1 Class Boundaries and Posterior Probabilities

The first step is implement the code that make for illustrate probability densities, data and contours on the posterior.

Fig. 1 and Fig. 2 for $m_1 = \begin{bmatrix}0 & 3\end{bmatrix}^T$, $m_2 = \begin{bmatrix}3 & 2.5\end{bmatrix}^T$, $C_1 = C_2 = \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$, $P_1 = P_2 = 0.5$

Fig. 3 and Fig. 4 for $m_1 = \begin{bmatrix}0 & 3\end{bmatrix}^T$, $m_2 = \begin{bmatrix}3 & 2.5\end{bmatrix}^T$, $C_1 = C_2 = \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$, $P_1 = 0.7$, $P_2 = 0.3$

Fig. 5 and Fig. 6 for $m_1 = \begin{bmatrix}0 & 3\end{bmatrix}^T$, $m_2 = \begin{bmatrix}3 & 2.5\end{bmatrix}^T$, $C_1 = \begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$, $C_2 = \begin{bmatrix}1.5 & 0\\ 0 & 1.5\end{bmatrix}$, $P_1 = P_2 = 0.5$

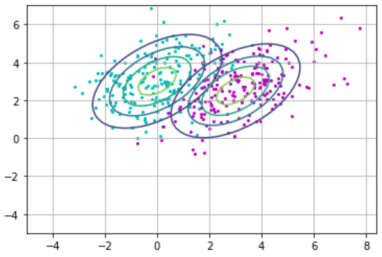

Figure 1: First example of probability densities and data

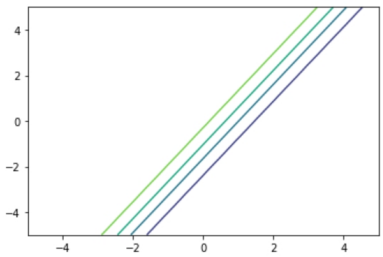

Figure 2: First example of contours on the posterior probability $P[w_1|x]$

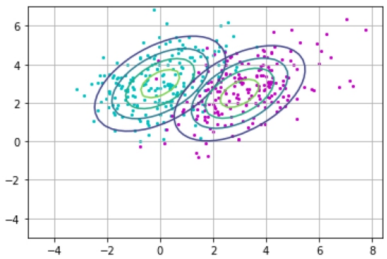

Figure 3: Second example of probability densities and data

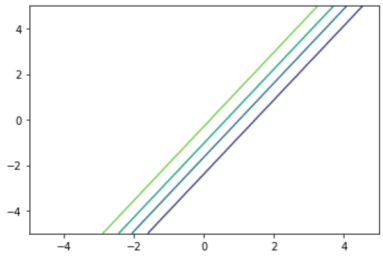

Figure 4: Second example of contours on the posterior probability $P[w_1|x]$

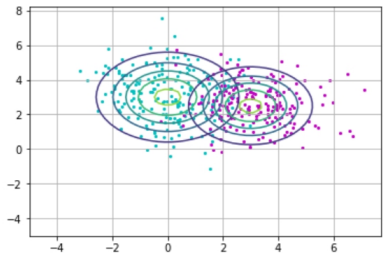

Figure 5: Third example of probability densities and data

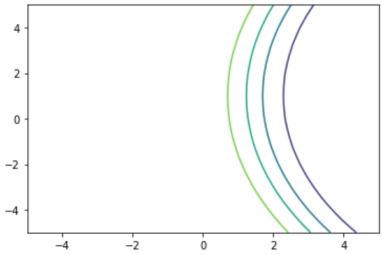

Figure 6: Third example of contours on the posterior probability $P[w_1|x]$

The probability densities, data and contours on the posterior that try more: Fig. 7 and Fig. 8 for $m_1 = \begin{bmatrix}2.4 & 3.2\end{bmatrix}^T$, $m_2 = \begin{bmatrix}1.2 & 0.2\end{bmatrix}^T$, $C_1 = \begin{bmatrix}2 & -1\\ -1 & 2\end{bmatrix}$, $C_2 = \begin{bmatrix}2 & 0\\ 0 & 4\end{bmatrix}$, $P_1 = P_2 = 0.5$

Fig. 9 and Fig. 10 for $m_1 = \begin{bmatrix}2.4 & 3.2\end{bmatrix}^T$, $m_2 = \begin{bmatrix}1.2 & 0.2\end{bmatrix}^T$, $C_1 = \begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$, $C_2 = \begin{bmatrix}2 & 0\\ 0 & 4\end{bmatrix}$, $P_1 = P_2 = 0.5$

Fig. 11 and Fig. 12 for $m_1 = \begin{bmatrix}0 & 0\end{bmatrix}^T$, $m_2 = \begin{bmatrix}0 & 4\end{bmatrix}^T$, $C_1 = \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$, $C_2 = \begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}$, $P_1 = P_2 = 0.5$

Fig. 13 and Fig. 14 for $m_1 = \begin{bmatrix}0 & 3\end{bmatrix}^T$, $m_2 = \begin{bmatrix}3 & 3\end{bmatrix}^T$, $C_1 = \begin{bmatrix}2 & -1\\ -1 & 2\end{bmatrix}$, $C_2 = \begin{bmatrix}2 & -1\\ -1 & 2\end{bmatrix}$, $P_1 = P_2 = 0.5$

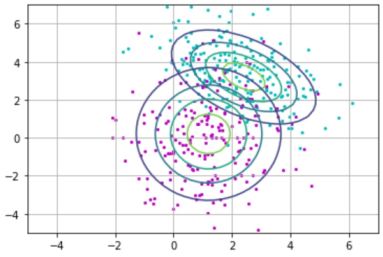

Figure 7: Fourth example of probability densities and data

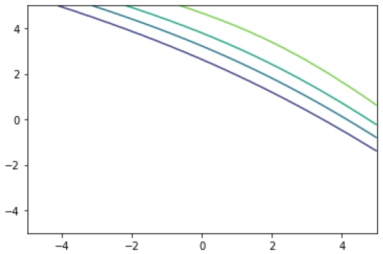

Figure 8: Fourth example of contours on the posterior probability $P[w_1|x]$

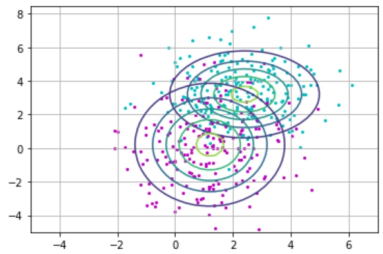

Figure 9: Fifth example of probability densities and data

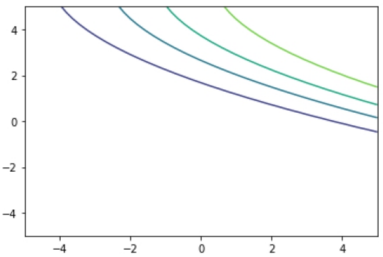

Figure 10: Fifth example of contours on the posterior probability $P[w_1|x]$

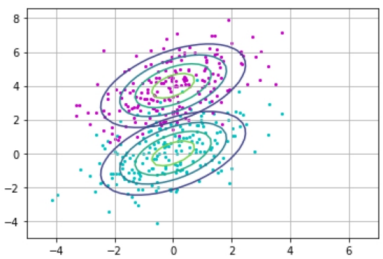

Figure 11: Sixth example of probability densities and data

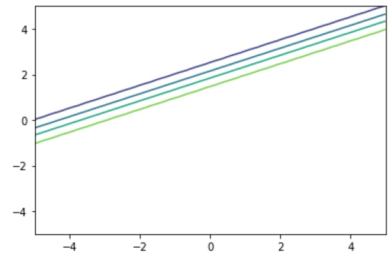

Figure 12: Sixth example of contours on the posterior probability $P[w_1|x]$

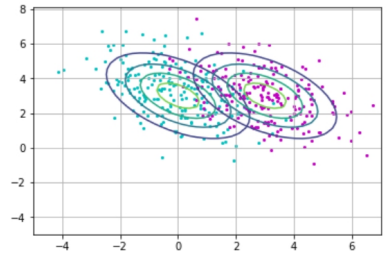

Figure 13: Seventh example of probability densities and data

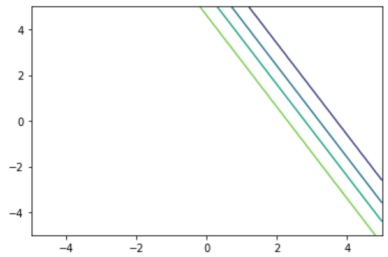

Figure 14: Seventh example of contours on the posterior probability $P[w_1|x]$

The scatter of data will change the position follow the mean of it and tilted follow it covariance matrix. All of this plot is good enough to preliminary evaluation class boundary by human.

2 Fisher LDA and ROC Curve

In fig. 15 is illustrate probability densities, data for used too find Fisher LDA and ROC Curve. Fig. 16 is show direction of Fisher discriminant by red arrow, the blue arrow is the direction of projections onto the direction connecting the means of the two classes and the green arrow is the direction of a random direction.

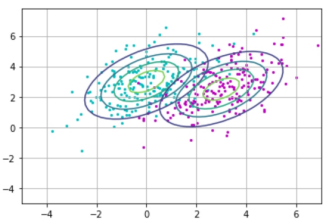

Figure 15: Example of probability densities and data for Fisher LDA and ROC Curve

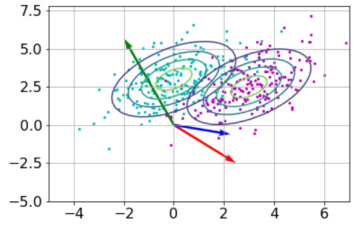

Figure 16: The vector of every Fisher LDA and ROC Curve

Next, histogram of the distribution of projection of Fisher Linear Discriminant direction and it ROC curve show in Fig. 17 and Fig. 18, respectively. It AUC is 0.68735626374 and classification accuracy is 0.89108910891.

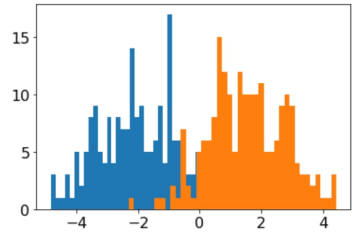

Figure 17: Histograms of the distribution of projections for Fisher LDA

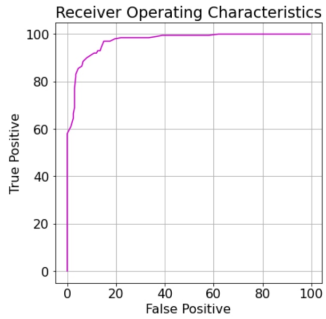

Figure 18: ROC Curve for Fisher LDA

For the Fig. 19 and Fig. 20 are histograms of the distribution of projections for a random direction and it ROC curve. It AUC is 0.62876485216 and classification accuracy is 0.85714285714.

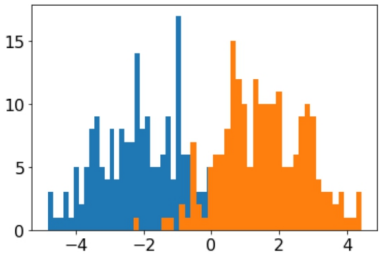

Figure 19: Histograms of the distribution of projections for a random direction (instead of the Fisher discriminant direction)

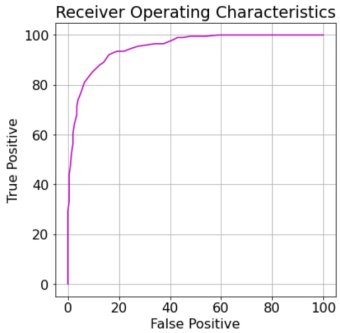

Figure 20: ROC Curve for a random direction (instead of the Fisher discriminant direction)

Last, histograms of the distribution of projections for projections onto the direction connecting the means of the two classes and it ROC curve illustrate in Fig. 21 and Fig. 22, respectively. It AUC is 0.38010017871 and classification accuracy is 0.24137931034.

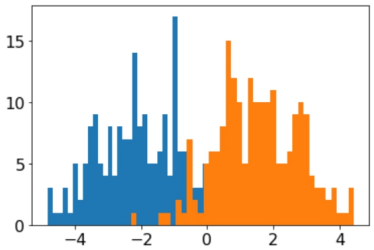

Figure 21: Histograms of the distribution of projections for projections onto the direction connecting the means of the two classes

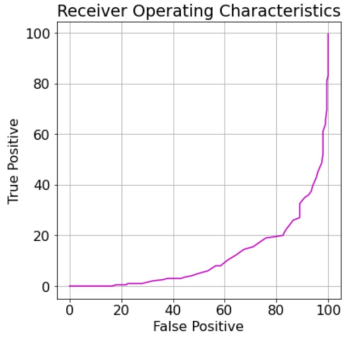

Figure 22: ROC Curve for projections onto the direction connecting the means of the two classes

AUC from this section is find by used numpy.trapz() function [1] from Numpy library and it work by Trapezoidal Rule that is a technique for approximating the definite integral. The result of AUC from area under the curve divided by area of all area under the graph and find area by use numpy.trapz() function.

3 Mahalanobis Distance

From Fig. 23, the left picture show the the scatter plot of dataset that less covariance when we find distance of point one and point two that are the same distance we can classify its is in this data set class. However, if we add more covariance into data set and point one , point two is the same distance we can found point one is outside. Mahalanobis Distance can fix this problem by standardization by use covariance matrix. We can see performance of Mahalanobis distance-to-mean classifier better than distance-to-mean classifier(Euclidean distance) in data set that have covariance as present in Fig. 24 and Fig. 25.

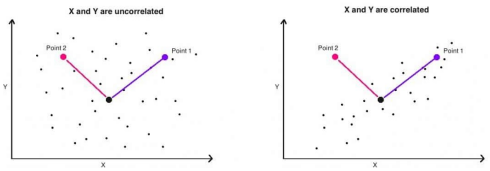

Figure 23: Example of explain Mahalanobis distance-to-mean classifier [2]

```
AUROC:  0.6931328320802005

Confusion Matrix:
 [[61 44]
 [14 81]]

Accuracy Score:  0.71
```

Figure 24: Performance of distance-to-mean classifier(Euclidean distance)

```
AUROC:  0.6672681704260651

Confusion Matrix:
 [[63 42]
 [28 67]]

Accuracy Score:  0.65
```

Figure 25: Performance of Mahalanobis distance-to-mean classifier

Reference

[1]: Trapezoidal Rule function in Numpy library
`https://numpy.org/doc/stable/reference/generated/numpy.trapz.html`
[2]: Picture of example of explain Mahalanobis distance-to-mean classifier
`https://www.machinelearningplus.com/statistics/mahalanobis-distance/#:~:text=1.-,Introduction,and%20more%20untapped%20use%20cases`