, 1118, 1-19.

[10] Friedman, J.H.(1989). Regularized discriminant analysis. Journal of American Statistical Association. 84,165-175.

[11] Hall, P., Marron J.S., Park B.U., (1992). Smoothed Cross-Validation, Probability Theory and Related Fields, 92, 1-20.

[12] Johnson, R. A. and Wichern, D. W., (1982). Applied Multivariate Statistical Analysis, Prentice Hall India Learning Private Limited; 6th edition.

[13] Kecman, V., (2005). Support vector machines–an introduction, In Support vector machines: theory and applications (pp. 1-47). Berlin, Heidelberg: Springer Berlin Heidelberg.

[14] Scholkopf B., Smola A., Muller K. R. (1998). Nonlinear Component Analysis as A Kernel Eigenvalue Problem, Neural Computation, 10, 1299-1319.

[15] Simonoff, J. S. (1996). Smoothing Methods in Statistics. Springer-Verlag. New York.

[16] Thompson, L.A., Davis, W., Young, P.H., Young, D. M. & Hill, J.S.(2019). A comparison of regularized linear discriminant functions for poorly-posed classfication problems. Journal of Data Science, 17(1), 1-36.

[17] Vapnik, V & Cortess, C. (1995). Support-vector networks, Machine Language, Vol 20, Issue 3, 273-29.

[18] Yang, J., Jin, Z., Yang, J. Y., Zhang, D., and Frangi, A. F. (2004). Essence of kernel Fisher discriminant: KPCA plus LDA. Pattern Recognition, 37(10), 2097-2100.

APPENDIX

		LDA		KFDA		SVM 1		SVM 2	
No.of Class	Variables (p)	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based
Binary	4	0.07849	0.05747	0.54151	0.11833	0.18064	0.17424	0.09166	0.07942
	5	0.08248	0.06037	0.22581	0.08204	0.18162	0.16512	0.07770	0.08040
	6	0.08633	0.05145	0.26252	0.11085	0.18832	0.16130	0.08904	0.09948
	7	0.09261	0.05632	0.28563	0.18337	0.18692	0.15580	0.06888	0.06934
	8	0.09933	0.06203	0.29703	0.17335	0.18780	0.16254	0.09856	0.08674
	9	0.10587	0.05955	0.32148	0.21018	0.19610	0.15672	0.01237	0.11642
	10	0.11319	0.06011	0.34624	0.34146	0.18710	0.15976	0.14174	0.15876
	15	0.14419	0.07298	0.46933	0.45407	0.19434	0.16626	0.34366	0.16202
	20	0.17552	0.07653	0.49921	0.49120	0.20772	0.18836	0.42704	0.24022
Three	4	0.15379	0.15162	0.470255	0.19902	0.27908	0.22528	0.15330	0.10938
	5	0.15178	0.11863	0.35458	0.24928	0.25194	0.30070	0.12220	0.14068
	6	0.15890	0.12603	0.36890	0.15949	0.25564	0.32254	0.11200	0.13684
	7	0.16820	0.10974	0.34820	0.23842	0.24736	0.29092	0.12104	0.16688
	8	0.17630	0.12777	0.36262	0.27453	0.27046	0.32350	0.16428	0.18720
	9	0.18545	0.12193	0.37828	0.39484	0.25682	0.33042	0.14764	0.17720
	10	0.19586	0.12049	0.41758	0.41192	0.25208	0.32794	0.19296	0.24692
	15	0.24047	0.13783	0.58754	0.53289	0.27452	0.37680	0.41876	0.34222
	20	0.28418	0.13881	0.68232	0.62234	0.26852	0.39110	0.61338	0.38064
Five	4	0.27843	0.18836	0.51232	0.31154	0.42402	0.48808	0.23124	0.29304
	5	0.24745	0.17203	0.48623	0.27965	0.42388	0.47176	0.19758	0.37474
	6	0.25428	0.17813	0.46338	0.51183	0.41578	0.47356	0.25500	0.34042
	7	0.26386	0.18622	0.43563	0.24180	0.41642	0.53220	0.20836	0.39152
	8	0.27726	0.17363	0.44652	0.32796	0.40724	0.47062	0.29812	0.39230
	9	0.28919	0.19715	0.38606	0.50911	0.41684	0.52056	0.29828	0.38652
	10	0.30281	0.19344	0.45188	0.51190	0.42320	0.53082	0.32370	0.43356
	15	0.35467	0.20806	0.69409	0.57130	0.44202	0.58644	0.59850	0.50318
	20	0.40193	0.23637	0.80943	0.70555	0.43448	0.60762	0.71778	0.59112

Table 6.: Comparative table of misclassification errors for different methods under two techniques when n=50

		LDA		KFDA		SVM 1		SVM 2	
No.of Class	Variables (p)	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based
Binary	4	0.06319	0.05927	0.11476	0.07757	0.14809	0.13541	0.06334	0.04608
	5	0.06274	0.04493	0.09921	0.05805	0.14254	0.13607	0.05114	0.05051
	6	0.06163	0.04087	0.13468	0.08183	0.13879	0.12712	0.05829	0.04760
	7	0.06505	0.05302	0.28591	0.05198	0.14418	0.11858	0.05757	0.05834
	8	0.06834	0.04227	0.30233	0.08111	0.14101	0.11899	0.05643	0.06399
	9	0.07249	0.05147	0.30124	0.07990	0.15015	0.10750	0.06212	0.07082
	10	0.07688	0.04895	0.33103	0.08441	0.14103	0.10944	0.06139	0.07110
	15	0.09621	0.04960	0.43290	0.42376	0.15795	0.11266	0.19100	0.10893
	20	0.11335	0.05357	0.49086	0.47970	0.16101	0.12889	0.34274	0.18509
Three	4	0.12822	0.10942	0.49525	0.13270	0.20491	0.18575	0.14251	0.10923
	5	0.11913	0.12416	0.12775	0.10170	0.19710	0.24692	0.09032	0.10931
	6	0.11951	0.09480	0.28172	0.11350	0.20274	0.24600	0.09091	0.14632
	7	0.12309	0.10013	0.35837	0.09160	0.21379	0.25288	0.10703	0.09951
	8	0.12821	0.09974	0.35143	0.14623	0.20841	0.26994	0.09663	0.17472
	9	0.13401	0.10527	0.37105	0.15362	0.20194	0.28050	0.10749	0.14205
	10	0.13936	0.10331	0.37927	0.20914	0.21087	0.27296	0.11392	0.13431
	15	0.16538	0.11076	0.46855	0.48179	0.21477	0.30161	0.20126	0.25188
	20	0.18987	0.11542	0.65997	0.57398	0.21384	0.30946	0.44349	0.27761
Five	4	00.21436	0.19692	0.43138	0.22686	0.37410	0.39628	0.18879	0.33341
	5	0.19790	0.18045	0.26646	0.25158	0.36446	0.39892	0.15359	0.22391
	6	0.19286	0.15657	0.48112	0.15951	0.35497	0.40348	0.15080	0.19102
	7	0.20034	0.14242	0.41156	0.16558	0.34894	0.43768	0.16145	0.18469
	8	0.20851	0.15275	0.40905	0.14217	0.36921	0.44186	0.20514	0.25281
	9	0.21605	0.15974	0.43386	0.28873	0.37109	0.43974	0.18048	0.16492
	10	0.22382	0.16461	0.37659	0.20566	0.35583	0.44435	0.19657	0.30087
	15	0.26275	0.16507	0.56416	0.52251	0.36647	0.47676	0.37434	0.40874
	20	0.29760	0.17852	0.78880	0.59544	0.37894	0.51519	0.50781	0.51630

Table 7.: Comparative table of misclassification errors for different methods under two techniques when n=100

		LDA		KFDA		SVM 1		SVM 2	
No.of Class	Variables (p)	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based
Binary	4	0.05033	0.05247	0.05381	0.05714	0.09607	0.09443	0.04659	0.05298
	5	0.04389	0.03686	0.04666	0.04566	0.09239	0.08067	0.04841	0.05633
	6	0.03765	0.02943	0.05503	0.02971	0.08740	0.07934	0.03104	0.04294
	7	0.03939	0.02369	0.04396	0.03303	0.07405	0.07531	0.02929	0.03846
	8	0.04009	0.02648	0.04570	0.04190	0.08235	0.06119	0.02373	0.03624
	9	0.04169	0.02545	0.24977	0.03272	0.09143	0.05977	0.03518	0.06042
	10	0.04376	0.02948	0.33269	0.03214	0.08487	0.05900	0.03039	0.06622
	15	0.05387	0.03174	0.36221	0.22175	0.08494	0.05822	0.05400	0.06903
	20	0.06267	0.03866	0.48261	0.43799	0.09662	0.06650	0.16300	0.10731
Three	4	0.10646	0.13475	0.10294	0.08654	0.15835	0.20570	0.11869	0.09452
	5	0.10390	0.09505	0.10450	0.08282	0.13615	0.20019	0.07072	0.10349
	6	0.09182	0.09517	0.06798	0.08590	0.13584	0.20677	0.07037	0.10779
	7	0.09278	0.08538	0.08947	0.05539	0.13239	0.20695	0.06534	0.10135
	8	0.09316	0.08861	0.12989	0.06778	0.13358	0.18539	0.06215	0.07539
	9	0.09497	0.08438	0.39061	0.06717	0.13692	0.20716	0.06423	0.05448
	10	0.09678	0.09032	0.41067	0.09934	0.14081	0.20903	0.06916	0.08892
	15	0.10697	0.08835	0.42966	0.11586	0.13988	0.22611	0.10216	0.15427
	20	0.11689	0.08703	0.59237	0.51623	0.14410	0.22641	0.21003	0.16514
Five	4	0.17623	0.18309	0.19032	0.18824	0.30429	0.34191	0.18029	0.17121
	5	0.15182	0.14520	0.19985	0.15473	0.26851	0.31504	0.11064	0.12369
	6	0.14195	0.13560	0.12814	0.12410	0.26702	0.33980	0.10776	0.11868
	7	0.13704	0.12338	0.16741	0.11117	0.26107	0.34313	0.09168	0.15518
	8	0.14051	0.14931	0.35814	0.09124	0.26169	0.33432	0.10777	0.17122
	9	0.14418	0.12045	0.42204	0.12037	0.26237	0.34058	0.10464	0.19738
	10	0.14820	0.11642	0.43683	0.13925	0.26067	0.33134	0.10267	0.16697
	15	0.16923	0.13056	0.45794	0.40122	0.27454	0.37132	0.19748	0.31545
	20	0.18830	0.12491	0.69775	0.55377	0.27958	0.39554	0.40920	0.36720

Table 8.: Comparative table of misclassification errors for different methods under two techniques when n=300

		LDA		KFDA		SVM 1		SVM 2	
No.of Class	Variables (p)	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based	Var.-based	PC-based
Binary	4	0.05119	0.04117	0.05579	0.05453	0.08411	0.08602	0.04523	0.04711
	5	0.03735	0.03138	0.04368	0.03241	0.07556	0.07219	0.03217	0.05176
	6	0.03273	0.02556	0.03922	0.02952	0.07541	0.06337	0.02780	0.03984
	7	0.03234	0.03165	0.02197	0.01360	0.08993	0.06292	0.02635	0.03654
	8	0.03144	0.03794	0.02130	0.04101	0.06417	0.06025	0.02048	0.03593
	9	0.03277	0.02122	0.02459	0.02391	0.06801	0.04415	0.02857	0.03338
	10	0.03431	0.02173	0.31059	0.02637	0.07109	0.04272	0.02610	0.03124
	15	0.04147	0.02484	0.20928	0.13026	0.07163	0.04387	0.05146	0.05070
	20	0.04816	0.02903	0.46177	0.44994	0.07417	0.04787	0.08708	0.09025
Three	4	0.11968	0.11152	0.10177	0.09733	0.15350	0.17805	0.07928	0.08914
	5	0.09182	0.09411	0.07269	0.08748	0.10869	0.16992	0.05841	0.06324
	6	0.08739	0.09003	0.05284	0.06042	0.11441	0.18778	0.04695	0.07289
	7	0.08685	0.09045	0.06462	0.06072	0.11478	0.19441	0.05008	0.07741
	8	0.08673	0.08280	0.07124	0.04721	0.11356	0.16085	0.04089	0.07721
	9	0.08753	0.08336	0.07685	0.05412	0.11699	0.18504	0.05183	0.04514
	10	0.08801	0.08362	0.43593	0.04595	0.12239	0.18881	0.05282	0.06645
	15	0.09364	0.08482	0.42103	0.06614	0.11637	0.20065	0.07588	0.10159
	20	0.10001	0.08440	0.55964	0.51454	0.12249	0.19850	0.12774	0.13863
Five	4	0.20356	0.19412	0.18369	0.14225	0.27120	0.32013	0.13113	0.15128
	5	0.13953	0.13037	0.14297	0.14032	0.24229	0.28055	0.13107	0.14833
	6	0.12695	0.12902	0.11604	0.10851	0.22530	0.27713	0.13581	0.09486
	7	0.12227	0.11339	0.14577	0.08106	0.22616	0.24300	0.09250	0.13474
	8	0.12361	0.11183	0.10486	0.09991	0.21886	0.29068	0.07920	0.15101
	9	0.12493	0.11535	0.29307	0.07082	0.22716	0.28114	0.09269	0.14174
	10	0.12738	0.11572	0.45560	0.09096	0.22346	0.28302	0.09666	0.12587
	15	0.14209	0.11404	0.46604	0.44149	0.23228	0.32161	0.11333	0.21826
	20	0.15554	0.11755	0.64108	0.54978	0.24131	0.35950	0.27407	0.34658

Table 9.: Comparative table of misclassification errors for different methods under two techniques when n=500