Foundations of Machine Learning Lab Five

RUN YING JIANG (rj1u20@soton.ac.uk)

I. Introduction

In this lab, I played with different means and covariance of Gaussian and drew the classification boundary by calculating the posterior probabilities and got a better understanding of the Bayes theory. Besides, I implemented Fisher Linear Discriminant Analysis to solve the classification problem which ,drew the ROC curve and computed the AUC value.In the end, I implemented Euclidean distance classifier and Mahalanobis Distance classifier and illustrate the difference between the two kinds of distance-to-mean classifier.

II. CLass Boundaries and Posterier Probabilities

In this section, I plotted contours on the likelihoods of two classes, scatters of 200 samples data and contours on the posterior probability of one of the classes in the following three cases.

- $\boldsymbol{m1} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$ and $\boldsymbol{m2} = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}$, $\boldsymbol{C1} = \boldsymbol{C2} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ P1 = P1 = 0.5 which is **Figure 1 : First graph of contours and scatters**

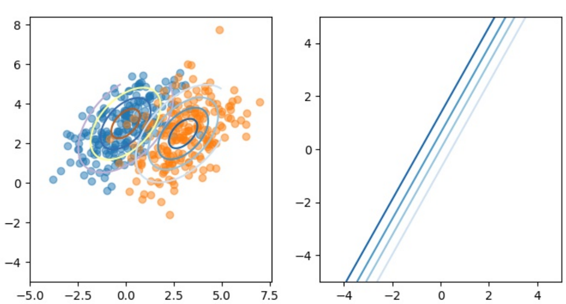

Fig. 1. **First graph of contours and scatters**: The right column shows the contours of the likelihoods and scatters of 200 sampled data while the left columns shows the decision boundary based on the posterior probability.

- $\boldsymbol{m1} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$ and $\boldsymbol{m2} = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}$, $\boldsymbol{C1} = \boldsymbol{C2} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ P1 = 0.7, P2=0.3 **Figure 2 : Second graph of contours and scatters**

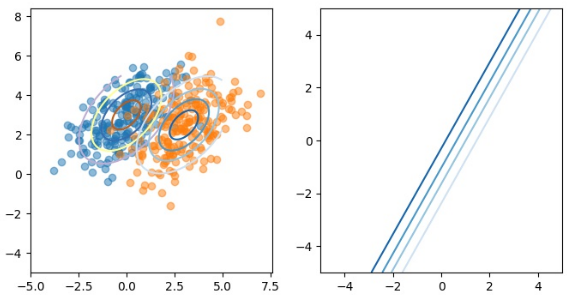

Fig. 2. **Second graph of contours and scatters**: The right column shows the contours of the likelihoods and scatters of 200 sampled data while the left columns shows the decision boundary based on the posterior probability.

- $\boldsymbol{m1} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$ and $\boldsymbol{m2} = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}$, $\boldsymbol{C1} = \begin{pmatrix} 2 & 0 \\ 0 & 2, \end{pmatrix}$, $\boldsymbol{C2} = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5, \end{pmatrix}$ P1 = 0.5, P2=0.5 **Figure 3 : Third graph of contours and scatters**

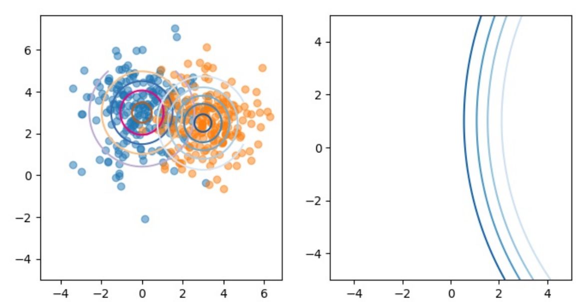

Fig. 3. **Third graph of contours and scatters**: The right column shows the contours of the likelihoods and scatters of 200 sampled data while the left columns shows the decision boundary based on the posterior probability.

By comparing **Figure 1** and **Figure 2** I found the boundary line shifted to the right as I changed the prior probability, which could be explained by the Bayes' theorem :

$$p(w_1|x) = \frac{p(x|w_1)p(w1)}{p(x)}$$

. I assigned a value of 0.7 to the first class and a value of 0.3 to the second class which lead to a higher posterior value of the samples and caused the boundary line to shift to the right. Meanwhile, with the comparison of **Figure 1** and **Figure 3**, I observed the effect of covariance matrices on the direction and size of the contours.

III. Fisher LDA and ROC Curve

In this section , I projected the data onto the Fisher discriminant directions which is $w_F = (C1 + C2)^{-1}(m1 - m2)$ and plotted a histogram of the distribution in the following graph :**Figure 4:Histogram Of Fisher Discriminant Directions** .

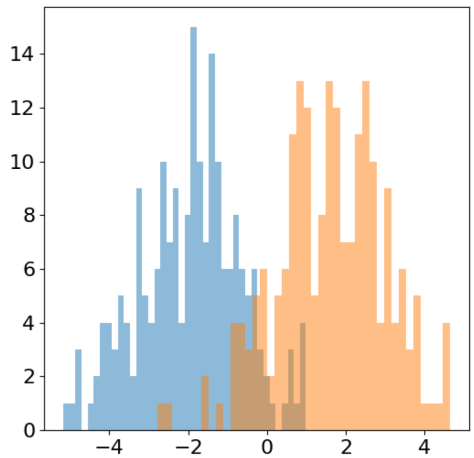

Fig. 4. **Histogram Of Fisher Discriminant Directions** :The histograms in blue shows the distribution of the first class on LDA while the histograms in orange shows the distribution of second class

Then I counted the number of true positive samples and false positive samples in different thresholds to get the ROC curve **Figure 5:ROC curve of LDA** and I imported numpy.trapz from sklearn to compute the AUC value which is the area under ROC curve. I got **0.9687** in AUC criterion. The AUC value ranged from 0 to 1. The better a calssification model performed, it would get a higher score in AUC criterion. And I found the best threshold was **-0.16** with the best accuracy **91.25%**.

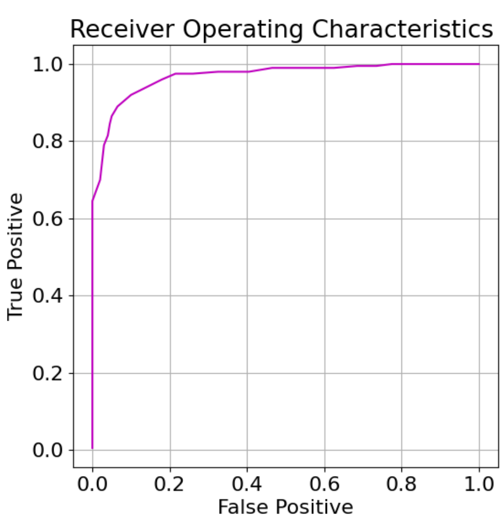

Fig. 5. **ROC curve of LDA**: The vertical axis is the true positive rate and the horizontal axis is the false positive rate in different thresholds.

Then I did the same thing with a random direction **Figure 6: Histogram Of Random** and a direction $m1 + m2$ separately. And I got a graph of histogram **Figure 6: Histogram Of Random** ,**Figure 8: Histogram Of mean** and ROC curve **Figure 7: ROC curve of random direction**, **Figure 9: ROC curve of related direction**. The results are as follow.

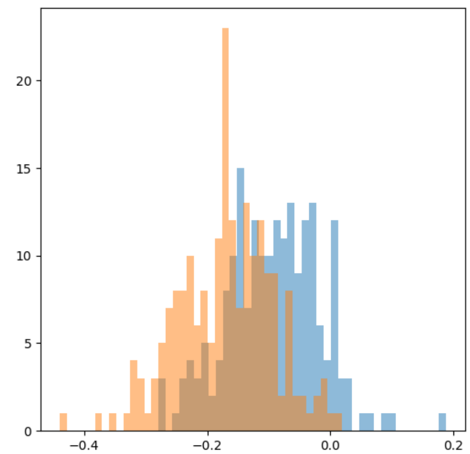

Fig. 6. **Histogram Of Random** :The histograms in blue shows the distribution of the first class on random direction while the histograms in orange shows the distribution of second class

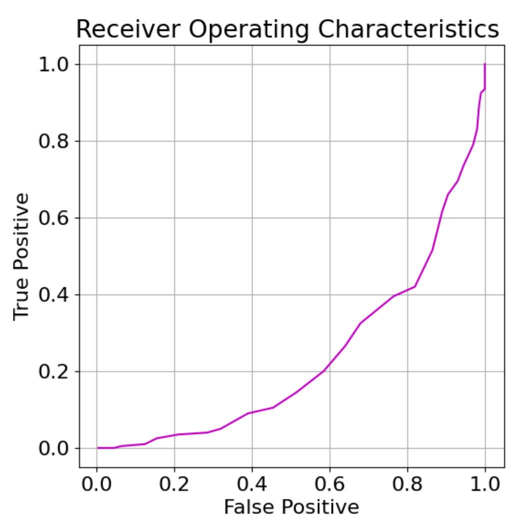

Fig. 7. **ROC curve of random direction**: The vertical axis is the true positive rate and the horizontal axis is the false positive rate in different thresholds.

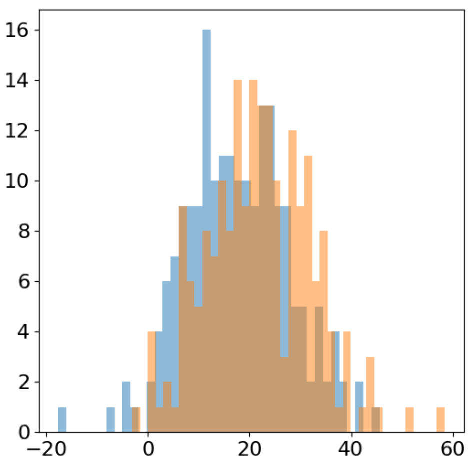

Fig. 8. **Histogram Of related direction** :The histograms in blue shows the distribution of the first class on related direction while the histograms in orange shows the distribution of second class

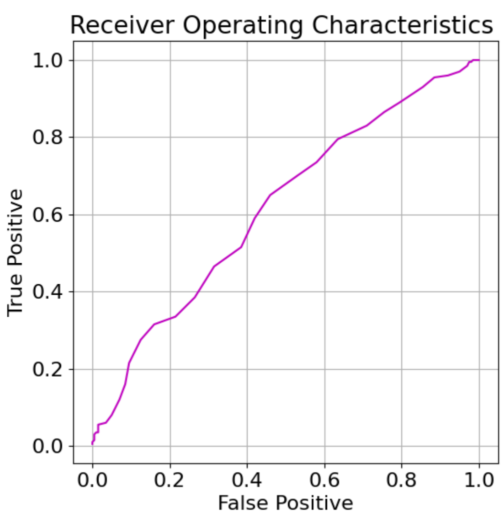

Fig. 9. **ROC curve of related direction**: The vertical axis is the true positive rate and the horizontal axis is the false positive rate in different thresholds.

In conclusion, LDA direction performs better overall and direction with a related value to mean was a little bit better in performance than the random direction. Intuitively, best accuracy , ACU values and best threshold were recorded in the following table.**Table I: The results of different directions**

	accuracy	AUC
LDA	91.25%	0.9687
random	50%	0.2331
m1+m2	51.95%	0.6159

TABLE I
THE RESULT OF DIFFERENT DIRECTIONS

IV. MAHALANOBIS DISTANCE

In this section use distance-to-mean classifier and a Mahalanobis distance-to mean classfier sepearely to classify the 200 sampled data with the mean of $\boldsymbol{m1} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$, $\boldsymbol{m2} = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}$ and a covariance of $\boldsymbol{C1} = \boldsymbol{C2} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. And I recorded the accuracy of this two classifier on training data and test data.**Table II: The results of different classifiers**

	train accuracy	test accuracy
Euclidean	0.8906	0.9125
Mahalanobis	0.9375	0.925

TABLE II
THE RESULT OF DIFFERENT CLASSIFIERS

The Mahalanobis distance calculation (equation 1) differs slightly from euclidean distance (equaiton 2).

$$d(vec(x), vec(y) = \sqrt{(vec(x) - vec(y))^T S^(-1)(vec(x) - vec(y)} \tag{1}$$

$$d(vec(x), vec(y) = \sqrt{(vec(x) - vec(y))^T (vec(x) - vec(y)} \tag{2}$$

The differences of Euclidean Distance and Mahalanobis Distance is that Mahalanobis Distance takes covariance into account.By doing this Mahalanobis remove the redundant information from correlated variables. In most cases the Mahalanobis Distance perfromes a little bit better than Euclidiean Distance in classification problem.