Pattern Recognition and Machine Learning Supervised Classifiers

Pavan Gurudath

February 14, 2018

Abstract

Pattern Classification can be broadly classified into two types, namely Supervised Learning and Unsupervised Learning. Supervised Learning, which is analysed in this project, is one form of Machine Learning where a function is inferred from a labelled data set. This type of learning is mostly used in classification problems where teh goal is to input a data and classify it to one of the K discrete classes C_k where $k = 1, 2, \ldots, K$. In most applications, every input is assigned to just one of these K classes. Therefore, decision surfaces or boundaries are formed, which are the divided regions in the input space that decides the particular class of the input data.

In this project, two such linear models for classification has been analysed and compared for three different data-sets. These data-sets are *wine, wallpaper* and *Taiji pose*.

The two models for classification that have been used in this project are ***Least Squares*** and ***Fisher's Projection***. The two algorithms have been comprehensively compared with respect to their performance efficiency and accuracy on the test data. .

Contents

1 Introduction

The concept of classification refers to the identification of the category of a new observation. Based on a certain set of data that would consist of data that could be separated into K classes, we find a way to classify to which amongst these K classes, the new input data would belong to. Classification is a typical problem for Machine Learning problems. One such example is the case of determining whether a patient has cancer that is benign or malignant.

In order to classify the input data, the decision boundaries between each of these K classes $i.e. C_k \forall k = 1, \ldots, K$ has to be learnt. We consider three datasets, namely *wine, wallpaper* and *Taiji pose*, which are linearly separable *i.e.* a linear decision surface also known as a hyper plane could be used to separate the different classes.

In this project, we use two methods to linearly separate the data. The algorithms are

- **Least squares classification** We apply this algorithm on the original dataset and thereby classify.
- **kNN: k-nearest neighbours** In this method, we reduce the dimensionality of the original dataset to a smaller dimension using the Fisher's projection and use kNN to classify the data.

2 Approach

The section 2 details the approach that has been taken in order to achieve our objective of classifying the given datasets into their respective classes. Section 2.1 introduces the datasets that have been used in this project while section 2.2.1 and 2.2.2 details the derivation of their respective algorithms.

2.1 Data

The data-sets that have been loaded are separated into test and train category. Each of these test and training data-set consists of feature vectors and its corresponding labels. Every data-set that has been used in this project has the following set of common parameters once they are loaded into the program. These parameters are namely: a) Number of Classes (K)
b) Size of features/dimensions (D)
c) Number of samples/data points (N)

The features and the parameters of each of the datasets have been explained in detail in Section 2.1.1, 2.1.2 and 2.1.3.

2.1.1 Wine

The wine dataset is used in order to determine the origin of wines using the chemical analysis. The data contains 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 thirteen constituents found in each of the three types of wines. The attributes that are included 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 Features: 13
Training parameters:
Number of training observations: 90
Number of samples per class in train data-set:
1 30
2 36
3 24
Testing parameters:
Number of testing observations: 88
Number of samples per class in test data-set:
1 29
2 35
3 24

2.1.2 Wallpaper

The data in this set consists of the features extracted from images containing the 17 Wallpaper Groups.

Number of Classes: 17
Number of Features: 500
Features of the dataset:

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

Training parameters:
Number of training observations: 1700
Number of samples per class in train data-set: 100
Testing parameters:
Number of testing observations: 1700
Number of samples per class in test data-set: 100

2.1.3 Taiji pose

The data in this set is a collection of the joint angles (in quaternions) of 35 sequences from 4 people performing Taiji at the Penn State motion capture lab. The '0' label corresponds to non-translational frames and the non '0' labels correspond to translational frames.

Number of Classes: 8
Number of Features: 64
Training parameters:
Nnumber of training observations: 11361
number of samples per class in train dataset:

0	1767	**1**	1066
2	2132	**3**	1066
5	1066	**6**	2132
7	1066	**9**	1066

Testing parameters:
Number of test observations: 3924
Number of samples per class in test dataset:

0	603	**1**	369
2	738	**3**	369
5	369	**6**	738
7	369	**9**	369

2.2 Methods

2.2.1 Least-Squares Classification

It has been studied that the minimization of a sum-of-squares error function would lead to a closed form solution for parameter values. Although least squares method is usually used for regression problems, it works 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 matrix, where its columns denote the classes. 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 its own linear discriminant model such that

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

where w_{k0} is the bias, which indicates the y-intercept of the linear discriminant.

Grouping the vectors into a Matrix form, we get

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

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

$$\text{and} \quad \tilde{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1D} \\ 1 & x_{21} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{bmatrix}$$

A new input x is assigned to the class for which $y_k = \tilde{x}\tilde{W}$ is the largest. The 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 $\frac{\partial E(\tilde{W})}{\partial \tilde{W}} = 0$

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

In order to simplify equation (4), we make use of the following axioms:

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

Therefore,

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

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

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

Equation (7) gives the optimal closed form solution, which is used to predict the class labels for the unseen data.
A single K class discriminant comprising of K linear functions of the form of equation (1) aids in avoiding unambiguous regions. A new point x is assigned to class C_k if $y_k(x) > y_j(x)\ \forall \text{j} \neq k$.

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.
Although the least squares method gives an exact closed-form solution for the discriminant function parameters, this approach lacks robustness for outlier data points. The additional points that are far away from the dataset reduces the efficiency of classification when such points are present in the data. The sum-of-squares error function penalizes predictions that are "too correct".

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 $(0, 1)$. However they do sum up to one. Therefore, adopting more appropriate models would result in better classification.

2.2.2 Fisher's Linear Discriminant Analysis

This section details the Fisher's Discriminant analysis in terms of multi-class classification. Mathematicians have always preferred linear models due to its ability to generalize over larger terms as well as the convenience of computation. Therefore, one of the ways of looking at a linear classification model could be in terms of feature space dimensionality reduction. While looking at this perspective, we know that projecting the data onto a lower dimension could lead to loss of information and classes that are well separated in the original D-dimensional space could probably overlap in a lower dimensional subspace, thereby its inability to linearly separate different classes. However, reducing the

dimension of data using a method which would not lead to losses, could result in ease of visualization, efficiency in terms of computing as well as classifying the model *i.e.* linear separability. This concept is the idea behind Fisher's Linear Discriminant Analysis.

Fisher's model is built based on the idea of finding a projection where the projected data has a maximum separation between inter-class means and simultaneously has a minimum intra-class variance.

Let us 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 number of points respectively. Let X be a D-dimensional feature matrix. 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. That 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 C_i and C_j are therefore given by equation 10

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

Here, x_n is D-dimensional vector that corresponds to a single row of the feature matrix. In general, $m_k \in \mathbb{R}^{D \times 1}$, where $k = 1, \ldots, K$
Let $D^{'}$ be the fisher dimension *i.e.* it is the dimension onto which the original feature matrix is to be projected. Hence, there would be $D^{'}$ linear features of the form $y_k = w_k^T x$, where $k = 1, \ldots, D^{'}$. The vectors w_k can be considered to be the columns of a matrix W such that

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

Here, y is the projected feature matrix in the $D^{'}$ dimensional feature space.

We 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 (covariance) 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 equation 12, we know that $S_W = \sum_{k=1}^{K} S_k$ Using this result, the total intra class variance is given by

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

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

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

where S_B is the inter-class Scatter Matrix. Hence, from equation 9, the Fisher function can be written 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 equation 16 and equate it to zero.

$$\begin{aligned}\frac{d}{dw}[J(w)] &= 0 \\ \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right] &\Rightarrow \frac{(w^T S_W w)(2S_B w) - (w^T S_B w)(2S_W w)}{(w^T S_W w)^2} = 0 \\ &\Rightarrow (w^T S_W w S_B w) = (w^T S_B w S_W w) \\ &\Rightarrow S_B w = \lambda S_W w, \\ &\Rightarrow w = S_W^{-1}(m_i - m_j)\end{aligned} \tag{17}$$

$(since \quad w^T S_W w \quad and \quad w^T S_B w \quad are \quad scalars)$.
Equation 17 uses the following 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 follows:

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

The following steps are used in order to find the projection matrix W:

1. Using equations 12 and 18, the matrices S_W and S_B are to be calculated respectively.
2. The eigen values and eigen vectors of the matrix $S_W^{-1} S_B$ are to be calculated.
3. Next, the eigen values are sorted in descending order along with their corresponding eigen vectors.
4. A subspace dimension D' is selected onto which the feature matrix is projected. In our case, it is taken to be one less than the number of classes of the data-set.

5. The first $D^{'}$ eigen vectors are selected which constitutes the columns of the projection matrix W.

With Fisher's linear Discriminant analysis, the following observations are made. 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. There are at most (K-1) non-zero eigen values. Therefore, it can be implied that the projection onto a (K-1) dimensional subspace spanned by the eigen vectors of the inter-class scatter matrix S_B does not alter the value of $J(w)$. Hence, a maximum of (K-1) linear features could be found using this approach.

2.2.3 Using a kNN classifier

After the pre-processing step using Fisher's Linear Discriminant Analysis, the projected data is used for classification. We use the method of classifying new input data using the k-Nearest Neighbourhood approach. In this method of kNN, it follows a type of unsupervised learning, where there technically is no "training" that can be done. Using the set of training data points, we calculate the distance of the new input point with respect to every trained data point that has already been clustered. Using this, we obtain a vector of distance with respect to all the points. These distances are sorted in an ascending order to obtain the k nearest neighbours of the data point. The parameter k is to be tinkered with in order to get maximum efficiency. Using the k-nearest neighbours, the mode of their corresponding classes are taken in order to obtain the predicted label of the new data point.

Let us suppose that we have a data set comprising of N_k points in class C_k with N points in total, such that $\sum_k N_k = N$. In order to classify a new point x, a sphere centred on x containing precisely K points irrespective of their class is drawn. If this sphere consists of K_k points from class C_k, then we obtain

$$p(x|C_k) = \frac{K_k}{N_k V} \tag{19}$$

where V is the volume of the resulting sphere.

The unconditional density and class priors are given by

$$p(x) = \frac{K}{NV} \tag{20}$$

$$p(C_k) = \frac{N_k}{N} \tag{21}$$

Using equation 19, 20 and 21, from Bayes' theorem we get the posterior probability of class membership

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} \tag{22}$$

To minimize the probability of misclassification, we assign the test point x to the class having the largest posterior probability, corrosponding to the largest value of K_k/K.

3 Results

Section 3 details the results that are obtained after performing the aforementioned algorithms/models on the three data set. Here, the data regarding the data points, classification and confusion matrices on the training and testing data as well as their respective accuracies have been reported. Visualisations of the classification decision boundaries in the two dimensional feature space has been portrayed for the case of least square classification.

In the MATLAB execution, the main function executes all the datasets one after the other and their values are saved in their respective structures. The struct of $wine, wallpaper$ and $taiji$ includes the feature indices that have been used, optimized weight matrix $W*$, number of classes in the dataset along with the feature vector X, target vector T, output vector y, target labels, predicted labels, classification matrix, confusion matrix, accuracy and standard deviation of training and testing data.
The struct $wine2, wallpaper2$ and $taiji2$ contains the same data for only two features that have been used to visualize the boundaries in the case of least squares classification.
The struct $knn\_wine, knn\_wallpaper$ and knn_taiji contains the classification matrix, confusion matrix, accuracy and standard deviation of the testing data.

In this way it is easy to observe all the key features without the hassle of using breakpoints and finding the data.

3.1 Least-Squares Classification

The least squares classification method provides us with a closed form solution for the discriminant function parameters. The error function is a simple quadratic convex function and leads to a simple optimization technique. However, the classifier is sensitive to outlier points and penalizes a point that is in a sense 'too correct'. These are some of the positives and negatives of the classifier. The following sections reports the classification and confusion matrices along with the accuracy of the dataset that has been considered. The confusion matrix is a table that allows visualization of the performance of an algorithm, *i.e.* it consist of rows that contains the details of the ground truth label while the columns consists of the predicted label. Therefore each cell corresponds to the number of predicted labels for that particular ground truth or vice versa. The classification matrix is the ratio of the number of labels in the predicted class to the ground truth class and is such that the row sums upto 1.

3.1.1 Wine Dataset

As explained in section 2.1.1, this dataset contains 3 classes each of which contains 13 features/dimensions. For the sake of visualization, only two of the features have been considered, so that the data points can be plotted on a 2-D space. The data points are as shown in Figure 1.

It can be observed that the data points belonging to different classes in Figure 1 are not completely scattered and there are a few points that do overlap. After applying least squares classification, the linear discriminants constructed using one vs one scheme over the test data points are as shown in Fig. 2.

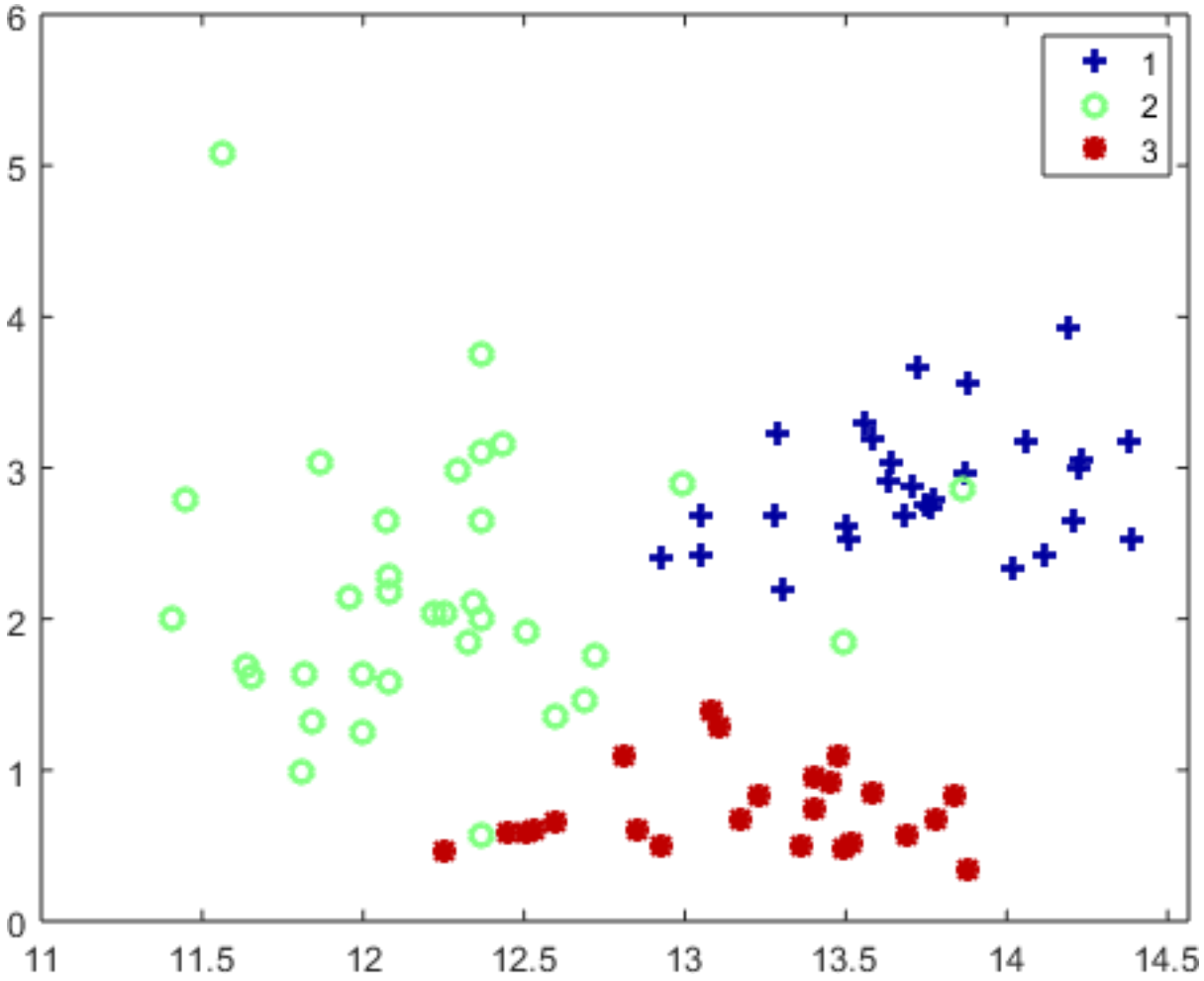

Figure 1: Test Data Points in 2-D subspace for wine data-set using features 1 and 7

Figure 3 and Figure 4 depicts the classification matrix and confusion matrix for the training data respectively. Figure 5 and Figure 6 depicts the same for the test data.

Using the confusion matrix, the accuracy for training and testing is calculated and is as follows for the entire dataset:

Training accuracy - 100%

Testing accuracy - 98.0592%

Thus it can be seen that the least square classifier performs well on wine dataset.

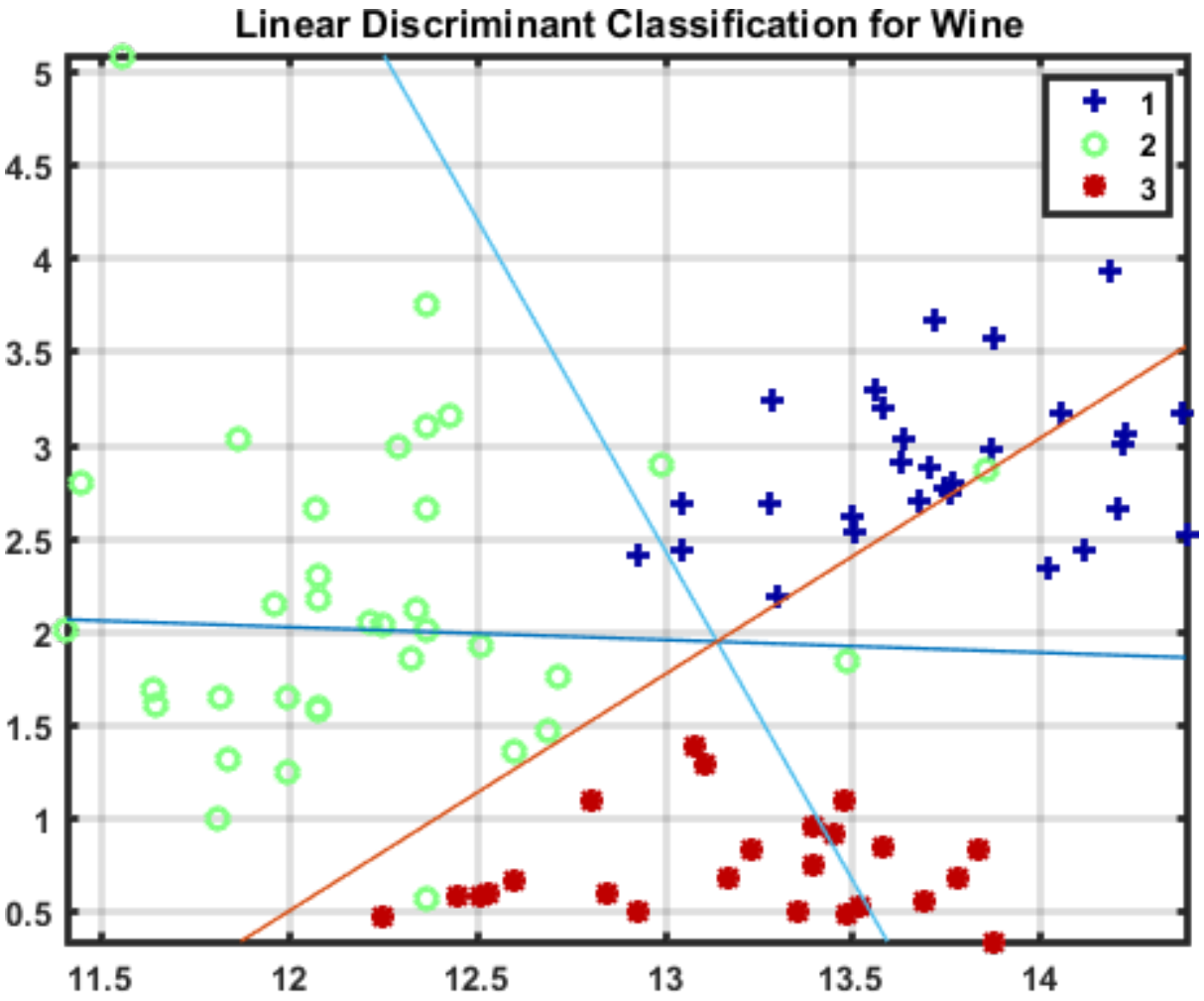

Figure 2: Linear Discriminant Boundaries over Test dataset plotted using one-vs-one scheme.

		PREDICTED LABELS		
		CLASS - 1	CLASS - 2	CLASS - 3
GROUND TRUTH	CLASS - 1	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
GROUND TRUTH	CLASS - 1	30	0	0
	CLASS - 2	0	36	0
	CLASS - 3	0	0	24

Figure 4: Confusion Matrix for Train Data set

		PREDICTED LABELS		
		CLASS - 1	CLASS - 2	CLASS - 3
GROUND TRUTH	CLASS - 1	1	0	0
	CLASS - 2	0.028571	0.942857	0.028571
	CLASS - 3	0	0	1

Figure 5: Classification Matrix for Test Data set

		PREDICTED LABELS		
GROUND TRUTH		CLASS - 1	CLASS - 2	CLASS - 3
	CLASS - 1	29	0	0
	CLASS - 2	1	33	1
	CLASS - 3	0	0	24

Figure 6: Confusion Matrix for Test Data set

3.1.2 Wallpaper Groups Dataset

As explained in section 2.1.2, this dataset contains 17 classes each of which contains 500 features/dimensions. For the sake of visualization, only two of the features have been considered, so that the data points can be plotted on a 2-D space. The data points, as shown in Figure 7, overlap over one another and aren't linearly separable in the 2 dimensional subspace. Therefore the reduction of dimension from 500 to 2 results in loss of information as well as linear separability.

After applying least squares classification, the linear discriminants constructed using one vs one scheme over the test data points are as shown in Fig. 8. The visualization is poor and does not convey any useful information. For computational convenience, the classes in the wallpaper dataset have been converted to an array of numbers from 1 to 17 as shown in section 2.1.2.

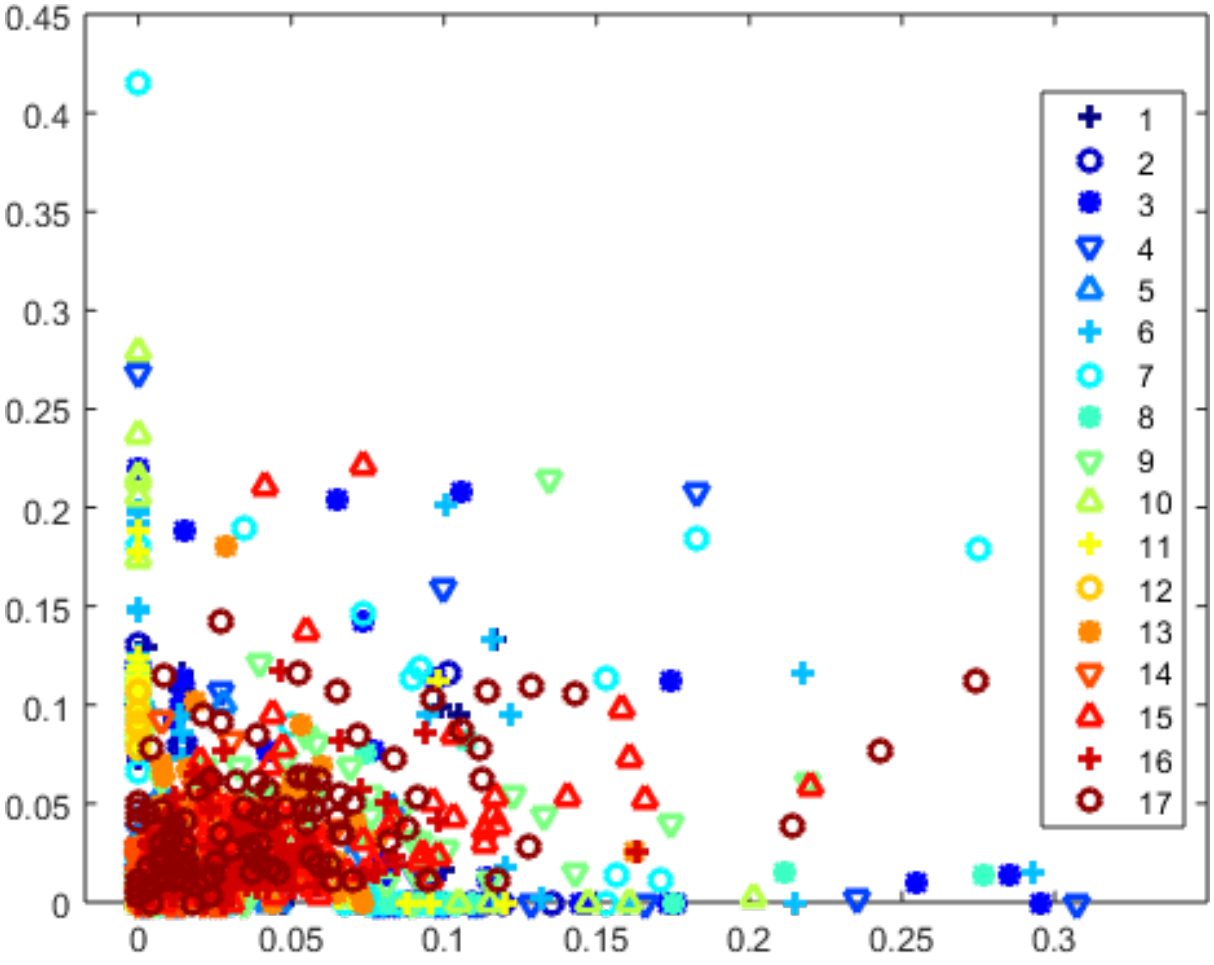

Figure 7: Wallpaper test data points in 2-D subspace using features 1 and 7

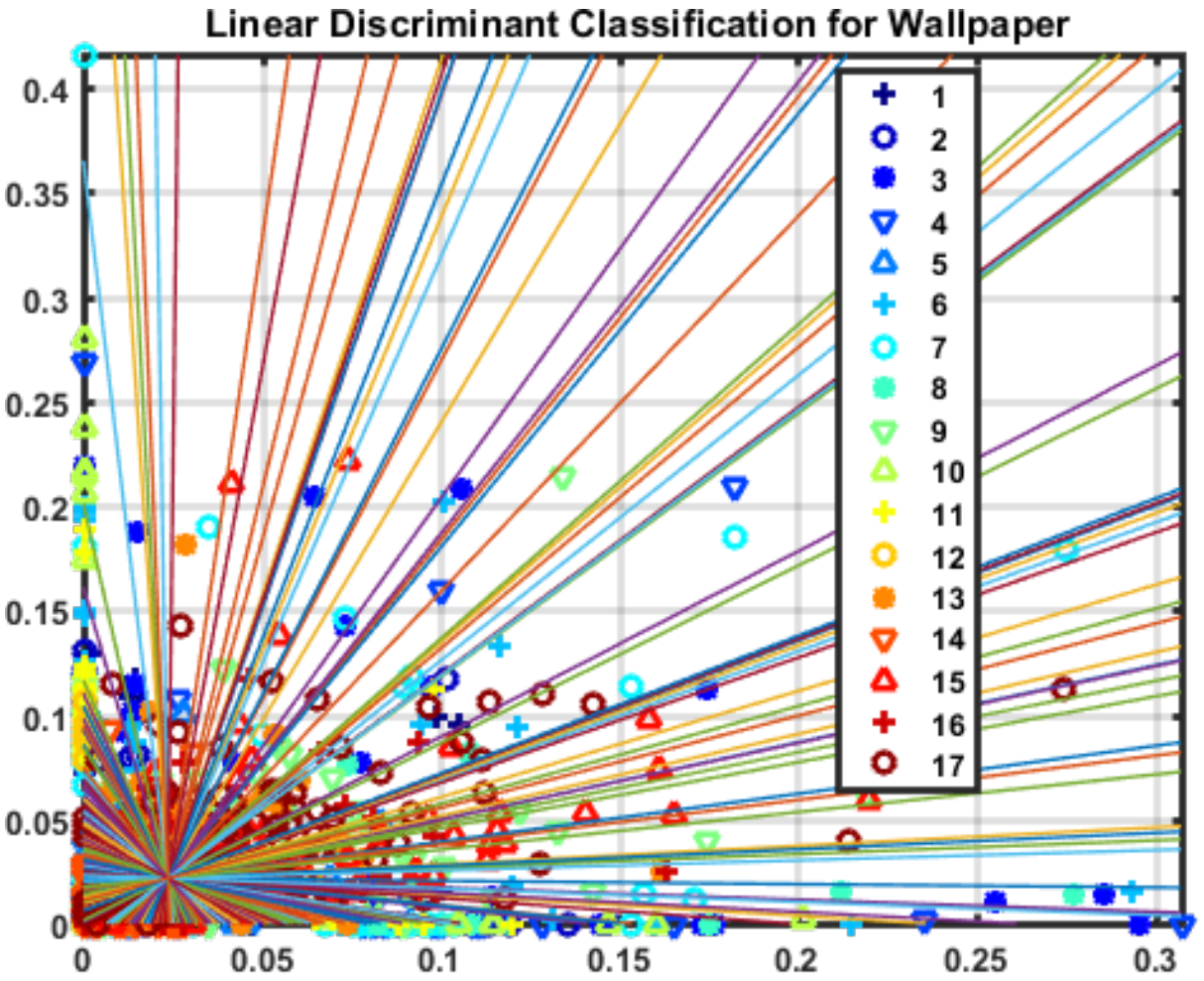

Figure 8: Linear Discriminant Boundaries over Test dataset plotted using one-vs-one scheme

Figure 9 and Figure 10 depicts the classification matrix and confusion matrix for the training data respectively. Figure 11 and Figure 12 depicts the same for the test data.

Using the confusion matrix, the accuracy for training and testing is calculated and is as follows for the entire dataset:

Training accuracy - 96.7059%

Testing accuracy - 61.0588%

Thus it can be seen that the least square classifier does not perform well on wallpaper group dataset.

		PREDICTED LABELS																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	0.96	0.01	0	0.02	0	0	0	0	0	0.01	0	0	0	0	0	0	0
	CLASS - 2	0.02	0.9	0.01	0.05	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0
	CLASS - 3	0.01	0	0.97	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0
	CLASS - 4	0.02	0.02	0	0.91	0	0	0.01	0.03	0	0.01	0	0	0	0	0	0	0
	CLASS - 5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	CLASS - 6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	CLASS - 7	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0
	CLASS - 8	0	0	0.01	0.02	0	0	0	0.91	0	0	0	0.06	0	0	0	0	0
	CLASS - 9	0	0	0	0	0.03	0	0	0	0.97	0	0	0	0	0	0	0	0
	CLASS - 10	0.01	0.01	0	0	0	0.01	0	0	0.01	0.94	0.02	0	0	0	0	0	0
	CLASS - 11	0	0	0	0	0	0	0	0	0	0	0.99	0.01	0	0	0	0	0
	CLASS - 12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	CLASS - 13	0.01	0	0	0	0.01	0	0	0	0	0	0	0	0.97	0	0.01	0	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	CLASS - 15	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0	0.99	0	0
	CLASS - 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.97	0.03
	CLASS - 17	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																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	96	1	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0
	CLASS - 2	2	90	1	5	0	0	0	0	0	0	1	0	0	0	1	0	0
	CLASS - 3	1	0	97	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	CLASS - 4	2	2	0	91	0	0	1	3	0	1	0	0	0	0	0	0	0
	CLASS - 5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
	CLASS - 6	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
	CLASS - 7	0	0	3	0	0	0	97	0	0	0	0	0	0	0	0	0	0
	CLASS - 8	0	0	1	2	0	0	0	91	0	0	0	6	0	0	0	0	0
	CLASS - 9	0	0	0	0	3	0	0	0	97	0	0	0	0	0	0	0	0
	CLASS - 10	1	1	0	0	0	1	0	0	1	94	2	0	0	0	0	0	0
	CLASS - 11	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0
	CLASS - 12	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
	CLASS - 13	1	0	0	0	1	0	0	0	0	0	0	0	97	0	1	0	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
	CLASS - 15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	99	0	0
	CLASS - 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	3
	CLASS - 17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	99

Figure 10: Confusion Matrix for Train Data set

		PREDICTED LABELS																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	0.12	0.18	0.04	0.21	0.01	0.03	0.02	0.05	0.05	0.11	0.02	0.04	0	0.05	0.06	0.01	0
	CLASS - 2	0.12	0.19	0.02	0.15	0.01	0.1	0.07	0.08	0.03	0.08	0.03	0.02	0.03	0.03	0.02	0.01	0.01
	CLASS - 3	0.02	0.02	0.61	0	0	0.03	0.26	0	0.01	0.01	0	0.01	0.01	0	0.01	0	0.01
	CLASS - 4	0.07	0.1	0.02	0.37	0	0.04	0.06	0.11	0	0.05	0.03	0.04	0.01	0.03	0.04	0.01	0.02
	CLASS - 5	0	0	0	0	0.9	0	0	0	0.02	0	0	0	0.04	0.02	0	0.02	0
	CLASS - 6	0.09	0.06	0.04	0.04	0.03	0.36	0.09	0.02	0.03	0.08	0.11	0	0.02	0	0.01	0	0.02
	CLASS - 7	0	0.03	0.14	0.02	0	0.02	0.7	0.01	0	0.01	0.01	0.02	0.01	0	0	0.01	0.02
	CLASS - 8	0.01	0.04	0.01	0.06	0.01	0	0.02	0.46	0	0	0	0.37	0	0	0.01	0.01	0
	CLASS - 9	0	0.03	0.02	0	0.16	0.01	0.01	0	0.63	0	0.01	0	0.01	0	0.02	0.06	0.04
	CLASS - 10	0.05	0.14	0	0.09	0	0.05	0.01	0.03	0.01	0.24	0.21	0.06	0.05	0.03	0	0.01	0.02
	CLASS - 11	0	0.02	0	0.02	0	0.11	0	0	0.02	0.09	0.69	0.02	0.01	0	0	0	0.02
	CLASS - 12	0	0	0	0	0	0	0	0.07	0	0	0	0.93	0	0	0	0	0
	CLASS - 13	0	0	0	0	0	0.01	0.01	0	0.01	0	0	0	0.71	0.11	0.14	0.01	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.98	0	0	0
	CLASS - 15	0	0.01	0	0	0	0.01	0	0	0	0.01	0	0	0.12	0.01	0.81	0.03	0
	CLASS - 16	0.01	0.01	0.01	0	0	0	0	0	0	0	0	0	0.02	0	0.02	0.86	0.07
	CLASS - 17	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

		PREDICTED LABELS																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	12	18	4	21	1	3	2	5	5	11	2	4	0	5	6	1	0
	CLASS - 2	12	19	2	15	1	10	7	8	3	8	3	2	3	3	2	1	1
	CLASS - 3	2	2	61	0	0	3	26	0	1	1	0	1	1	0	1	0	1
	CLASS - 4	7	10	2	37	0	4	6	11	0	5	3	4	1	3	4	1	2
	CLASS - 5	0	0	0	0	90	0	0	0	2	0	0	0	4	2	0	2	0
	CLASS - 6	9	6	4	4	3	36	9	2	3	8	11	0	2	0	1	0	2
	CLASS - 7	0	3	14	2	0	2	70	1	0	1	1	2	1	0	0	1	2
	CLASS - 8	1	4	1	6	1	0	2	46	0	0	0	37	0	0	1	1	0
	CLASS - 9	0	3	2	0	16	1	1	0	63	0	1	0	1	0	2	6	4
	CLASS - 10	5	14	0	9	0	5	1	3	1	24	21	6	5	3	0	1	2
	CLASS - 11	0	2	0	2	0	11	0	0	2	9	69	2	1	0	0	0	2
	CLASS - 12	0	0	0	0	0	0	0	7	0	0	0	93	0	0	0	0	0
	CLASS - 13	0	0	0	0	0	1	1	0	1	0	0	0	71	11	14	1	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	2	98	0	0	0
	CLASS - 15	0	1	0	0	0	1	0	0	0	1	0	0	12	1	81	3	0
	CLASS - 16	1	1	1	0	0	0	0	0	0	0	0	0	2	0	2	86	7
	CLASS - 17	0	0	4	0	0	0	0	0	4	1	0	0	0	0	0	9	82

Figure 12: Confusion Matrix for Test Data set

3.1.3 Taiji pose Dataset

As explained in section 2.1.3, this dataset contains 8 classes each of which contains 64 features/dimensions. For the sake of visualization, only two of the features have been considered, so that the data points can be plotted on a 2-D space. The data points, as shown in Figure 13, overlap over one another and aren't linearly separable in the 2 dimensional subspace. Therefore the reduction of dimension from 64 to 2 results in loss of information as well as linear separability.

After applying least squares classification, the linear discriminants constructed using one vs one scheme over the test data points are as shown in Fig. 14. The visualization is poor and does not convey any useful information. For computational convenience, the classes in the wallpaper dataset have been converted to an array of numbers from 1 to 8 as shown below:

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

Figure 15 and 16 depicts the classification matrix and confusion matrix for the training data respectively. Figure 17 and Figure 18 depicts the same for the test data.

Using the confusion matrix, the accuracy for training and testing is calculated and is as follows for the entire dataset:

Training accuracy - 96.1304%

Testing accuracy - 92.8068%

Thus it can be seen that the least square classifier performs decently on taiji pose dataset.

In Figure 13 and Figure 14 those portions of thick edged lines are seen since there are a large number of data at the same position, thereby making it not look well to visualize.

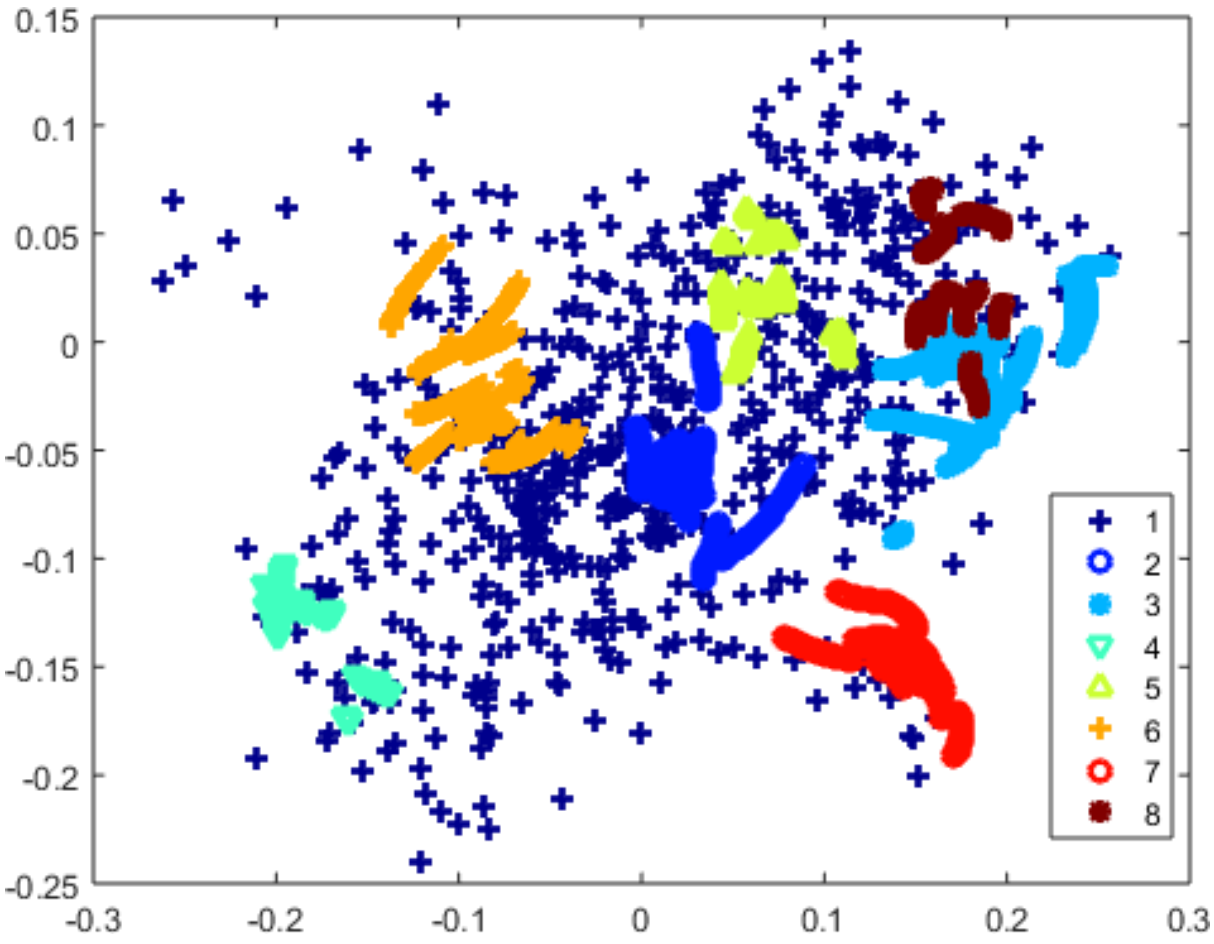

Figure 13: Test Data Points in a 2-D subspace for wallpaper data set using features 1 and 7.

Also, it can be noticed from Figure 15 through Figure 18 thatthe predicted labels for the case of class 1, the predicted labels seem to have portions in every other class. It can be visualized from the Figure 13 that while all the other classes seem to form a cluster, the class 1 which is denoted by the blue + are scaterred throughout the entire plane. Therefore while the other classes seem to be linearly separable, Class 1 seem to have its share in every other class' cluster. Hence the results obtained in 15 through Figure 18 is justified.

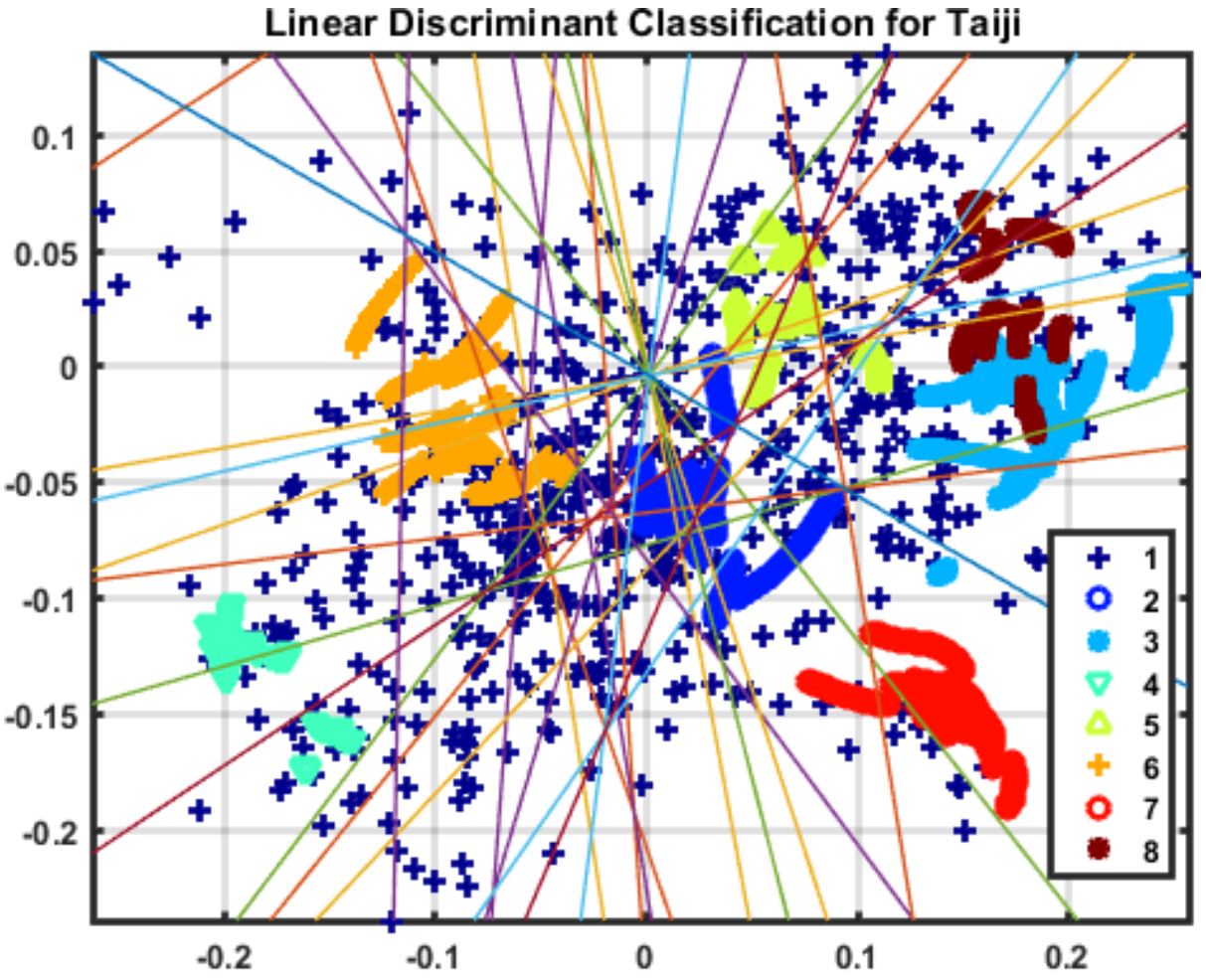

Figure 14: Linear Discriminant Boundaries over Test Dataset in 2-D subspace constructed using one-vs-one scheme.

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	0.6904	0.0357	0.0617	0.0334	0.0368	0.0521	0.0294	0.0606
	CLASS - 2	0	1	0	0	0	0	0	0
	CLASS - 3	0	0	1	0	0	0	0	0
	CLASS - 4	0	0	0	1	0	0	0	0
	CLASS - 5	0	0	0	0	1	0	0	0
	CLASS - 6	0	0	0	0	0	1	0	0
	CLASS - 7	0	0	0	0	0	0	1	0
	CLASS - 8	0	0	0	0	0	0	0	1

Figure 15: Classification Matrix for Train Data set

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	1220	63	109	59	65	92	52	107
	CLASS - 2	0	1066	0	0	0	0	0	0
	CLASS - 3	0	0	2132	0	0	0	0	0
	CLASS - 4	0	0	0	1066	0	0	0	0
	CLASS - 5	0	0	0	0	1066	0	0	0
	CLASS - 6	0	0	0	0	0	2132	0	0
	CLASS - 7	0	0	0	0	0	0	1066	0
	CLASS - 8	0	0	0	0	0	0	0	1066

Figure 16: Confusion Matrix for Train Data set

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	0.4245	0.0531	0.0813	0.0431	0.1725	0.0779	0.0415	0.1061
	CLASS - 2	0	1	0	0	0	0	0	0
	CLASS - 3	0	0	1	0	0	0	0	0
	CLASS - 4	0	0	0	1	0	0	0	0
	CLASS - 5	0	0	0	0	1	0	0	0
	CLASS - 6	0	0	0	0	0	1	0	0
	CLASS - 7	0	0	0	0	0	0	1	0
	CLASS - 8	0	0	0	0	0	0	0	1

Figure 17: Classification Matrix for Test Data set

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	256	32	49	26	104	47	25	64
	CLASS - 2	0	369	0	0	0	0	0	0
	CLASS - 3	0	0	738	0	0	0	0	0
	CLASS - 4	0	0	0	369	0	0	0	0
	CLASS - 5	0	0	0	0	369	0	0	0
	CLASS - 6	0	0	0	0	0	738	0	0
	CLASS - 7	0	0	0	0	0	0	369	0
	CLASS - 8	0	0	0	0	0	0	0	369

Figure 18: Confusion Matrix for Test Data set

3.2 Fisher's projection

Fisher's projection is the method in which the feature vector is reduced from a higher dimension to a lower dimension in order to be able to linearly separate the data and increase the efficiency of classification. The advantages of using Fisher's Linear Discriminant Analysis is that by reducing the dimensionality of feature space, we decrease the cost of computation. Also, for reducing the dimensionality, FIsher's LDA provides us a flexibility in choosing a lower order subspace dimension. This method supersedes the least square classification by catering to the issue of outlier data points.

In the algorithm that has been executed on MATLAB, the data is projected onto a $k-1$ dimension where K is the number of classes of the dataset. After the data has been projected, k-Nearest Neighbour classification algorithm has been used in order to obtain the accuracy of the dataset. The algorithm is as explained in Section 2.2.3.

The following sections showcases the classification and confusion matrix of the test data of each of the dataset along with the accuracy obtained using Fisher's projection.

For the case of wine dataset, Figure 19 and Figure 20 shows the classification and confusion matrix. The testing accuracy is calculated to be 97.14% with a standard deviation of 4.95%.

		PREDICTED LABELS		
		CLASS - 1	CLASS - 2	CLASS - 3
GROUND TRUTH	CLASS - 1	1	0	0
	CLASS - 2	0.057143	0.914286	0.028571
	CLASS - 3	0	0	1

Figure 19: Classification Matrix for test data set of Wine from a kNN classifier.

For the case of wallpaper groups dataset, Figure 21 and Figure 22 shows the classification and confusion matrix. The testing accuracy is calculated to be 70.24% with a standard deviation of 23.67%.

For the case of wallpaper groups dataset, Figure 23 and Figure 24 shows the classification and confusion matrix. The testing accuracy is calculated to be 92.87% with a standard deviation of 11.04%.

		PREDICTED LABELS		
		CLASS - 1	CLASS - 2	CLASS - 3
GROUND TRUTH	CLASS - 1	29	0	0
	CLASS - 2	2	32	1
	CLASS - 3	0	0	24

Figure 20: Confusion Matrix for test data set of Wine from a kNN classifier.

		PREDICTED LABELS																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	0.35	0.21	0.02	0.2	0	0.05	0.01	0	0	0.14	0.02	0	0	0	0	0	0
	CLASS - 2	0.21	0.28	0.02	0.16	0	0.09	0.02	0.04	0	0.15	0.03	0	0	0	0	0	0
	CLASS - 3	0.03	0.02	0.7	0.01	0	0.03	0.21	0	0	0	0	0	0	0	0	0	0
	CLASS - 4	0.15	0.14	0.03	0.47	0	0.01	0.06	0.06	0	0.07	0.01	0	0	0	0	0	0
	CLASS - 5	0	0	0	0	0.95	0	0	0	0.01	0	0	0	0.04	0	0	0	0
	CLASS - 6	0.1	0.08	0.03	0.02	0	0.56	0.06	0	0	0.08	0.07	0	0	0	0	0	0
	CLASS - 7	0.04	0.03	0.16	0.03	0	0.05	0.67	0.01	0	0	0.01	0	0	0	0	0	0
	CLASS - 8	0.02	0.05	0	0.1	0	0	0.02	0.59	0	0	0	0.22	0	0	0	0	0
	CLASS - 9	0	0	0	0	0.1	0	0	0	0.81	0	0	0	0	0	0.04	0.03	0.02
	CLASS - 10	0.15	0.22	0	0.05	0	0.01	0.01	0.05	0	0.34	0.13	0.04	0	0	0	0	0
	CLASS - 11	0.03	0.03	0	0.03	0	0.06	0	0	0	0.17	0.68	0	0	0	0	0	0
	CLASS - 12	0	0	0	0	0	0	0	0.09	0	0	0	0.91	0	0	0	0	0
	CLASS - 13	0	0	0	0	0.01	0	0	0	0	0	0	0	0.93	0.02	0.04	0	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	CLASS - 15	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0.91	0.03	0
	CLASS - 16	0	0	0	0	0	0	0	0	0.01	0	0	0	0.02	0	0.03	0.89	0.05
	CLASS - 17	0	0	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0.05	0.9

Figure 21: Classification Matrix for test data set of Wallpaper Groups from a kNN classifier.

		PREDICTED LABELS																
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8	CLASS - 9	CLASS - 10	CLASS - 11	CLASS - 12	CLASS - 13	CLASS - 14	CLASS - 15	CLASS - 16	CLASS - 17
GROUND TRUTH	CLASS - 1	35	21	2	20	0	5	1	0	0	14	2	0	0	0	0	0	0
	CLASS - 2	21	28	2	16	0	9	2	4	0	15	3	0	0	0	0	0	0
	CLASS - 3	3	2	70	1	0	3	21	0	0	0	0	0	0	0	0	0	0
	CLASS - 4	15	14	3	47	0	1	6	6	0	7	1	0	0	0	0	0	0
	CLASS - 5	0	0	0	0	95	0	0	0	1	0	0	0	4	0	0	0	0
	CLASS - 6	10	8	3	2	0	56	6	0	0	8	7	0	0	0	0	0	0
	CLASS - 7	4	3	16	3	0	5	67	1	0	0	1	0	0	0	0	0	0
	CLASS - 8	2	5	0	10	0	0	2	59	0	0	0	22	0	0	0	0	0
	CLASS - 9	0	0	0	0	10	0	0	0	81	0	0	0	0	0	4	3	2
	CLASS - 10	15	22	0	5	0	1	1	5	0	34	13	4	0	0	0	0	0
	CLASS - 11	3	3	0	3	0	6	0	0	0	17	68	0	0	0	0	0	0
	CLASS - 12	0	0	0	0	0	0	0	9	0	0	0	91	0	0	0	0	0
	CLASS - 13	0	0	0	0	1	0	0	0	0	0	0	0	93	2	4	0	0
	CLASS - 14	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
	CLASS - 15	0	0	0	0	0	0	0	0	0	0	0	0	6	0	91	3	0
	CLASS - 16	0	0	0	0	0	0	0	0	1	0	0	0	2	0	3	89	5
	CLASS - 17	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	5	90

Figure 22: Confusion Matrix for test data set of Wallpaper Groups from a kNN classifier.

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	0.8076	0.0133	0.0348	0.0232	0.0498	0.0332	0.0199	0.0182
	CLASS - 2	0.2927	0.7073	0	0	0	0	0	0
	CLASS - 3	0.0447	0	0.9553	0	0	0	0	0
	CLASS - 4	0.0352	0	0	0.9648	0	0	0	0
	CLASS - 5	0	0	0	0	1	0	0	0
	CLASS - 6	0	0	0	0	0	1	0	0
	CLASS - 7	0	0	0	0	0	0	1	0
	CLASS - 8	0.0054	0	0	0	0	0	0	0.9946

Figure 23: Classification Matrix for test data set of Taiji pose from a kNN classifier.

		PREDICTED LABELS							
		CLASS - 1	CLASS - 2	CLASS - 3	CLASS - 4	CLASS - 5	CLASS - 6	CLASS - 7	CLASS - 8
GROUND TRUTH	CLASS - 1	487	8	21	14	30	20	12	11
	CLASS - 2	108	261	0	0	0	0	0	0
	CLASS - 3	33	0	705	0	0	0	0	0
	CLASS - 4	13	0	0	356	0	0	0	0
	CLASS - 5	0	0	0	0	369	0	0	0
	CLASS - 6	0	0	0	0	0	738	0	0
	CLASS - 7	0	0	0	0	0	0	369	0
	CLASS - 8	2	0	0	0	0	0	0	367

Figure 24: Confusion Matrix for test data set of Taiji pose from a kNN classifier.

4 Conclusion

After performing this project, we can come to a set of conclusions based on the results obtained in the previous section. It is observed that although least squares are used for regression problems, they seem to perform well for certain datasets for the case of classification as well. It generates a closed form solution for the parameter matrix W. However, it fails in the case where there are outlier points and the least squares algorithm penalizes the points that are "too correct" *i.e.* if they lie on the same side of the class but far away from the cluster. The method of least squares classification was found to be working well in the case of wine and taiji pose datasets while it performed poorly for the wallpaper group dataset.

On the other hand, Fisher's linear discriminant analysis was used as a pre-processing step to reduce the dimensions of the data that is being projected onto in order to obtain well separated clusters of data points. It is observed that the accuracies improve for the case of wallpaper groups and taiji pose and doesn't change much for the case of wine dataset. Therefore, it can be concluded that the classification accuracies for test datasets increase by using Fisher's data points.

References

[1] Bishop, Christopher M. *Pattern Recognition and Machine Learning.* 2006.

[2] Prince, Simon JD *Computer vision: models, learning, and inference.* 2012

[3] Lichman, M.(2013) UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science