Pattern Recognition and Machine Learning (EE552) Project 2

Savinay Nagendra (sxn265)

February 13, 2018

Abstract

Supervised learning problems are formulated in applications where the training data comprises of input vectors along with their corresponding target vectors. If the target labels are discrete, the process is called **Classification**. Classification is used to assign a class or category K to an unknown data vector using the observed data X and their corresponding known category labels. The input space is divided into decision regions whose boundaries are called **decision surfaces**. In this report, two linear classification algorithms have been explored in the context of classifying data of three different data sets. The three data sets used are :
1) Wine
2) Wallpaper
3) Taiji (Motion Capture)

The first linear classification algorithm uses the method of **Least-Squares** to construct a multi-class classifier using a single K class discriminant scheme. The method of least squares gives a **closed form solution**, which can be used to predict the class of an unseen data point.

Loss-less feature space dimension reduction is a crucial process in classification problems. The classification algorithms applied on the reduced feature space are computationally inexpensive. **Fisher Projection** is used to achieve well separated clusters of data points in the reduced feature space, which ensures enhanced classification efficiency. **Maximum a Priori Estimation** (MAP), which is a classifier using **Decision theory** is implemented on the new data set with a reduced feature space dimensionality.

The two algorithms have been quantitatively and comprehensively compared in terms of their performance efficiency and accuracy.

Contents

1 Introduction

Classification is the problem of identifying the category that a new observation belongs to, on the basis of a training set of data and their corresponding known category labels.[1] The goal of Classification, at the highest level is to take an input vector x and assign it to one of K discrete classes C_i, where $i = 1, ..., K$. The classes are usually taken to be disjoint and the input space is divided into unique decision regions, so that each input is assigned to only one class. Classification is achieved by learning decision boundaries between classes, which can be used to make decisions on assigning a category to an unseen data point. Data sets whose classes can be separated exactly by linear decision surfaces (hyper-planes) are said to be linearly separable.

In this report, two Linear Classification algorithms, namely:
1) **Least Squares Classification** on original data
2) **Classification based on Decision Theory (Maximum a posteriori Estimation)** on data generated using **Fisher's Projection**[1]

Reducing dimension of feature space, but still being able to retain all the necessary information with minimum loss factor will help ensure that the process of classification is computationally inexpensive. Fisher Projection for reduced dimensionality of feature space has also been explored on all three data sets as a pre-processing step before applying the MAP Classifier.[3]
The organization of the report is as follows:

Section 2 contains the Approach taken to complete this project, with subsections explaining about the Data used, Method of Least Squares Classification, Fisher's Projection and Decision Theory Classification. Section 3 contains observations and results of each method denoting test and train accuracies and classification visualizations. Section 4 contains conclusions and inferences based on results of Section 3. Section 5 contains references and bibliography

2 Approach

In this section, approach taken to solve the given problem is explained. All the three data sets have been explained. The methods of classification used have been mathematically derived.

2.1 Data

Data sets used in Classification problems have three parameters:
1) Number of Classes (K)
2) Number of Features/Dimensions (D
3) Number of samples of each category.($N_k, \quad k = 1, ..., K$

$$\sum_{k=1}^{K} N_k = N$$

A data frame is divided into train and test data. These are tuples having a feature Matrix and corresponding label vector.
$Data = [FeatureMatrix, \quad LabelVector]$

In this report, three data sets have been used:

2.1.1 Wine

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivators. The analysis determines the quantities of 13 constituents found in each of the three types of wines. The attributes are
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

number of Classes: 3
number of Training Observations: 90
number of Features: 13
number of samples per class in train dataset:
1 30
2 36
3 24

number of Test Observations: 88
number per class in test dataset:
1 29
2 35
3 24

2.1.2 Wallpaper Group

This dataset consists of the features extracted from images containing the 17 Wallpaper Groups.

number of Classes: 17
number of Training Observations: 1700

number of Features: 500
number of samples per class in train dataset:

P1	100
P2	100
PM	100
PG	100
CM	100
PMM	100
PMG	100
PGG	100
CMM	100
P4	100
P4M	100
P4G	100
P3	100
P3M1	100
P31M	100
P6	100
P6M	100

number of Test Observations: 1700
number of samples per class in test dataset:

P1	100
P2	100
PM	100
PG	100
CM	100
PMM	100
PMG	100
PGG	100
CMM	100
P4	100
P4M	100
P4G	100
P3	100
P3M1	100
P31M	100
P6	100
P6M	100

2.1.3 Taiji Pose

This is a dataset of the joint angles (in quaternions) of 35 sequences from 4 people performing Taiji. The '0' label corresponds to non-translational frames and the non '0' labels correspond to translational frames.

number of Classes: 8
number of Training Observations: 6995
number of Features: 64
number of samples per class in train dataset:

0	1091
1	656
2	1312
3	656
5	656
6	1312
7	656
9	656

number of Test Observations: 3932
number of samples per class in test dataset:

0	611
1	269
2	738
3	369
5	369
6	738
7	369
9	369

2.2 Methods

2.2.1 Least-Squares Classification

[1] Minimization of a sum-of-squares error function leads to a closed form solution for parameter values. Even though Least squares method is generally used for regression problems, it works well on Classification problems as well.

Let us consider a general classification problem with K classes, with a 1-of-K binary coding scheme for the target vector $\mathbf{t}$, i.e., $\mathbf{t}$ is a square matrix, where its columns denote the classes/categories. All the elements in a column are 0s except the ones corresponding to the features that belong to the class denoted by that column. This method approximates the conditional expectation $E[\mathbf{t}|x]$ of the target values given the input vector x. For binary coding scheme, this conditional expectation is given by the vector of posterior class probabilities.

Each class C_k is defined by a linear discriminant model:

$$y_k(x) = w_k^T x + w_{k0}, \tag{1}$$

where w_{k0} is the bias, which indicates the intercept of the linear discriminant. Grouping the vectors into a Matrix form,

$$y(x) = \tilde{X}\tilde{W} \tag{2}$$

$$\text{where} \quad \tilde{W} = \quad = \quad \begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1K} \\ w_{20} & w_{21} & \cdots & w_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(D+1)0} & w_{(D+1)1} & \cdots & w_{(D+1)K} \end{bmatrix} \in \quad \mathbb{R}^{(D+1)\times K}$$

$$\text{and} \quad \tilde{X} \quad = \quad \begin{bmatrix} 1 & x_{11} & \cdots & x_{1D} \\ 1 & x_{21} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{bmatrix} \quad \in \quad \mathbb{R}^{N\times(D+1}$$

A new input x is assigned to the class for which $y_k = \tilde{x}\tilde{W}$ is the largest
Parameter matrix $\tilde{W}$ is determined by minimizing the sum-of-squares error function,

$$E_D(\tilde{W}) = \frac{1}{2}Tr\Big[(\tilde{X}\tilde{W} - T)^T(\tilde{X}\tilde{W} - T)\Big] \tag{3}$$

To minimize the error function in (3), We compute $\frac{\partial E(\tilde{W})}{\partial \tilde{W}} = 0$

$$\begin{aligned} \frac{\partial E(\tilde{W})}{\partial \tilde{W}} &= \frac{\partial}{\partial \tilde{W}}\Big[(\tilde{X}\tilde{W} - T)^T(\tilde{X}\tilde{W} - T)\Big] = 0 \\ &\Rightarrow \frac{\partial}{\partial \tilde{W}}\Big[(\tilde{W}^T\tilde{X}^T - T^T)(\tilde{X}\tilde{W} - T)\Big] = 0 \\ \tilde{W}&\frac{\partial}{\partial \tilde{W}}\Big[(\tilde{W}^T\tilde{X}^T\tilde{X}\tilde{W} - \tilde{W}^T\tilde{X}^TT - T^T\tilde{X}\tilde{W} + T^TT\Big] = 0 \\ &\Rightarrow \frac{\partial}{\partial \tilde{W}}\Big[(\tilde{X}\tilde{W})^T\tilde{X}\tilde{W} - (T^T\tilde{X}\tilde{W}) - (T^T\tilde{X}\tilde{W})^T + T^TT\Big] = 0 \end{aligned} \tag{4}$$

To simplify (4) we use the following axioms:

$$\begin{aligned} \frac{\partial x^Tx}{\partial x} &= 2x \\ \frac{\partial ax}{\partial a} &= x^T \end{aligned} \tag{5}$$

Using the above two axioms to simplify (4),

$$\begin{aligned} &\Rightarrow 2\tilde{X}\tilde{W}\tilde{X}^T - 2\tilde{X}^TT = 0 \\ &\Rightarrow \tilde{X}^T\tilde{X}\tilde{W} = \tilde{X}^Tt \end{aligned} \tag{6}$$

Hence, $\tilde{W}^*$ that minimizes $E(\tilde{W})$ is:

$$\tilde{W}^* = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^TT \tag{7}$$

(7) gives the optimal closed form solution, which is used to predict class labels for unseen data.
A single K class discriminant comprising of K linear functions of the form (1) aids in

avoiding unambiguous regions. A new point x is assigned to class C_k if $y_k(x) > y_j(x)$ for all $j \neq k$.

The decision boundary between class C_k and C_j is therefore given by $y_k(x) = y_j(x)$, which corresponds to a $(D-1)$ dimensional hyperplane defined by

$$(w_k - w_j)^T x + (w_{k0} - w_{j0}) = 0 \tag{8}$$

(8) has been used to plot decision boundaries between classes in the code.

Even though the least squares method gives the exact closed form solution for the discriminant function parameters, the approach is not robust for outlier data points. This will reduce the efficiency of classification when such points are present in the data. Also, the predictions made by using 1 of K coding scheme cannot be assumed to be probabilities since they are not inherently constrained to lie in the in the interval $(0, 1)$.
Hence, adopting Discriminative or Generative probabilistic models will give better classification efficiency than least squares.

2.2.2 Fisher's Linear Discriminant Analysis

[1]

In this section, Fisher's Discriminant analysis is explored in the context of multi-class classification.

Linear models of Classification have always been preferred for mathematical convenience and the flexibility of computation. Linear classification can be viewed in the perspective of feature space dimensionality reduction. Projecting data to a lower dimension can lead to considerable loss of information. Classes that are well separated in the original D-dimensional space may become strongly overlapping in a lower dimensional subspace, and may also become linearly inseparable. However, reducing the dimension of data using a method which does not incur losses, will result in computational efficiency, accuracy, ease of visualization and linear separability. This very concept is the motivation of Fisher's Linear Discriminant Analysis.

The goal of Fisher's Model is two fold:
1) **Maximize the separation between inter class projected means**
2) **Minimize the intra-class or within class variance**
Hence, the idea is to find a projection where the projected data has a maximum separation between inter-class means while simultaneously has a minimum intra-class variance. Consider a multi-class Classification problem with K classes. Let C_i and C_j be any two of the K classes having N_i and N_j points respectively. Let X be a D-dimensional feature matrix. The cost function must be constructed to encompass both the constraints. Hence, the solution proposed by Fisher is to maximize a function that represents the difference between the inter-class means, normalized by a measure of the intra-class variance. This function is given by:

$$J(w) = \frac{(m_i - m_j)^2}{s_i^2 + s_j^2} \tag{9}$$

The mean vectors of the two classes are therefore given by:

$$m_i = \frac{1}{N_i} \sum_{n \in C_i} x_n \qquad m_j = \frac{1}{N_j} \sum_{n \in C_j} x_n \tag{10}$$

Here, x_n is D-dimensional vector, which corresponds to a row of the feature matrix.
In general, $m_k \in \mathbb{R}^{D \times 1}$, where $k = 1, ..., K$

Let $D^{'}$ be the dimension of the subspace onto which which the original feature matrix is to be projected. Hence, there will be $D^{'}$ linear features of the form $y_k = w_k^T x$, where $k = 1, ..., D^{'}$. The vectors w_k can be considered to be the columns of a matrix W so that

$$y = XW \tag{11}$$

Here, y is the projected feature matrix in the $D^{'}$ dimensional feature space.
$X \in \mathbb{R}^{N \times D}$ $W \in \mathbb{R}^{D \times D^{'}}$

We can define a measure of the scatter in multivariate feature space in terms of Scatter Matrices:

$$S_W = \sum_{k=1}^{K} \sum_{n \in C_i} (x_n - m_k)(x_n - m_k)^T \tag{12}$$

Here S_W is the intra-class scatter (variance) matrix
Now, the scatter of the projection y can be expressed as a function of the scatter matrix in feature space x.

$$\begin{aligned} s_k^2 &= \sum_{n \in C_k} (y_n - m_k)^2 \\ &= \sum_{n \in C_k} (w^T x - w^T m_k)^2 \\ &= \sum_{n \in C_k} w^T (x - m_k)(x - m_k)^T w \\ &= w^T S_i w \end{aligned} \tag{13}$$

From 12, we know that $S_W = \sum_{k=1}^{K} S_k$ Using this result, the total intra class variance is

$$\sum_{k=1}^{K} s_k^2 = w^T S_W w \tag{14}$$

Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space.

$$(m_i - m_j)^2 = (w^T m_i - w^T m_j)^2 = w^T (m_i - m_j)(m_i - m_j)^T w = w^T S_B w \tag{15}$$

where S_B is the inter-class Scatter Matrix. Hence, from equation 9, the Fisher function can be expressed as

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{16}$$

To find the optimal W which maximizes $J(w)$, we differentiate 16 and equate it to zero.

$$\begin{aligned}
\frac{d}{dw}[J(w)] = \frac{d}{dw}\Big[\frac{w^T S_B w}{w^T S_W w}\Big] \\
&\Rightarrow \frac{(w^T S_W w)(2S_B w) - (w^T S_B w)(2S_W w)}{(w^T S_W w)^2} = 0 \\
&\Rightarrow (w^T S_W w S_B w) = (w^T S_B w S_W w) \\
&\Rightarrow S_B w = \lambda S_W w, \quad (since \quad w^T S_W w \quad and \quad w^T S_B w \quad are \quad scalars). \\
&\Rightarrow w = S_W^{-1}(m_i - m_j)
\end{aligned} \tag{17}$$

The above derivation uses this axiom:

$\frac{d}{da} a^T X a = 2Xa \quad$ if $a = a^T$

For K classes, the inter-class Scatter matrix S_B can be generalized as

$$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T, \quad where \quad m = \frac{1}{N}\sum_{k=1}^{K} N_k m_k \tag{18}$$

Steps to find the Projection Matrix W:

1. Using equations 12 and 18, find the matrices S_W and S_B respectively.
2. Compute the eigen values and eigen vectors of the matrix $S_W^{-1} S_B$
3. Sort the eigen values and corresponding eigen vectors in descending order.
4. Select a subspace dimension $D^{'}$ to which the feature matrix is to be projected.
5. Select the first $D^{'}$ eigen vectors. These constitute the columns of the projection matrix W.

Hence, Fisher's Linear Discriminant Analysis aids in well separated clusters of data points belonging to one class in the projected feature space, thereby improving linear separability and classification efficiency.

We can observe the following: S_B is composed of the sum of K matrices, each of which is a product of two vectors, and therefore of rank 1. Only (K-1) of these matrices are independent. Hence, there are at most (K-1) non-zero eigen values. This implies that the projection onto a (K-1) dimensional subspace spanned by the eigen vectors of S_B does not alter the value of $J(w)$. Hence, a maximum of (K-1) linear features can be found by this method.

2.2.3 Probabilistic Generative Models for classification using Decision Theory

[2] After the pre-processing step using Fisher's Linear Discriminant Analysis, the projected data is used for Classification by the approach of Generative Probabilistic Model - **Maximum a Posteriori Estimation**.
It is assumed that the class-conditional densities are Gaussian and all classes share the same covariance matrix. Thus, the likelihood function for each class C_k is given by a Multivariate Gaussaian Density:

$$p(x|C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{\frac{1}{2}}} exp\Big[-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)\Big] \tag{19}$$

where, μ_k : Mean of the inputs for category k.
Σ : Covariance matrix (common to all classes)

Using Baye's rule, The posterior probability of class k, given input x is:

$$p(c_k|x) = \frac{likelihood.prior}{marginal} = \frac{p(x|C_k) * \pi_k}{p(x)} \tag{20}$$

Here, π_k is the prior probability of each class k.
Maximizing the posterior probability in 20 will indicate the class that the input x belongs to.
Let us absorb every expression which does not depend on k into a constant $C^{'}$,

$$p(C_k|x) = C^{'} \pi_k exp(-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-mu_k) \tag{21}$$

Taking logarithm on both sides:

$$\log p(C_k|x) = \log C^{'} + \log \pi_k - \frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k) \tag{22}$$

The goal is to maximize 22 over k,

$$\begin{aligned} argmax_k \quad & \log \pi_k - \frac{1}{2}[x^T \Sigma^{-1} x + \mu^T \Sigma^{-1} \mu_k] + x^T \Sigma^{-1} \mu_k \\ \Rightarrow argmax_k \quad & C^{"} + \log \pi_k - \frac{1}{2}\mu_k \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k \end{aligned} \tag{23}$$

Hence, an objective function can be defined:

$$\delta_k(x) = \log \pi_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k \tag{24}$$

At an input x, the category corresponding to the highest $\delta_k(x)$ can be predicted.

The decision boundary between two classes refers to the set of points in which two classes

are equally probable, i.e., (one-vs-one) scheme.

$$\begin{aligned} \delta_k(x) &= \delta_l(x) \\ \Rightarrow log\pi_k - \tfrac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + x^T\Sigma^{-1}\mu_k &= log\pi_l - \tfrac{1}{2}\mu_l^T\Sigma^{-1}\mu_l + x^T\Sigma^{-1}\mu_l \end{aligned} \tag{25}$$

$$w_0 = -\frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \frac{1}{2}\mu_l^T\Sigma^{-1}\mu_l + \log\frac{\pi_k}{\pi_l} \qquad ; \qquad w = \Sigma^{-1}(\mu_k - \mu_l)$$

The resulting decision boundaries correspond to surfaces along which the posterior probabilities $p(C_k|x)$ are constant, which will be given by linear functions of x. Hence, the decision boundaries are linear in input space. The prior probabilities $p(C_k)$ enter through the bias parameters, so that priors have the effect of making parallel shifts of the decision boundary.

For the general case of K classes, the decision boundary can also be defined as (one vs all scheme)

$$\begin{aligned} a_k(x) &= w_k^T x + w_{k0}, \\ where \qquad w_k &= \Sigma^{-1}\mu_k \\ w_{k0} &= -\frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k \end{aligned} \tag{26}$$

3 Results

This section deals with the results obtained by implementing the fore mentioned classifiers on the three data sets. Confusion Matrices, Classification Matrices and Accuracies for training and testing data sets have been included. Visualizations of the classification decision boundaries in the two dimensional feature space have also been portrayed.

3.1 Least-Squares Classification

Even though Least-Squares method is usually used for Regression problems, it gives good results on Classification problems as well. Advantages of using Least-Squares for Classification are:

1. This method gives a simple Closed-form solution for the parameter matrix W.
2. The constructed error function is a simple quadratic convex function and thereby leads to a simple optimization with a global minimum.
3. Gives Linear Discriminant Functions
4. It uses one-of-K encoding, which is simple to construct.

The disadvantages are:

1. The method is very sensitive to outlier data points.

2. Magnitudes of weights cannot be assumed to probabilities, since they are not constrained be in the range $(0, 1)$

3.1.1 Wine Dataset

Wine data set has 13 features and 3 classes. Two features - 1 and 7 have been chosen out of the 13 features for convenience of visualization. The plot of test data points in a 2-D subspace are shown in figure 1. We can observe that the points are overlapping. Each class is not a well separated cluster of points in the 2 dimensional subspace. Figures 2 to 5 show the confusion and classification matrices for wine dataset. From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 100%.
The testing accuracy for this data set is 99.047%.
Even though the train accuracy is 100%, which could be interpreted as over-fitting, the test accuracy is also very high. Hence, the factor of over-fitting is very less and the least square classifier performs well on wine dataset.
Figure 6 shows the linear discriminants constructed using one vs one scheme over the test data points in figure 1. It can be visualized that the linear boundaries separate the classes well, aiding in a good test accuracy.

Points for Wine data set.png

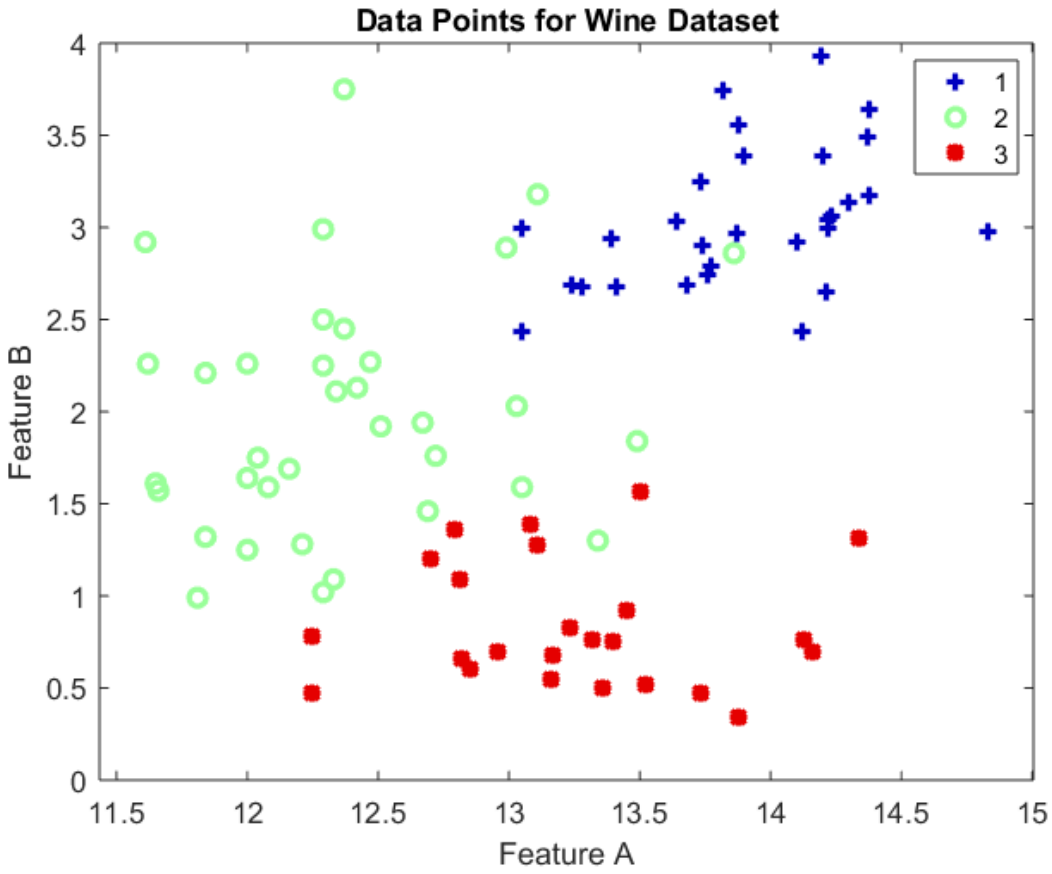

Figure 1: Test Data Points in a 2-D subspace for wine data set using features 1 and 7.

		Predicted Labels		
		Class 1	Class 2	Class 3
Actual Labels	Class1	30	0	0
	Class 2	0	36	0
	Class 3	0	0	24

Figure 2: Confusion Matrix for Train Data set

		Predicted Labels		
		Class 1	Class 2	Class 3
Actual Labels	Class1	1	0	0
	Class 2	0	1	0
	Class 3	0	0	1

Figure 3: Classification Matrix for Train Data set

		Predicted Labels		
		Class 1	Class 2	Class 3
Actual Labels	Class1	29	0	0
	Class 2	0	34	1
	Class 3	0	0	24

Figure 4: Confusion Matrix for Test Data set

		Predicted Labels		
		Class 1	Class 2	Class 3
Actual Labels	Class1	1	0	0
	Class 2	0	0.971429	0.028571
	Class 3	0	0	1

Figure 5: Classification Matrix for Test Data set

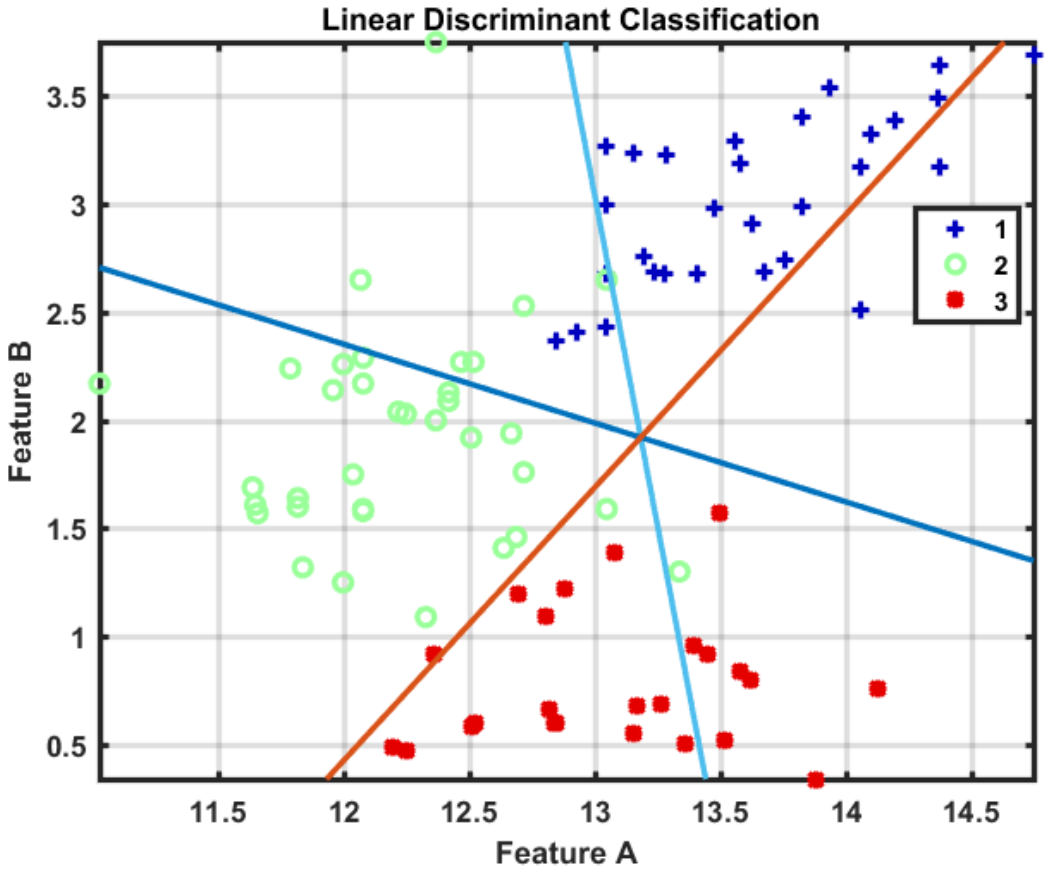

Figure 6: Linear Discriminant Boundaries over Test Dataset in 2-D subspace constructed using one-vs-one scheme.

3.1.2 Wallpaper Groups Dataset

Wallpaper data set has 500 features and 17 classes. Figure 7 shows the plot of test data points of all 17 classes in a two dimensional subspace. Two features have been selected for the convenience of visualization. It is evident that these points are completely overlapping and not linearly separable in the 2-D subspace. This proves that it is redundant to reduce the feature dimension from 500 to 2, which will result in loss of information and linear inseparability.
Figures 8 to 11 show the confusion and classification matrices for wine dataset. From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 96.76%.
The testing accuracy for this data set is 61.71%.
It can be observed that in the case of wallpaer dataset, unlike wine dataset, the test accuracy is very less than the training accuracy. This implies that the classifier is performing very well on the train data set and its efficiency is less on the test dataset, which is the consequence of **over-fitting**.

Figure 12 shows the linear discriminants constructed using one vs one scheme over the test data points shown in figure 7. For a feature space of 2 and 17 classes, the visualization is poor and does not convey useful information.

Note:
For mathematical convenience and manipulation, the string classes in Wallpaper dataset has been converted to array of numbers from 1 to 17. The mapping is as follows:

P1	1
P2	2
PM	3
PG	4
CM	5
PMM	6
PMG	7
PGG	8
CMM	9
P4	10
P4M	11
P4G	12
P3	13
P3M1	14
P31M	15
P6	16
P6M	17

Points for Wallpaper data set.png

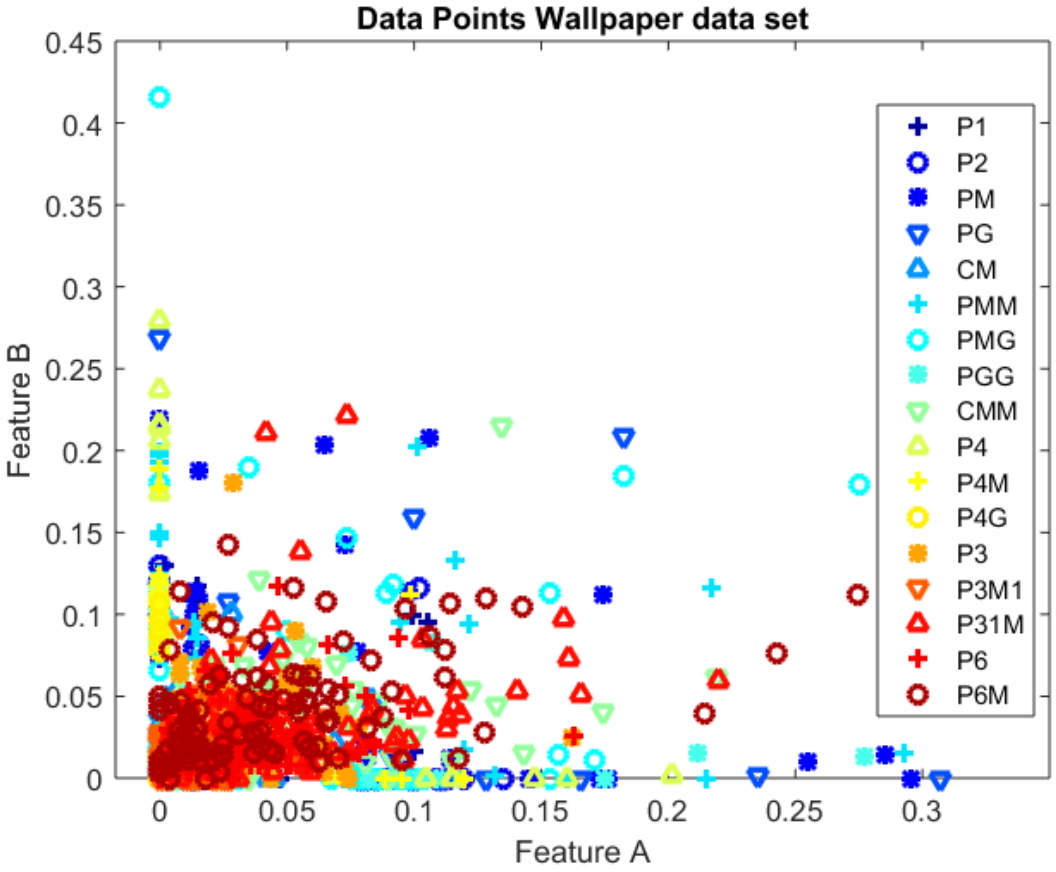

Figure 7: Test Data Points in a 2-D subspace for wallpaper data set using features 1 and 7.

		PREDICTED LABELS																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
ACTUAL LABELS	P1	96	1	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0
	P2	2	90	1	5	0	0	0	0	0	0	1	0	0	0	1	0	0
	PM	1	0	97	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	PG	2	2	0	91	0	0	1	3	0	1	0	0	0	0	0	0	0
	CM	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
	PMG	0	0	3	0	0	0	97	0	0	0	0	0	0	0	0	0	0
	PGG	0	0	1	2	0	0	0	91	0	0	0	6	0	0	0	0	0
	CMM	0	0	0	0	3	0	0	0	97	0	0	0	0	0	0	0	0
	P4	1	1	0	0	0	1	0	0	0	95	2	0	0	0	0	0	0
	P4M	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
	P3	1	0	0	0	1	0	0	0	0	0	0	0	97	0	1	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	1	0	99	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	3
	P6M	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	99

Figure 8: Confusion Matrix for Train Data set

		PREDICTED LABELS																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
ACTUAL LABELS	P1	0.96	0.01	0	0.02	0	0	0	0	0	0.01	0	0	0	0	0	0	0
	P2	0.02	0.9	0.01	0.05	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0
	PM	0.01	0	0.97	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0
	PG	0.02	0.02	0	0.91	0	0	0.01	0.03	0	0.01	0	0	0	0	0	0	0
	CM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	PMG	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0
	PGG	0	0	0.01	0.02	0	0	0	0.91	0	0	0	0.06	0	0	0	0	0
	CMM	0	0	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0
	P4	0.01	0.01	0	0	0	0.01	0	0	0	0.95	0.02	0	0	0	0	0	0
	P4M	0	0	0	0	0	0	0	0	0	0	0.99	0.01	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	P3	0.01	0	0	0	0.01	0	0	0	0	0	0	0	0.97	0	0.01	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.99	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.97	0.03
	P6M	0	0	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0.99

Figure 9: Classification Matrix for Train Data set

		PREDICTED LABELS																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
ACTUAL LABELS	P1	13	19	4	23	1	5	2	4	3	13	2	2	1	4	4	0	0
	P2	12	19	2	16	1	10	7	9	0	10	3	2	3	1	1	2	2
	PM	2	2	62	0	0	3	26	0	1	1	0	1	1	0	1	0	0
	PG	8	11	2	37	0	4	6	13	0	5	3	4	1	3	3	0	0
	CM	0	0	0	0	90	0	0	0	2	0	0	0	4	2	0	2	0
	PMM	9	6	7	3	3	36	9	3	3	8	11	0	0	0	0	0	2
	PMG	0	2	14	2	0	2	71	1	0	0	3	2	0	0	0	1	2
	PGG	1	4	0	6	1	0	4	46	0	0	0	37	0	0	0	1	0
	CMM	0	3	2	0	16	1	0	0	64	0	0	0	1	0	4	5	4
	P4	5	15	0	10	0	3	1	3	1	28	20	5	5	1	0	1	2
	P4M	0	2	0	2	0	11	0	0	1	10	70	3	0	0	0	0	1
	P4G	0	0	0	0	0	0	0	7	0	0	0	93	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	1	0	0	0	72	11	15	1	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	2	98	0	0	0
	P31M	0	1	0	0	0	0	0	0	0	1	0	0	12	1	82	3	0
	P6	1	1	1	0	0	0	0	0	0	0	0	0	2	0	2	86	7
	P6M	0	0	4	0	0	0	0	0	4	1	0	0	0	0	0	9	82

Figure 10: Confusion Matrix for Test Data set

		PREDICTED LABELS																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
ACTUAL LABELS	P1	0.13	0.19	0.04	0.23	0.01	0.05	0.02	0.04	0.03	0.13	0.02	0.02	0.01	0.04	0.04	0	0
	P2	0.12	0.19	0.02	0.16	0.01	0.1	0.07	0.09	0	0.1	0.03	0.02	0.03	0.01	0.01	0.02	0.02
	PM	0.02	0.02	0.62	0	0	0.03	0.26	0	0.01	0.01	0	0.01	0.01	0	0.01	0	0
	PG	0.08	0.11	0.02	0.37	0	0.04	0.06	0.13	0	0.05	0.03	0.04	0.01	0.03	0.03	0	0
	CM	0	0	0	0	0.9	0	0	0	0.02	0	0	0	0.04	0.02	0	0.02	0
	PMM	0.09	0.06	0.07	0.03	0.03	0.36	0.09	0.03	0.03	0.08	0.11	0	0	0	0	0	0.02
	PMG	0	0.02	0.14	0.02	0	0.02	0.71	0.01	0	0	0.03	0.02	0	0	0	0.01	0.02
	PGG	0.01	0.04	0	0.06	0.01	0	0.04	0.46	0	0	0	0.37	0	0	0	0.01	0
	CMM	0	0.03	0.02	0	0.16	0.01	0	0	0.64	0	0	0	0.01	0	0.04	0.05	0.04
	P4	0.05	0.15	0	0.1	0	0.03	0.01	0.03	0.01	0.28	0.2	0.05	0.05	0.01	0	0.01	0.02
	P4M	0	0.02	0	0.02	0	0.11	0	0	0.01	0.1	0.7	0.03	0	0	0	0	0.01
	P4G	0	0	0	0	0	0	0	0.07	0	0	0	0.93	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	0.01	0	0	0	0.72	0.11	0.15	0.01	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.98	0	0	0
	P31M	0	0.01	0	0	0	0	0	0	0	0.01	0	0	0.12	0.01	0.82	0.03	0
	P6	0.01	0.01	0.01	0	0	0	0	0	0	0	0	0	0.02	0	0.02	0.86	0.07
	P6M	0	0	0.04	0	0	0	0	0	0.04	0.01	0	0	0	0	0	0.09	0.82

Figure 11: Classification Matrix for Test Data set

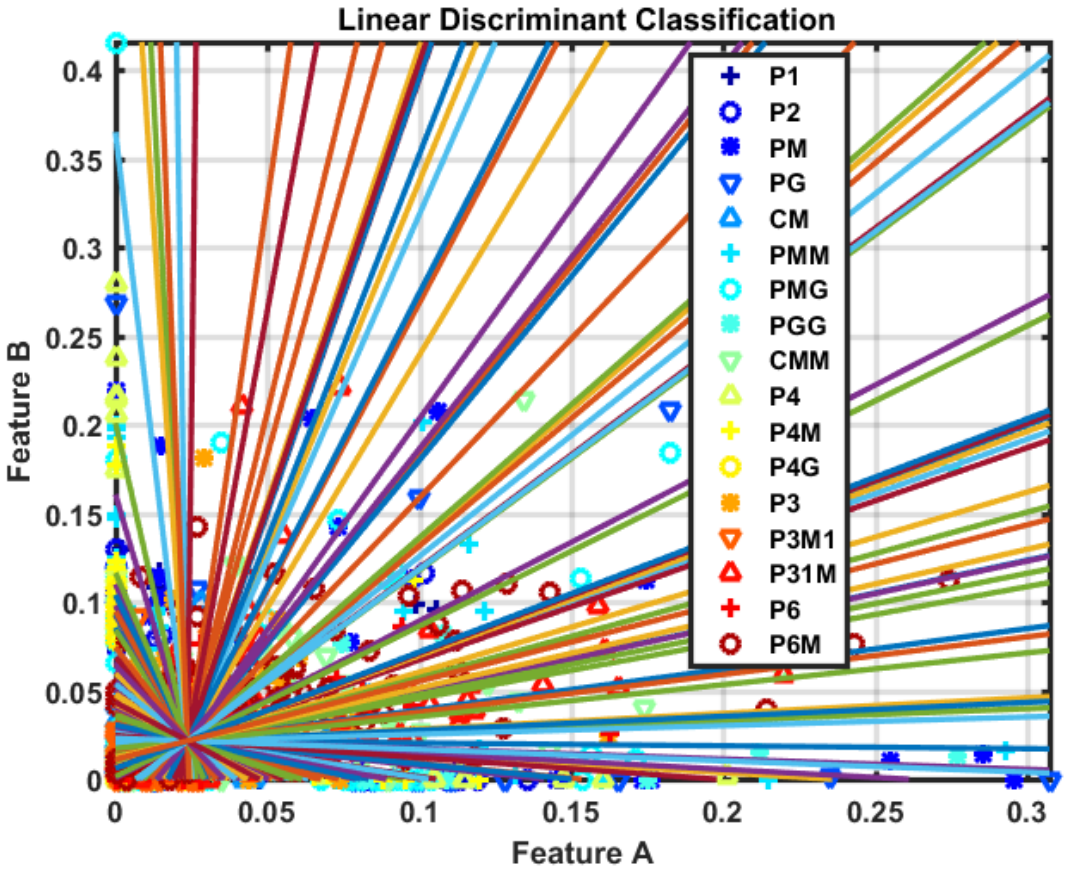

Figure 12: Linear Discriminant Boundaries over Test Dataset in 2-D subspace constructed using one-vs-one scheme.

3.1.3 Taiji Pose Dataset

Taiji data set has 64 features and 8 classes. Figure 13 shows the plot of test data points of all 8 classes in a two dimensional subspace. Two features have been selected for the convenience of visualization. It is evident that the reduction of dimensionality from 64 to 2 has made the data points linearly inseparable in the two dimensional subspace. This proves that it is redundant to reduce the feature dimension from 64 to 2, which will result in loss of information and linear inseparability. Infact, the data points belonging to class label '0' is spread out in the feature space, all other classes overlapping over it.
Figures 14 to 17 show the confusion and classification matrices for wine dataset. From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 88.68%.
The testing accuracy for this data set is 96.52%.
It can be observed that, unlike wallpaper dataset, the training accuracy is lesser than test accuracy. This proves that the designed classifier is not over-fitting the data and hence, has a better performance efficiency than the wallpaper dataset classifier.
Figure 18 shows the linear discriminants constructed using one vs one scheme over the test data points shown in figure 13. For a feature space of 2 and 8 classes, the visualization is poor and does not convey useful information.
For mathematical convenience, discontinuous numbering of classes in Taiji Dataset has been made to a continuous array from 1 to 8. The mapping is as follows:

0	1
1	2
2	3
3	4
5	5
6	6
7	7
9	8

Points for Taiji data set.png

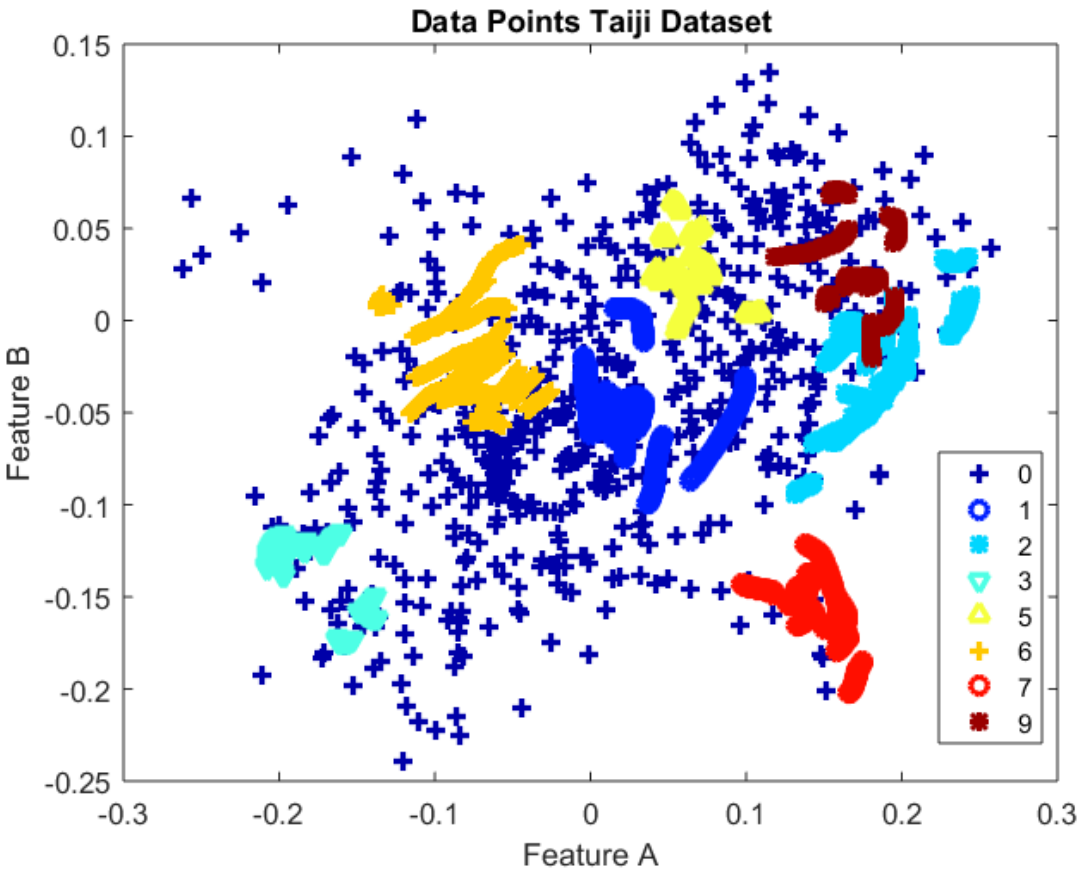

Figure 13: Test Data Points in a 2-D subspace for wallpaper data set using features 1 and 7.

ACTUAL LABELS	PREDICTED LABELS							
	0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
0	787	39	63	33	41	45	24	59
1	0	656	0	0	0	0	0	0
2	0	0	1312	0	0	0	0	0
3	0	0	0	656	0	0	0	0
5	0	0	0	0	656	0	0	0
6	0	0	0	0	0	1312	0	0
7	0	0	0	0	0	0	656	0
9	0	0	0	0	0	0	0	656

Figure 14: Confusion Matrix for Train Data set

ACTUAL LABELS	PREDICTED LABELS							
	0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
0	0.72136	0.03575	0.05775	0.03025	0.03758	0.04125	0.022	0.05408
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	1

Figure 15: Classification Matrix for Train Data set

ACTUAL LABELS	PREDICTED LABELS							
	0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
0	190	55	38	34	131	62	33	68
1	0	369	0	0	0	0	0	0
2	0	0	738	0	0	0	0	0
3	0	0	0	369	0	0	0	0
5	0	0	0	0	369	0	0	0
6	0	0	0	0	0	738	0	0
7	0	0	0	0	0	0	369	0
9	80	0	0	0	0	0	0	289

Figure 16: Confusion Matrix for Test Data set

ACTUAL LABELS	PREDICTED LABELS							
	0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
0	0.31097	0.09002	0.06219	0.05565	0.2144	0.10147	0.05401	0.11129
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
9	0.2168	0	0	0	0	0	0	0.7832

Figure 17: Classification Matrix for Test Data set

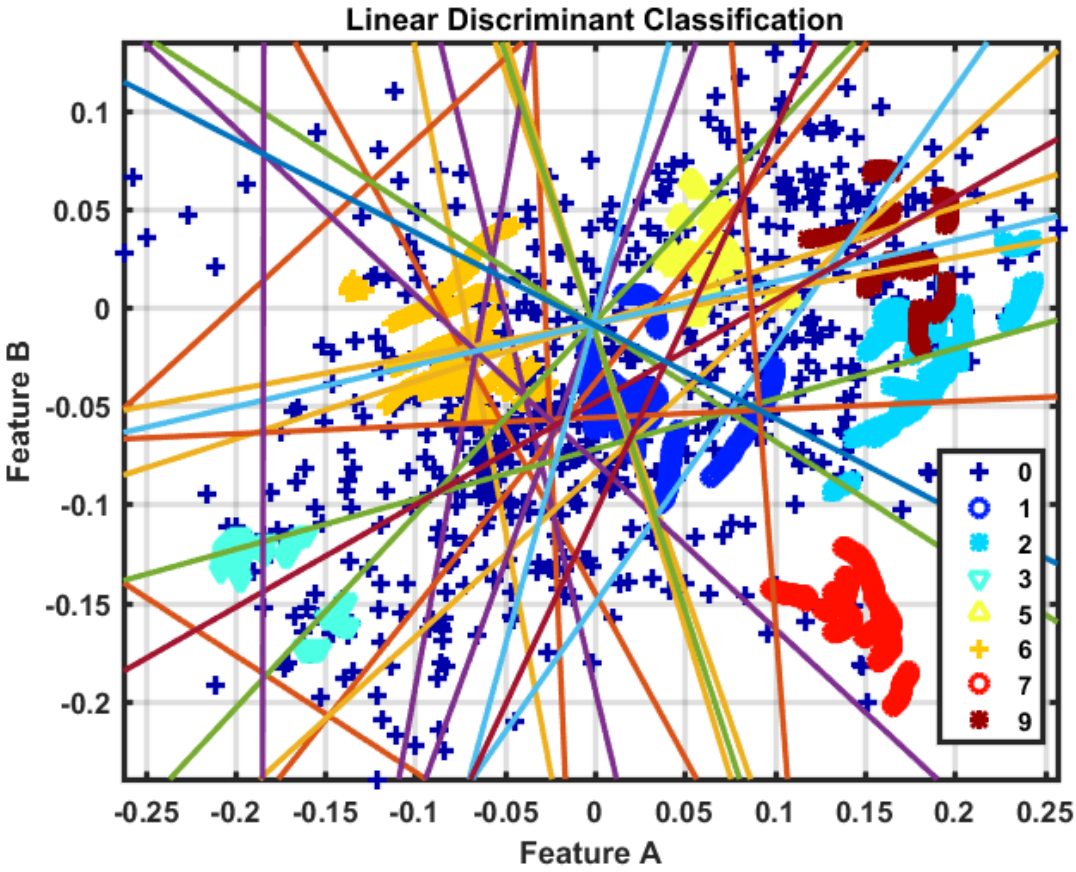

Figure 18: Linear Discriminant Boundaries over Test Dataset in 2-D subspace constructed using one-vs-one scheme.

3.2 Fisher Linear Discriminant Analysis

Fisher's Linear Discriminant Analysis is used a pre-processing step in Classification, which aids in increased Classification efficiency. The advantages of using Fisher's LDA are:

1. Reduces the dimensionality of feature space, there by decreasing the cost of computation.
2. Forms linearly separable, well distinguishable, tight clusters of class data points in a reduced dimensional feature space.
3. Offers flexibility in choosing a subspace dimension
4. Caters to the problem of outlier data points by combining data points belonging to a class in a reduced dimensional feature space.

It could be visualized that the data points become linearly inseparable and overlap when the feature dimension is reduced by a large scale as portrayed in figures 7 and 13. But, Fisher's Projection into a lower dimensional subspace caters to this problem. As it can observed in figures 19,20 and 21, the data points of wine, wallpaper and Taiji data sets have been projected onto a lower dimensional subspace, equal to $(K-1)$, where K is the number of classes in each dataset. It can be observed that there are well separated, discrete clusters of classes formed after the application of Fisher's Projection. These plots convey better information about the data set than the plots without the application of Fisher's LDA.

Data Points for Wine .png

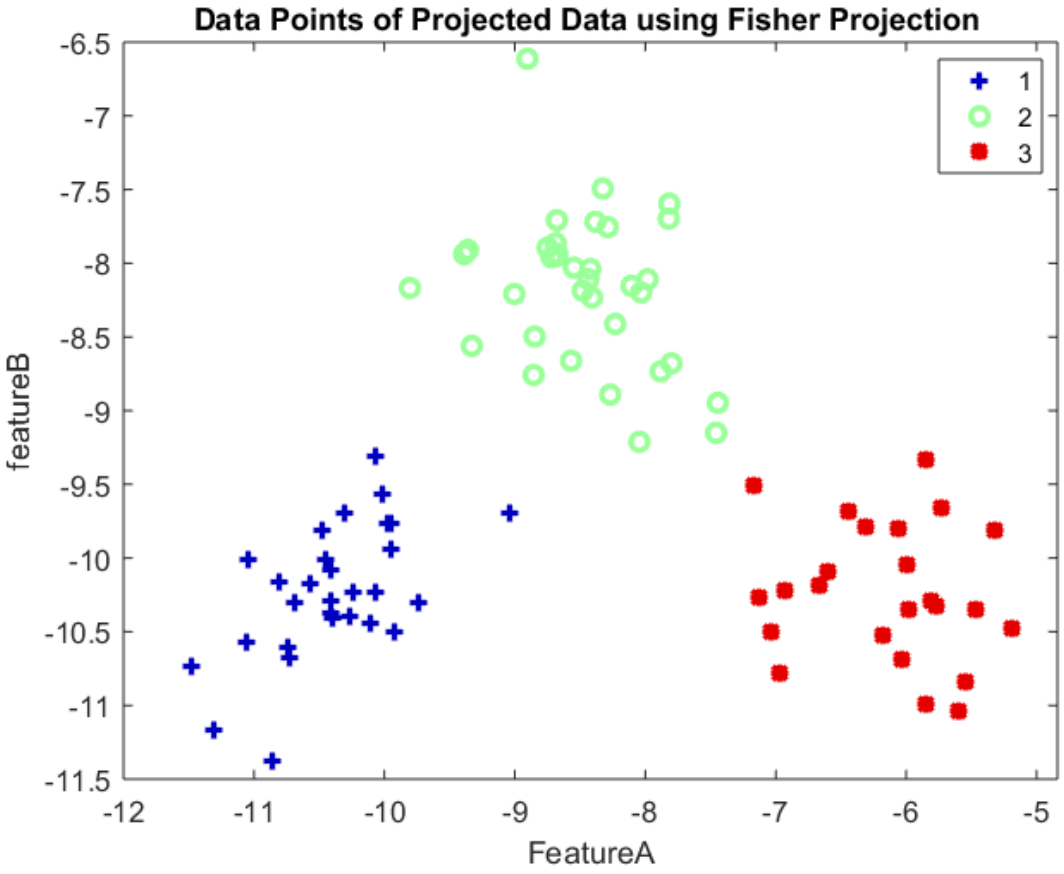

Figure 19: Test Data Points in a 2-D subspace for wine data set using features 1 and 2 after Fisher's Projection.

Data Points for Wallpaper .png

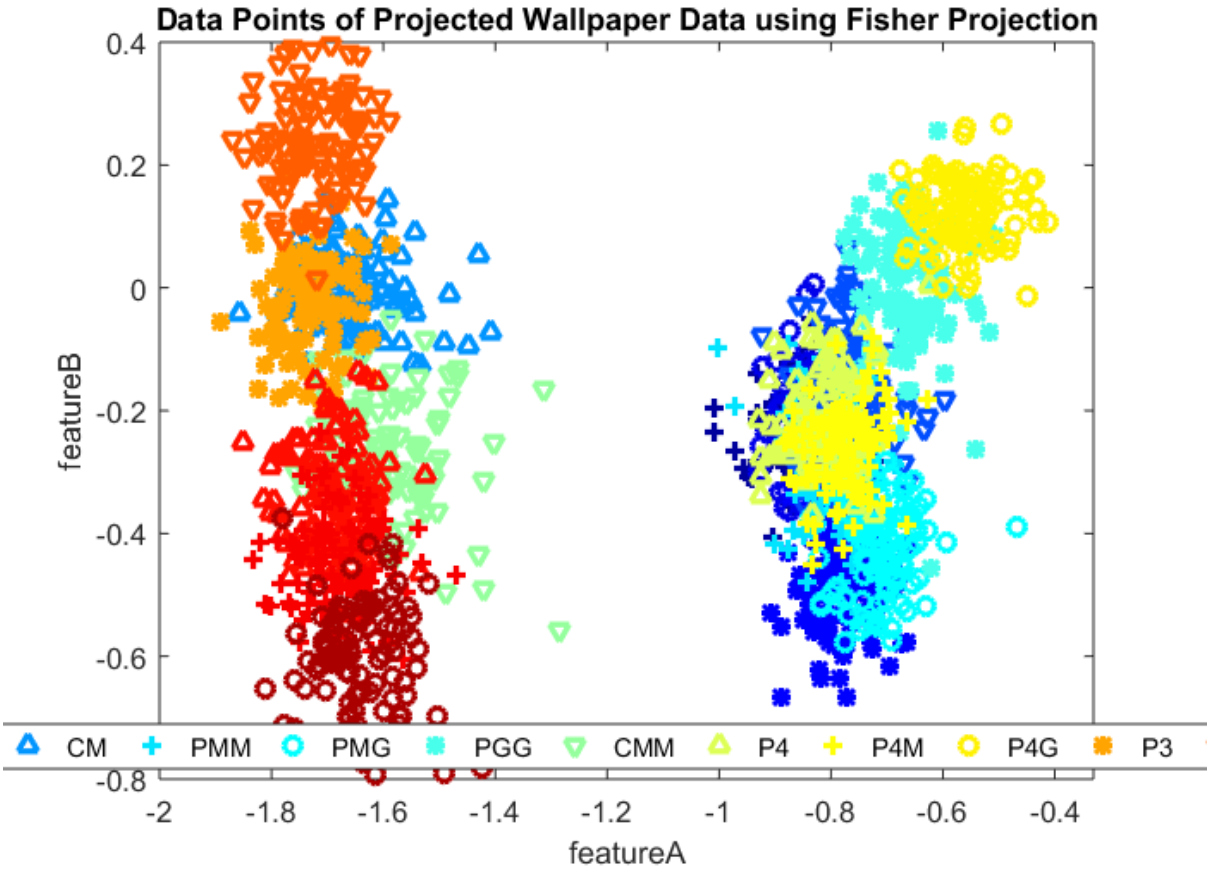

Figure 20: Test Data Points in a 2-D subspace for wallpaper data set using features 1 and 2 after Fisher's Projection.

Data Points for Taiji .png

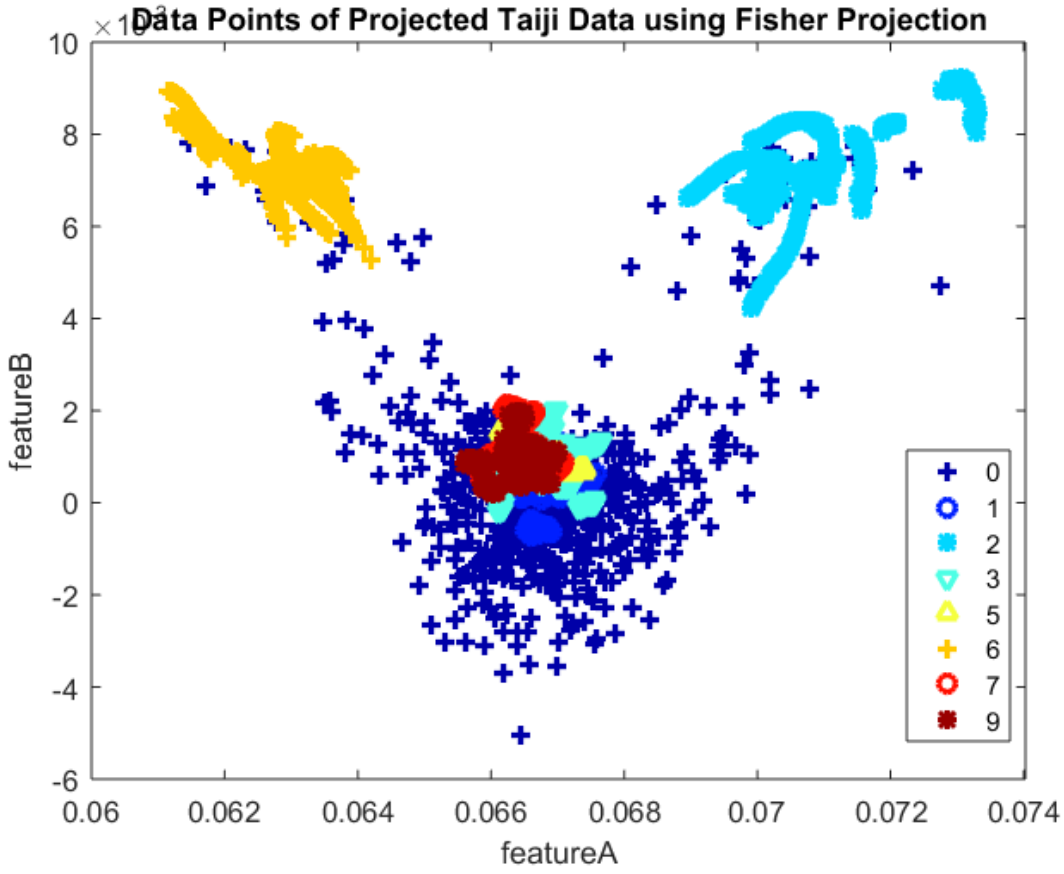

Figure 21: Test Data Points in a 2-D subspace for Taiji data set using features 1 and 2 after Fisher's Projection.

3.3 Classification using Probabilistic Generative Model - Maximum a Priori Estimation

This method gives a probabilistic view of classification which shows how linear decision boundaries arise from simple assumptions about the distribution of the data. The classification is based on the maximum posterior probability, based on a generative approach.

The prior probability is chosen to be the histogram of means of each class. The likelihood function is chosen to be a Multivariate Gaussian. It is assumed that all classes share the same Covariance matrix, which leads to a linear model of discriminants.
After the pre-processing step of Fisher's Linear Discriminant analysis, the generative maximum a priori classification approach is applied on the projected data. In this section, the results corresponding to the three datasets have been shown.

3.3.1 Wine Dataset

Wine data set has 13 features and 3 classes. Fisher's LDA is applied on the dataset to project the data to $(K-1) = 2$ dimensional subspace. Features 1 and 2 from the projected data set is shown in figure 19. Prior Vector, Covariance Matrix and and Mean Vector have been calculated for both train and test datasets to construct the linear discriminant functions for each of the three classes. Figures 22 to 25 show the confusion and class matrices for the classifier built on projected test and training data points.
From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 98.61%.
The testing accuracy for this data set is 97.46%.
Even if it seems as if there is no significant increase in the training and test accuracies, the fact that the classifier is able to attain the same accuracy with only 2 features instead of 13 is remarkable, which is the result of Fisher's LDA.

		Predicted Labels		
Actual Labels	Class 1	30	0	0
	Class 2	0	36	0
	Class 3	0	1	23

Figure 22: Confusion Matrix for Train Data set

		Predicted Labels		
Actual Labels	Class 1	0.965517	0.034483	0
	Class 2	0	1	0
	Class 3	0	0.041667	0.958333

Figure 23: Classification Matrix for Train Data set

		Predicted Labels		
Actual Labels	Class 1	28	1	0
	Class 2	0	35	0
	Class 3	0	1	23

Figure 24: Confusion Matrix for Test Data set

		Predicted Labels		
Actual Labels	Class 1	0.965517	0.034483	0
	Class 2	0	1	0
	Class 3	0	0.041667	0.958333

Figure 25: Classification Matrix for Test Data set

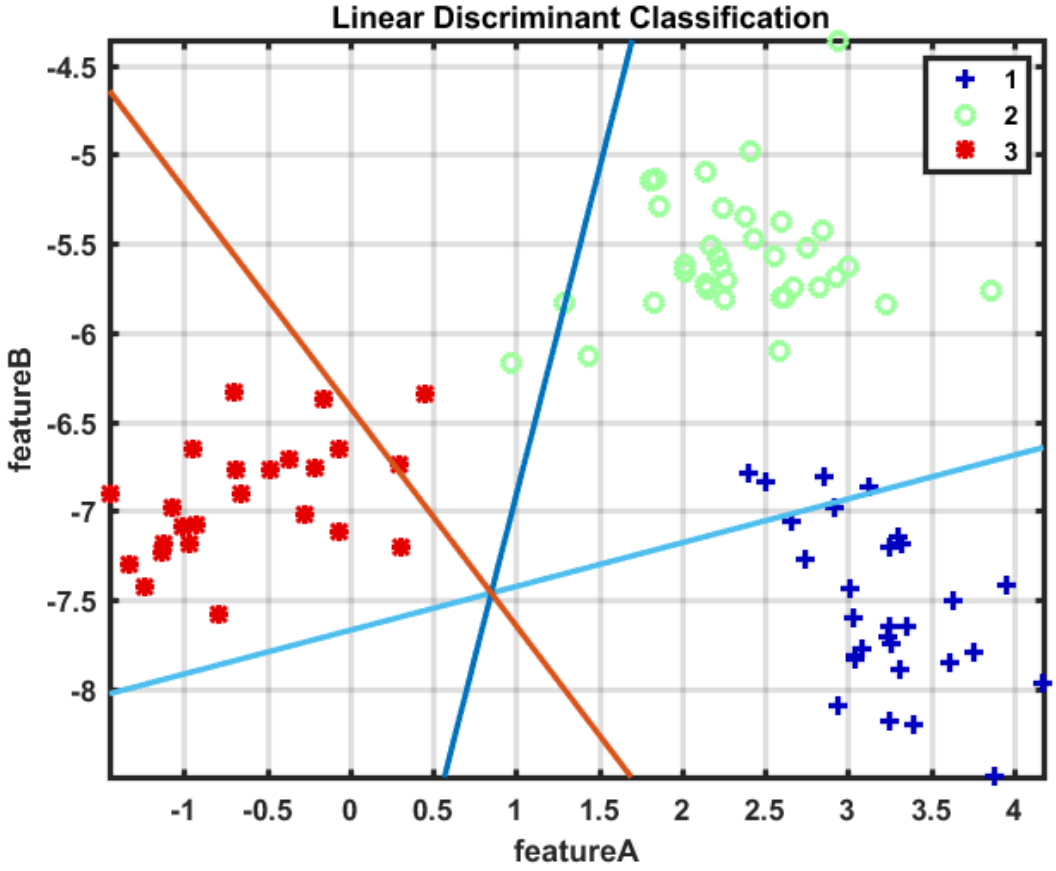

Figure 26: Linear Discriminants constructed on the projected test data in a 2-D subspace

3.3.2 Wallpaper Group Dataset

Wallpaper data set has 500 features and 17 classes. Fisher's LDA is applied on the dataset to project the data to $(K-1) = 16$ dimensional subspace. Features 1 and 2 from the projected data set is shown in figure 20. Prior Vector, Covariance Matrix and and Mean Vector have been calculated for both train and test datasets to construct the linear discriminant functions for each of the 17 classes. Figures 27 to 31 show the confusion and class matrices for the classifier built on projected test and training data points.
From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 98.06%.
The testing accuracy for this data set is 97.59%.
The accuracy of Classification on Test set of Wallpaper data is significantly higher than that of Least-Squares Classification, where all the 500 features were used. This increase in performance of the classifier, even when only 16 features are used, is remarkable and mainly due to the well separated clusters of classes when reduced to a lower dimensional subspace, because of Fisher's LDA. Also, Generative Probabilistic Models are robust than Least Squares Method of classification. The visualization in figure 31 is however not clear, because of high number of classes. Even if it seems as if there is no significant increase in the training accuracy, the fact that the classifier is able to attain the same accuracy with only 16 features instead of 500 is remarkable, which is the result of Fisher's LDA.

		Predicted Labels																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
Actual Labels	P1	98	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	P2	0	97	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
	PM	1	0	96	1	0	0	2	0	0	0	0	0	0	0	0	0	0
	PG	1	1	0	93	0	0	0	5	0	0	0	0	0	0	0	0	0
	CM	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
	PMG	0	0	2	0	0	0	98	0	0	0	0	0	0	0	0	0	0
	PGG	0	0	0	3	0	0	0	94	0	0	0	3	0	0	0	0	0
	CMM	0	0	0	0	1	0	0	0	99	0	0	0	0	0	0	0	0
	P4	1	1	0	0	0	0	0	0	0	97	1	0	0	0	0	0	0
	P4M	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	0	0	0	0	99	0	1	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	2
	P6M	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	99

Figure 27: Confusion Matrix for Train Data set

		Predicted Labels																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
Actual Labels	P1	0.97	0.02	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
	P2	0.02	0.95	0	0	0	0.01	0	0.01	0	0.01	0	0	0	0	0	0	0
	PM	0	0	0.96	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0
	PG	0.02	0	0	0.96	0	0	0	0.01	0	0.01	0	0	0	0	0	0	0
	CM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	0.98	0	0	0	0	0.02	0	0	0	0	0	0
	PMG	0.01	0	0.08	0	0	0	0.91	0	0	0	0	0	0	0	0	0	0
	PGG	0	0.01	0	0	0	0	0.02	0.93	0	0	0	0.04	0	0	0	0	0
	CMM	0	0	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0
	P4	0	0.01	0	0	0	0	0	0.01	0	0.97	0	0.01	0	0	0	0	0
	P4M	0.01	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	P6M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 28: Classification Matrix for Train Data set

		Predicted Labels																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
Actual Labels	P1	97	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	P2	2	95	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
	PM	0	0	96	0	0	0	4	0	0	0	0	0	0	0	0	0	0
	PG	2	0	0	96	0	0	0	1	0	1	0	0	0	0	0	0	0
	CM	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	98	0	0	0	0	2	0	0	0	0	0	0
	PMG	1	0	8	0	0	0	91	0	0	0	0	0	0	0	0	0	0
	PGG	0	1	0	0	0	0	2	93	0	0	0	4	0	0	0	0	0
	CMM	0	0	0	0	3	0	0	0	97	0	0	0	0	0	0	0	0
	P4	0	1	0	0	0	0	0	1	0	97	0	1	0	0	0	0	0
	P4M	1	0	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
	P6M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 29: Confusion Matrix for Test Data set

		Predicted Labels																
		P1	P2	PM	PG	CM	PMM	PMG	PGG	CMM	P4	P4M	P4G	P3	P3M1	P31M	P6	P6M
Actual Labels	P1	0.97	0.02	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0
	P2	0.02	0.95	0	0	0	0.01	0	0.01	0	0.01	0	0	0	0	0	0	0
	PM	0	0	0.96	0	0	0	0.04	0	0	0	0	0	0	0	0	0	0
	PG	0.02	0	0	0.96	0	0	0	0.01	0	0.01	0	0	0	0	0	0	0
	CM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	PMM	0	0	0	0	0	0.98	0	0	0	0	0.02	0	0	0	0	0	0
	PMG	0.01	0	0.08	0	0	0	0.91	0	0	0	0	0	0	0	0	0	0
	PGG	0	0.01	0	0	0	0	0.02	0.93	0	0	0	0.04	0	0	0	0	0
	CMM	0	0	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0
	P4	0	0.01	0	0	0	0	0	0.01	0	0.97	0	0.01	0	0	0	0	0
	P4M	0.01	0	0	0	0	0	0	0	0	0	0.99	0	0	0	0	0	0
	P4G	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	P3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	P3M1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	P31M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	P6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	P6M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 30: Classification Matrix for Test Data set

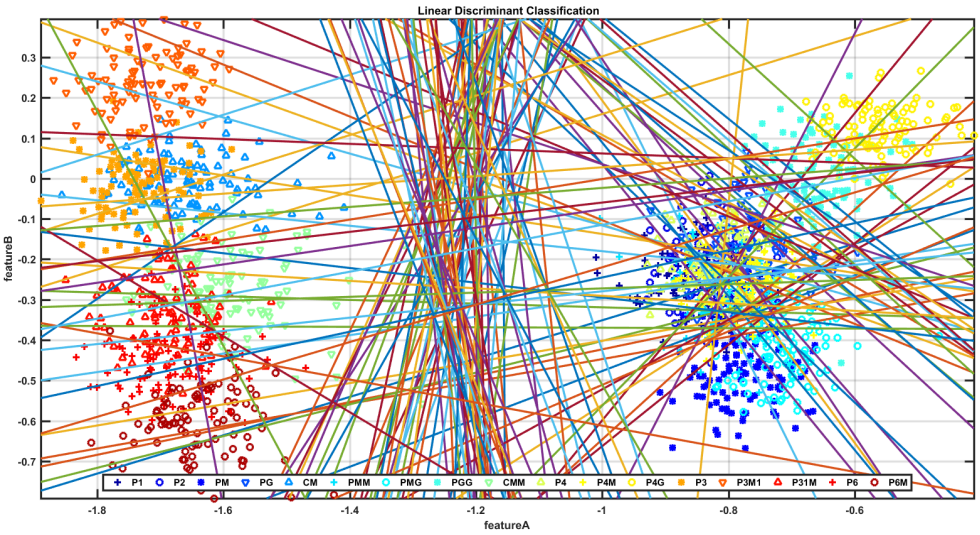

Figure 31: Linear Discriminants constructed on the projected test data in a 2-D subspace

3.3.3 Taiji Pose Dataset

Taiji data set has 64 features and 8 classes. Fisher's LDA is applied on the dataset to project the data to $(K-1) = 7$ dimensional subspace. Features 1 and 2 from the projected data set is shown in figure 21. Prior Vector, Covariance Matrix and and Mean Vector have been calculated for both train and test datasets to construct the linear discriminant functions for each of the 8 classes. Figures 32 to 35 show the confusion and class matrices for the classifier built on projected test and training data points.
From Test and Train Classification Matrices, it can be observed that
The training accuracy for this data set is 90.75%.
The testing accuracy for this data set is 92.31%.
Even if it seems as if there is no significant increase in the training and test accuracies, the fact that the classifier is able to attain the same accuracy with only 7 features instead of 64 is remarkable, which is the result of Fisher's LDA. The visualization in figure 36 is however not clear, because of large number of classes.

		Predicted Labels							
		0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
Actual Labels	**0**	284	111	95	106	187	64	63	181
	1	0	656	0	0	0	0	0	0
	2	0	0	1312	0	0	0	0	0
	3	0	0	0	656	0	0	0	0
	5	0	0	0	0	656	0	0	0
	6	0	0	0	0	0	1312	0	0
	7	0	0	0	0	0	0	656	0
	9	0	0	0	0	0	0	0	656

Figure 32: Confusion Matrix for Train Data set

		Predicted Labels							
		0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
Actual Labels	**0**	0.260312	0.101742	0.087076	0.097159	0.171402	0.058662	0.057745	0.165903
	1	0.384615	0.07365	0.065466	0.062193	0.175123	0.067103	0.070376	0.101473
	2	0	1	0	0	0	0	0	0
	3	0	0	1	0	0	0	0	0
	5	0	0	0	1	0	0	0	0
	6	0	0	0	0	1	0	0	0
	7	0	0	0	0	0	1	0	0
	9	0	0	0	0	0	0	1	0

Figure 33: Classification Matrix for Train Data set

		Predicted Labels							
		0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
Actual Labels	**0**	235	45	40	38	107	41	43	62
	1	0	369	0	0	0	0	0	0
	2	0	0	738	0	0	0	0	0
	3	0	0	0	369	0	0	0	0
	5	0	0	0	0	369	0	0	0
	6	0	0	0	0	0	738	0	0
	7	0	0	0	0	0	0	369	0
	9	0	0	0	0	0	0	0	369

Figure 34: Confusion Matrix for Test Data set

		Predicted Labels							
		0	**1**	**2**	**3**	**5**	**6**	**7**	**9**
Actual Labels	**0**	0.260312	0.101742	0.087076	0.097159	0.171402	0.058662	0.057745	0.165903
	1	0.384615	0.07365	0.065466	0.062193	0.175123	0.067103	0.070376	0.101473
	2	0	1	0	0	0	0	0	0
	3	0	0	1	0	0	0	0	0
	5	0	0	0	1	0	0	0	0
	6	0	0	0	0	1	0	0	0
	7	0	0	0	0	0	1	0	0
	9	0	0	0	0	0	0	1	0

Figure 35: Classification Matrix for Test Data set

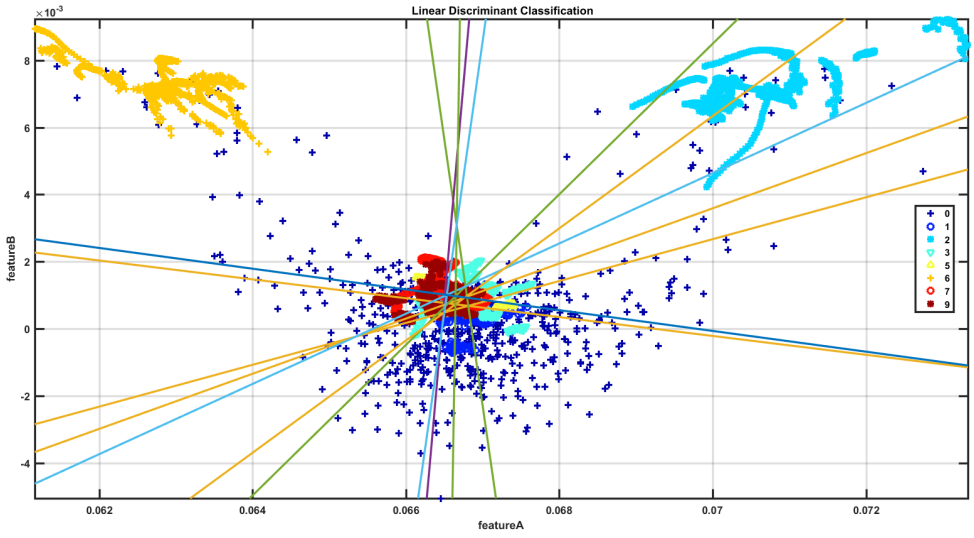

Figure 36: Linear Discriminants constructed on the projected test data in a 2-D subspace

4 Conclusions

These are some of the conclusions that can be drawn from the project:

1. Least-Squares, although generally used for regression problems, performs well on Classification problems as well.

2. Least-Squares method of Classification is preferred as it gives a Closed form Solution for Parameter Matrix W and a Linear discriminant Model.

3. Least Squares fails when there are outlier points in the data set, which will result in reduction in performance efficiency.

4. The elements of weight matrix of least squares cannot be considered to be probabilities since the weights are constrained to be in the range (0,1).

5. Method of least squares performs well on Wine and Taiji datasets, with both train and test accuracies above 90%. But, the test accuracy of Wallpaper data set is only 62%.

6. Fisher's Linear Discriminant Analysis is used a pre-processing step to form well-separated, distinguishable, tight clusters of class data points.

7. Fisher's LDA is very flexible in terms of choosing a subspace dimension.

8. Generative Probabilistic Models are robust than Least-Squares. They can cater to the problem of outlier points.

9. Generative Probabilistic Model applied on Fisher's projected data points significantly increases the classification efficiency on both training and test data, even with reduced dimension of feature space.

   Hence, it can be concluded that the classification accuracies for test and train datasets significantly increase by applying Generative Probabilistic Model on Fisher's Projected data points.

5 Extra Credits

For a two class problem, Fisher's Criterion can be obtained a s a special case of least squares. The goal of least squares was to determine the linear discriminant based on the goal of making the model predictions as close as possible to a set of target values. By contrast, Fisher's criterion was derived by requiring maximum class separation in the feature space.

Instead of 1 of K coding, consider the changed scheme of assigning target values.[1]

$$\begin{aligned} tn &= \frac{N}{N_1} \quad if \quad x_n \in C_1 \\ tn &= -\frac{N}{N_2} \quad if \quad x_n \in C_2 \end{aligned} \tag{27}$$

The sum of squares error function can be written as

$$E = \frac{1}{2}\sum_{n=1}^{N}(w^T x_n + w_0 - t_n)^2 \tag{28}$$

Setting the derivatives of E with respect to w_0 and w to zero,

$$\begin{aligned} \sum_{n=1}^{N}(w^T x_n + w_0 - t_n) = 0 \\ \sum_{n=1}^{N}(w^T x_n + w_0 - t_n)x_n = 0 \end{aligned} \tag{29}$$

From the first equation in 29, we get

$$\begin{aligned} \sum_{n\in C_1}(w^T x_n + w_0 - fracNN_1) + \sum_{n\in C_2}(w^T x_n + w_0 - fracNN_2) = 0 \\ But, \quad \sum_{n=1}^{N} t_n = N_1\frac{N}{N_1} - n_2\frac{N}{N_2} = 0 \\ \Rightarrow Nw_0 = -w^T\sum_{n=1}^{N} x_n \\ \Rightarrow w_0 = -w^T m \end{aligned} \tag{30}$$

where, m is the mean of the total data set.
We know that the within class scatter matrix

$$\begin{aligned} S_W = \sum_{n\in C_1}(x_n - m_1)(x_n - m_1)^T + \sum_{n\in C_2}(x_n - m_2)(x_n - m_2)^T \\ and \quad S_B = (m\_m_1)(m_2 - m_1)^T \\ where, \quad m_i = \frac{1}{N_i}\sum_{n\in C_i} x_n \end{aligned} \tag{31}$$

31, second equation in 29 and equation 27, we have

$$\begin{aligned} S_W w &= a(m_2 - m_1) - N(m_2 - m_1) \\ \Rightarrow \left(S_W + \frac{N_1 N_2}{N} S_B\right) w &= N(m_1 - m_2) \end{aligned} \tag{32}$$

Since $S_B w$ is always in the direction of $(m_2 - m1)$, we can write,

$$w \propto S_W^{-1}(m_2 - m_1) \tag{33}$$

Equation 33 denotes the Fisher Criterion, derived from Least squares for two classes. Thus, the weight vector of least squares coincides with that found from the Fisher Criterion. From the expression of w_0 from equation 30, it is evident that the decision boundary for unseen data x is given by
$y(x) = w^T(x - m)$
Figure 37 denotes the plot of wine data points for two classes. Figure 38 denotes the plot for projected wine data points for the same two classes. It can be observed that the data points in these figures are mirror-images of each other. This is because, when subspace dimension is not lesser than the original dimension, Fisher Projection behaves like a Rotation Matrix. It was observed that when Least-squares method were applied to both these data sets, they yielded the same weight matrix, which proves empirically that Fisher Criterion is a special case of Least Squares for two classes. The plots of decision boundaries for both the data points have been shown in figures 39 and fig:40. The decision boundaries can be visualized to be exactly equivalent.

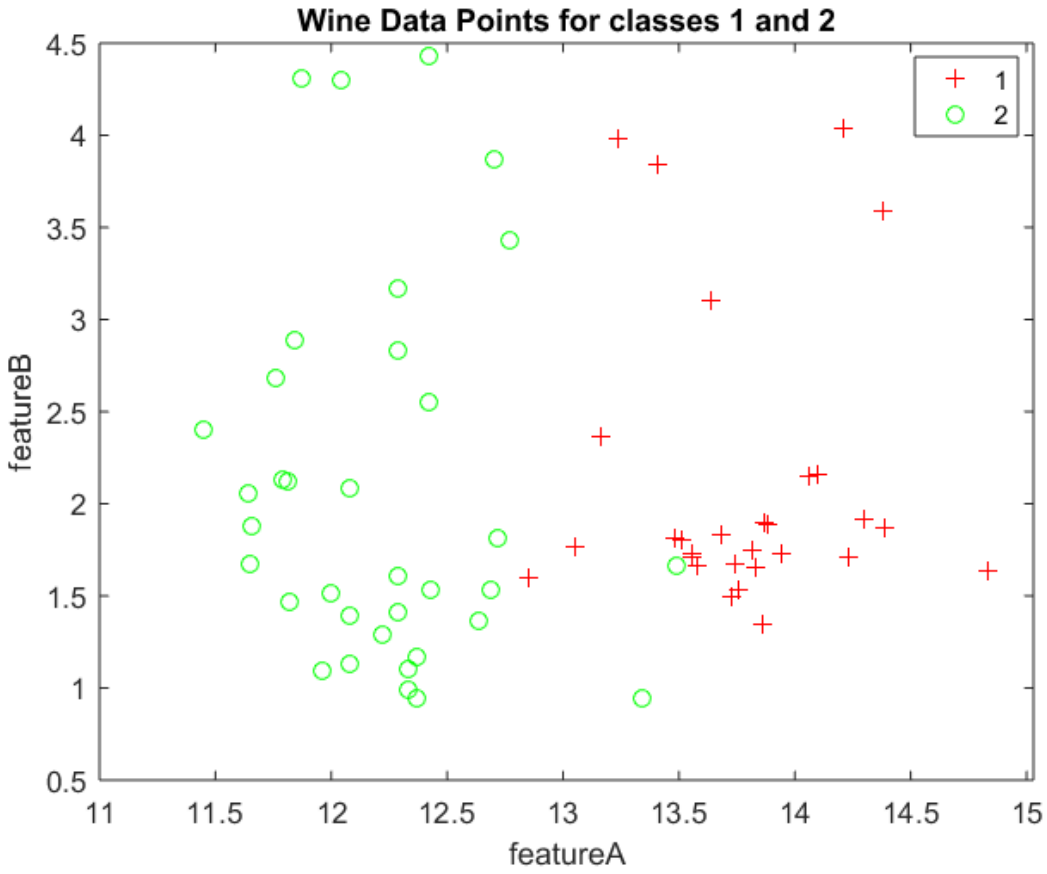

Figure 37: Plot of Test Wine Data Points of class 1 and 2, with two features

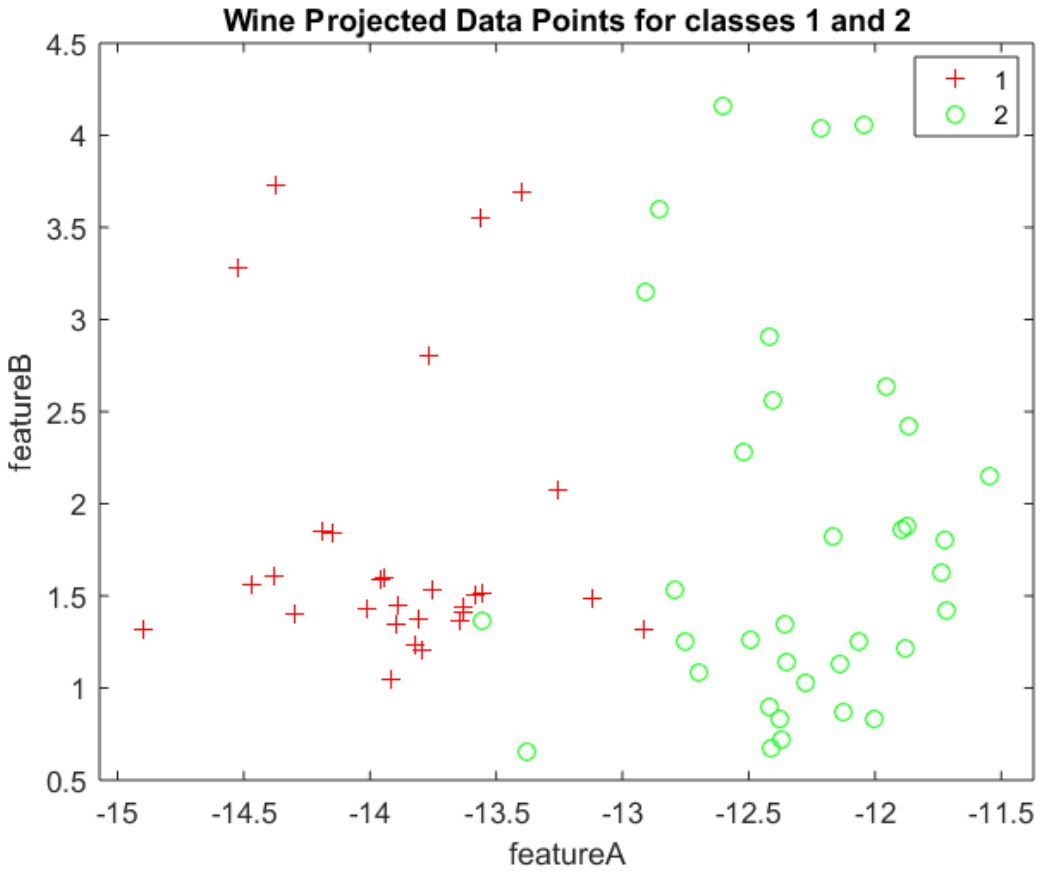

Figure 38: Plot of Projected Test Wine Data Points of class 1 and 2, with two features

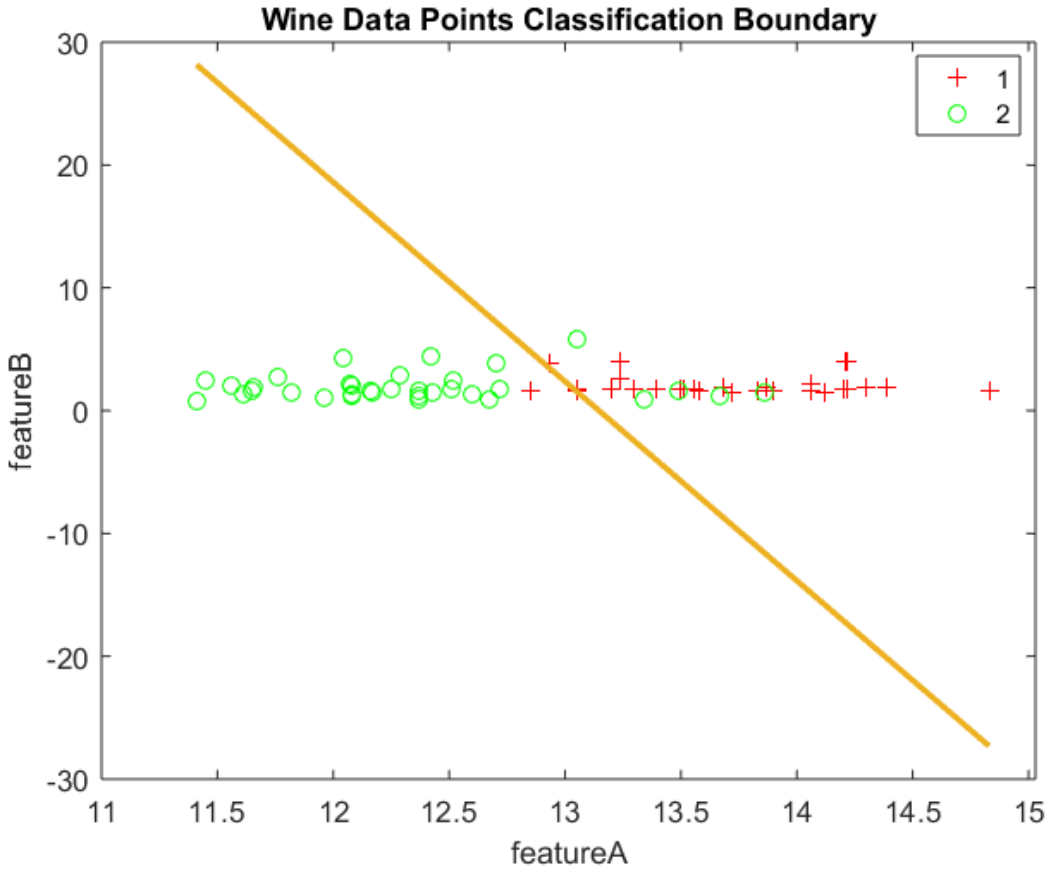

Figure 39: Linear Discriminants constructed on the wine data points with classes 1 and 2

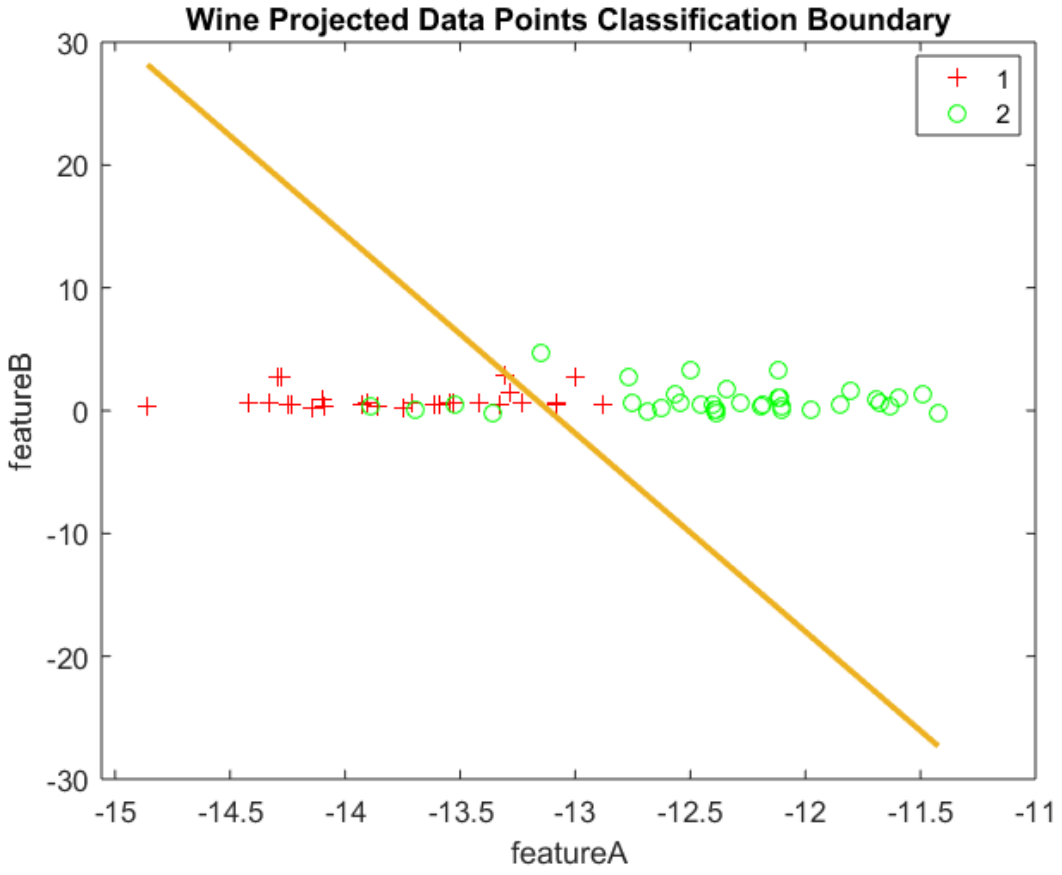

Figure 40: Linear Discriminants constructed on the projected wine data points with classes 1 and 2

References

[1] Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006 3, 6, 8, 41

[2] Stat 202 *Data Mining and Analysis*. October, 2017 11

[3] Prince, Simon JD. *Computer Vision: Models, Learning and Inference*. 2012 3