Supervised Classifiers

Skanda Bharadwaj

June 2, 2019

Supervised classifiers are built to address multi-class problems. Classifiers are built using linear models. The project also explores one of the dimensionality reduction techniques called Fisher's projection, widely known as the Linear Discriminant Analysis (LDA). The project discusses the results of classifying three different data sets namely, wine, wallpaper and the taiji data set along with the inferences of the results obtained with and without using dimensionality reductions.

Contents

1 Introduction

Supervised classifiers are machine learning techniques that learn specific function from a given input-output pairs more commonly referred to as training data and labels of the data set. Classifiers are expected to map a given new observation of the data set into the respective class having learnt the trend in the training data. In this project we consider discriminant functions to classify the data. Firstly, we explore least-squares method to solve multi-class problems by considering k-class discriminant function. We then explore a dimensionality reduction techniques which is widely referred to as Linear Discriminant Analysis (LDA) which is a generalisation of Fisher's discriminant analysis. This helps greatly in maximising the separation between the classes in order to better learn the data set. Fisher's discriminant analysis is then applied on both KNN and least squares to analyse the results.

2 Data

In this section we discuss the data we are interested in detail.

2.1 Wine

Wine recognition data is multi-class data set. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The parameters of the wine data set is as follows -

- Number of Classes : 3
- Number of Features : 13
- Number of Training Observations: 90
- Observations per class in training dataset:
  - Class 1 : 30
  - Class 2 : 36
  - Class 3 : 24
- Number of Testing Observations: 88
- Observations per class in testing dataset:
  - Class 1 : 29
  - Class 2 : 35
  - Class 3 : 24

3 Discriminant Functions

3.1 Least Squares For Classification

We first explore one of the simplest discriminant functions – Linear Discriminant Function, given by

$$y(x) = \mathbf{w}^T\mathbf{x} + w_0 \tag{1}$$

Where, $\mathbf{w}$ is the weight vector and w_0 is the bias. An input vector x is assigned to class C_1 if $y(x) \geq 0$ and to class C_2 otherwise. The corresponding decision boundary is simply $y(x) = 0$, which corresponds to a $(D-1)$ - dimensional hyperplane within the D - dimensional input space. A more compact form of the above equation can be written as

$$y(x) = \widetilde{\mathbf{W}}^T\widetilde{\mathbf{x}} \tag{2}$$

where, $\widetilde{\mathbf{W}} = (w_0, \mathbf{w})$ and $\widetilde{\mathbf{x}} = (1, \mathbf{x})$. Using the above equation we construct a single $K-$ class discriminant comprising of K linear equations of the form

$$y_k(x) = \mathbf{w_k}^T\mathbf{x} + w_{k0} \tag{3}$$

and then assigning a point x to class C_k if $y_k(x) > y_j(x)$ for all $j \neq k$. The decision boundary between class C_k and class C_j is therefore given by $y_k(x) = y_j(x)$ and hence corresponds to a $(D-1)$ - dimensional hyperplane given by

$$(\mathbf{w_k} - \mathbf{w_j})^T\mathbf{x} + (w_{k0} - w_{j0}) = 0 \tag{4}$$

Using the above $K-$ class discriminant function we create a classifier using least squares. We know that the closed form solution of the least squares is given by

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^T\mathbf{T} = \widetilde{\mathbf{X}}^\dagger\mathbf{T} \tag{5}$$

where, $\widetilde{\mathbf{X}}^\dagger$ is the pseudo-inverse of the matrix $\widetilde{\mathbf{X}}$ and we obtain the discriminant function as

$$\mathbf{y}(\mathbf{x}) = \mathbf{T}^T(\widetilde{\mathbf{X}}^\dagger)\widetilde{\mathbf{x}} \tag{6}$$

3.2 Fisher's Linear Discriminant

Fisher's linear discriminant, also known as linear discriminant analysis (LDA) is a dimensionality reduction technique to create a linear classification model. LDA primarily aims to maximise a function that will give a large separation between the projected class means while also giving a small variance within each class, thereby minimising the class overlap. The within-class variance of the transformed data from class C_k is therefore given by

$${s_k}^2 = \sum_{n \in C_k} (y_n - m_k)^2 \tag{7}$$

where $y_n = w^T x_n$. Also,

$$m_k = \mathbf{w}^T \mathbf{m_k} \tag{8}$$

$$\mathbf{m_k} = \frac{1}{N_k} \sum_{n \in C_k} \mathbf{x_n} \tag{9}$$

We can define the total within-class variance for the whole data set to be simply ${s_1}^2 + {s_2}^2$. The Fisher criterion is defined to be the ratio of the between-class variance to the within-class variance and is given by

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{{s_1}^2 + {s_2}^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{10}$$

For a multi-class problem, a discriminant function similar to 2 can be used, only without the bias parameter. That can be written as

$$y(x) = \mathbf{W}^T \mathbf{x} \tag{11}$$

Within-class and between-class variances are given by

$$\mathbf{S}_W = \sum_{k=1}^{K} {\mathbf{S}_k}^2 \tag{12}$$

Where,

$$\mathbf{S}_k = \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \tag{13}$$

and

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T \tag{14}$$

where

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \tag{15}$$

$\mathbf{m}_k$ is the mean of all patterns in each class K and $\mathbf{m}$ is the mean of all the patterns in the data set.

It can be seen that, for a multi-class problem the weight vector $\mathbf{w}$ can be found by solving the below Eigen value problem in ${S_w}^{-1} S_B$ given by

$$({S_w}^{-1} S_B)\mathbf{w} = \lambda \mathbf{w} \tag{16}$$

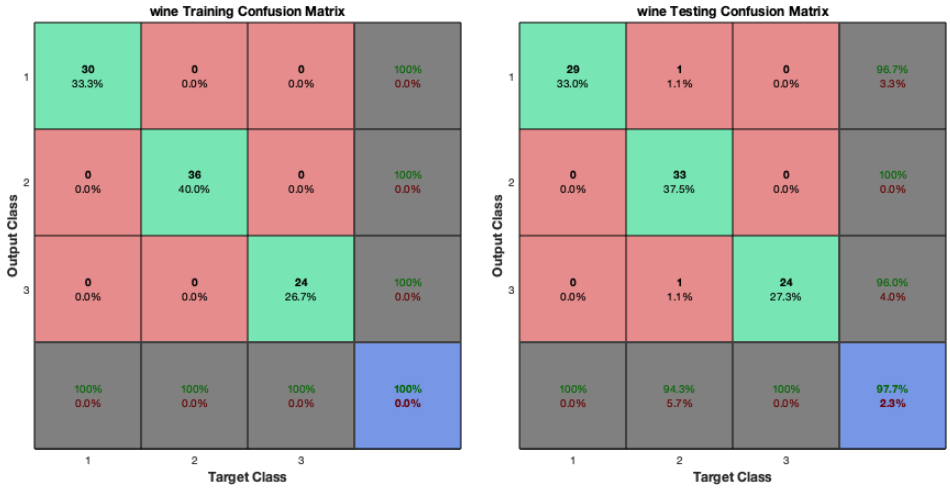
The weight values are determined by those eigenvectors of ${S_w}^{-1} S_B$ that correspond to D' largest eigenvalues.

4 Experiments

This section compares the results obtained by classifying the above-mentioned data sets using the discriminant functions explained in section 3.

4.1 Results

The first set of results given below are the classified results of the wine dataset using a linear discriminant function defined by least squares. Results of a simple classification run on the data set is summarised in Figure 1. Figure 1a and 1b represent the training and testing confusion matrices respectively. The right most column represents the precision and the bottom most row represents the recall. Visualisation is a little tricky when the number of feature dimensions increases more than 3 dimensions. In order to visualise the data, only 2 features were selected at random from the available 13 to visualise the data. Figure 2 completely summarises the results of classifying the wine dataset using $k-$ class discriminant function with K linear functions. Figures 2a and 2b represent the confusion matrices of training and the testing respectively. Figure 2c represents the linear discriminant classification. The 3 lines represent the surface boundaries classifying the respective pair of classes. Similarly, Figure 2d represents the area of classification given the 3 linear functions of the discriminant functions. It is worth nothing that the accuracy increases when all the features are taken into consideration.



a b

Figure 1: Figures a and b represent the confusion matrices of training and testing of the wine data set. All 13 features are considered for classification.

A similar experiment was carried out on the wine dataset, but now applying Fisher's linear discriminant analysis on the training feature vectors. The training feature vectors are first used to find the $(K-1)$ - dimensional hyperplane that satisfies the Fisher's criterion on to which the training features are projected. In the case of wine dataset, the number of classes $K = 1$. So the features vectors are projected on to a $2D$ surface. These projected data are then sent into a classifier for classification. Figure 3 summarises

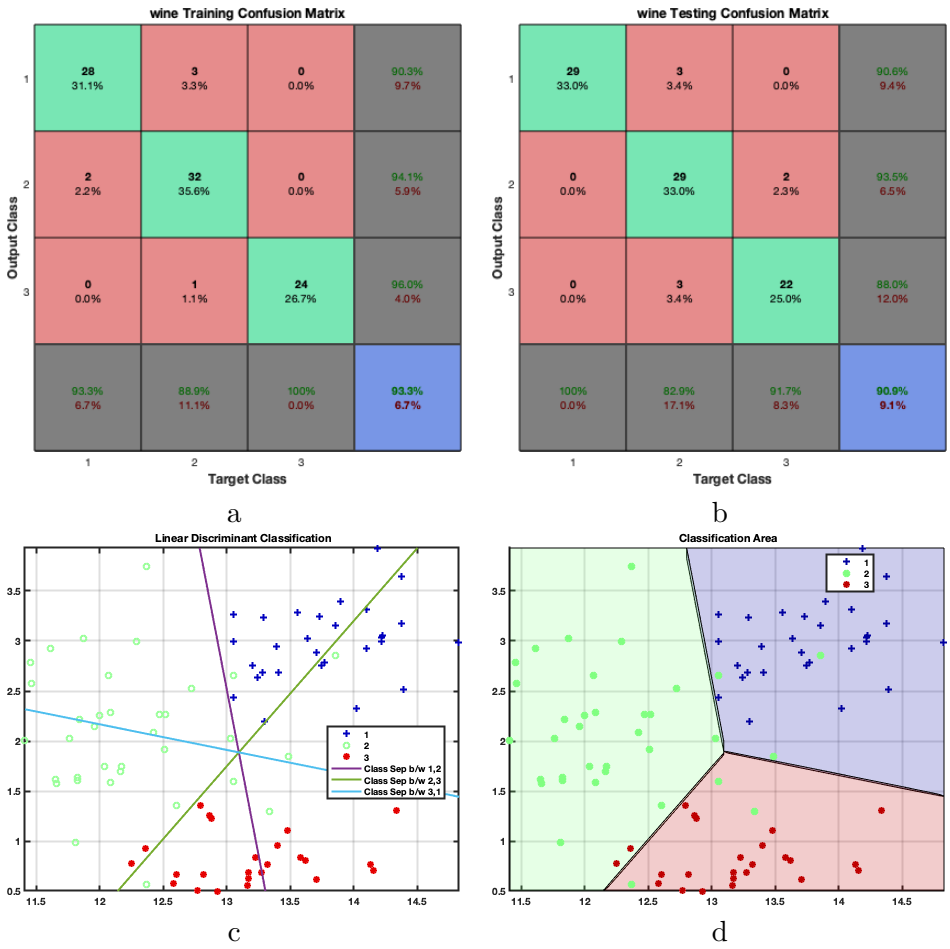


Figure 2: Figures *a* and *b* are the confusion matrices of the classified output. Figure *c* represents the linear discriminant classification. Only 2 features among the 13 features have been considered. 3 linear functions, one for each pair of classes, can been seen classifying the entire dataset. Figure *d* represents classification area as bifurcated by the 3 linear functions.

these results. Figures 3a and 3b represent the confusion matrices and Figures 3c and 3d represent linear discriminant classification and classification area respectively. It can be seen from the figures that the separation between the data is more compared to that in Figure 2. It also worth noting that, even with just 2 dimensional features, LDA attains reasonable accuracy compared to Figure 1.

Figure 4 represents the testing confusion matrix obtained by using a KNN classifier on the projected training data using LDA. The accuracy is further increased using KNN classifier.

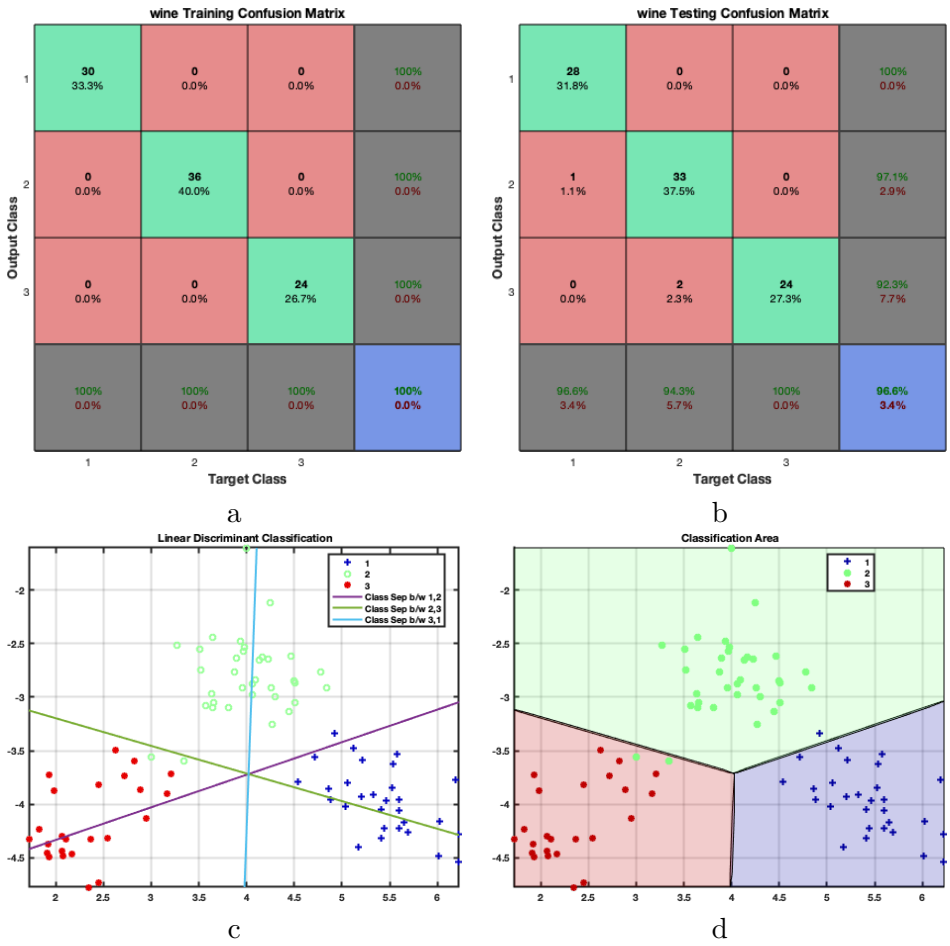


Figure 3: Figures *a* and *b* are the confusion matrices of the classified output. Figure *c* represents the linear discriminant classification. Fisher's projection is applied on the input vectors. It can be seen that the classes are well separated as compared to Figure 2c. Figure *d* represents classification area as bifurcated by the 3 linear functions.

5 Conclusion

Supervised classifiers such as least squares and LDA were applied on three different datasets namely wine, wallpaper and taiji. Discriminant functions were modelled to classify the datasets. Fisher's projection was applied to the feature vectors for dimensionality reduction. Different parameters were compared to understand the nature of data distribution. Analysis, results and inferences were drawn for different permutation of experiments.

6 References

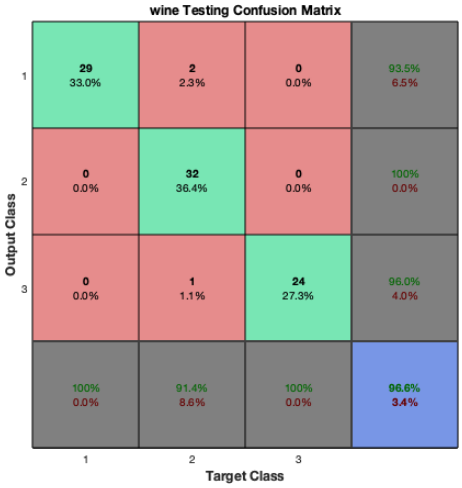


Figure 4: Classification results of wine dataset using KNN classifier with $K = 5$.

References

[1] C. M. Bishop (2006) *Pattern Recognition and Machine Learning*, Springer.

[2] http://archive.ics.uci.edu/ml/datasets/Wine