AVD613 - Machine Learning for Signal Processing

Assignment - 2 (Programming)

Classification and Regression

Submitted by

Vatsalya Gupta
SC19B098
B.Tech. ECE VII Semester

Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram - 695 547
October 2022

1 Optical Recognition of Handwritten Digits Dataset 1

- The training data was read into features and labels, corresponding to the handwritten digit value (5 or 6) as given in the labels csv.
- The number of 5's and 6's were counted and found out to be 396 and 381 respectively. Accordingly, the priors were found for both the classes as the number of occurrences divided by the total number of samples. The prior for 5 was found to be 0.5096 and for 6 was 0.4903.
- The sufficient statistics (mean and covariance) for the sample data were calculated for both the classes.
- Using Bayes' theory, the posteriors were found using likelihood and priors to obtain a classifier.
- The confusion matrix was calculated to be

$$CM = \begin{bmatrix} \text{True Positives} & \text{False Positives} \\ \text{False Negatives} & \text{True Negatives} \end{bmatrix} = \begin{bmatrix} 151 & 49 \\ 27 & 106 \end{bmatrix}$$

- We define the elements of the confusion matrix as follows
  TP = True Positive (expected = 6, classified = 6)
  TN = True Negative (expected = 5, classified = 5)
  FP = False Positive (expected = 5, classified = 6)
  FN = False Negative (expected = 6, classified = 5)
- Accuracy for different classes is found out as given below

$$\text{Accuracy for 6} = \frac{TP}{FN + TP} = 0.8483$$

$$\text{Accuracy for 5} = \frac{TN}{TN + FP} = 0.6839$$

$$\text{Total Accuracy} = \frac{TP + TN}{FN + TP + TN + FP} = 0.7718$$

- The misclassification rates were calculated from the confusion matrix as

$$\text{Misclassification rate for 6} = \frac{FN}{FN + TP} = 0.1517$$

$$\text{Misclassification rate for 5} = \frac{FP}{TN + FP} = 0.6839$$

Please refer the IPython notebook for the detailed results.

2 Optical Recognition of Handwritten Digits Dataset 2

- The training data was read into features, corresponding to the labels as given in the labels csv.
- The number of instances of both classes were counted. Accordingly, the priors were found for both the classes as the number of occurrences divided by the total number of samples.
- The sufficient statistics (mean and covariance) for the sample data were calculated for both the classes.
- For the first case, equal diagonal σ_s of equal variances along both dimensions, the metrics were as follows

$$CM = \begin{bmatrix} 40 & 3 \\ 0 & 47 \end{bmatrix}$$

$$\text{Misclassification rate for 1} = \frac{FN}{FN + TP} = 0.0$$

$$\text{Misclassification rate for 0} = \frac{FP}{TN + FP} = 0.06$$

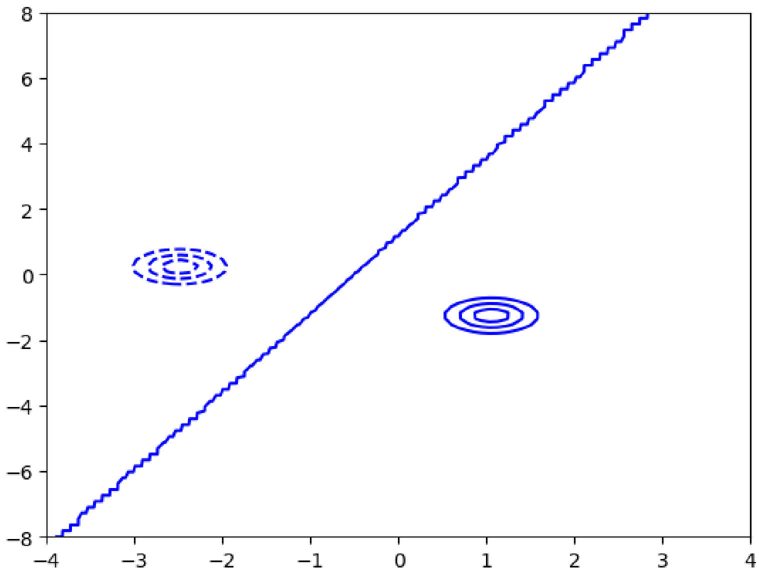

Figure 1: Discriminant and iso-probability contours for equal variances and equal diagonal.

- For the second case, equal diagonal σ_s with unequal variances along different dimensions, the metrics were as follows

$$CM = \begin{bmatrix} 39 & 0 \\ 1 & 50 \end{bmatrix}$$

$$\text{Misclassification rate for 1} = \frac{FN}{FN + TP} = 0.025$$

$$\text{Misclassification rate for 0} = \frac{FP}{TN + FP} = 0.0$$

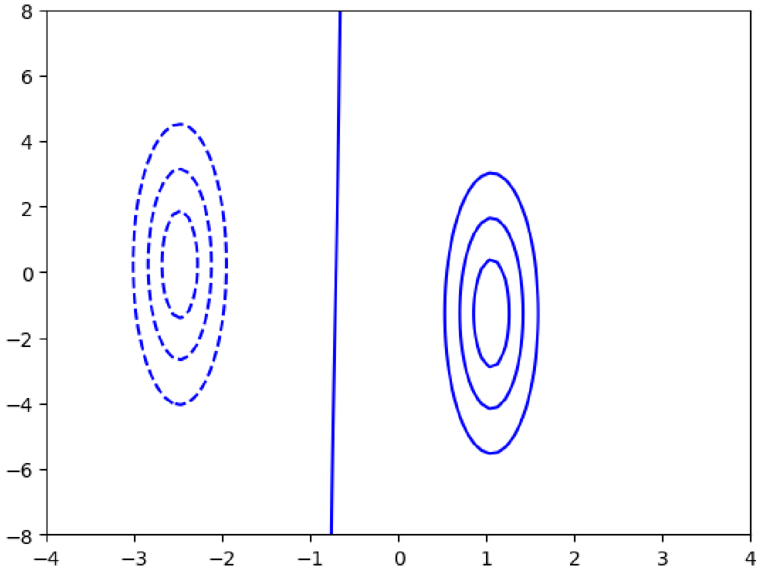

Figure 2: Discriminant and iso-probability contours for equal variances and different diagonal.

- For the third case, arbitrary σ_s but shared by both classes, the metrics were as follows

$$CM = \begin{bmatrix} 39 & 0 \\ 1 & 50 \end{bmatrix}$$

$$\text{Misclassification rate for 1} = \frac{FN}{FN + TP} = 0.025$$

$$\text{Misclassification rate for 0} = \frac{FP}{TN + FP} = 0.0$$

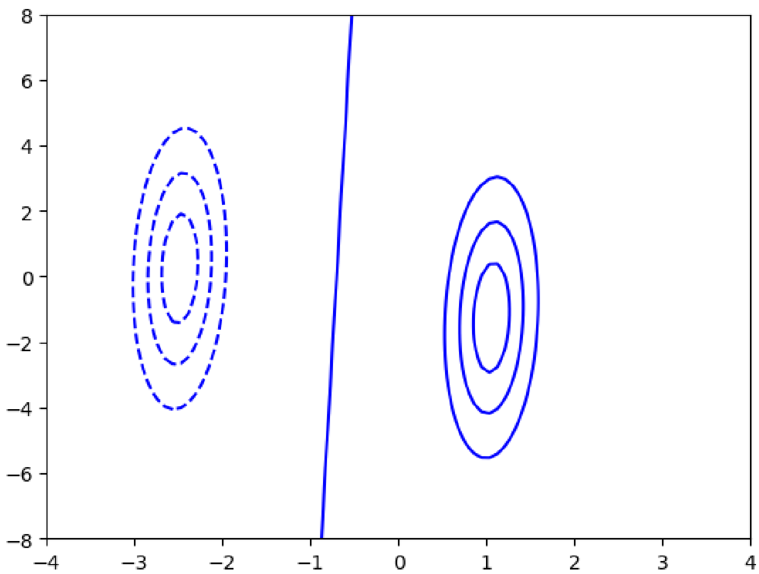

Figure 3: Discriminant and iso-probability contours for equal variances and arbitrary values.

- For the fourth case, different arbitrary σ_s for the two classes, the metrics were as follows

$$CM = \begin{bmatrix} 40 & 0 \\ 0 & 50 \end{bmatrix}$$

$$\text{Misclassification rate for } 1 = \frac{FN}{FN + TP} = 0.0$$

$$\text{Misclassification rate for } 0 = \frac{FP}{TN + FP} = 0.0$$

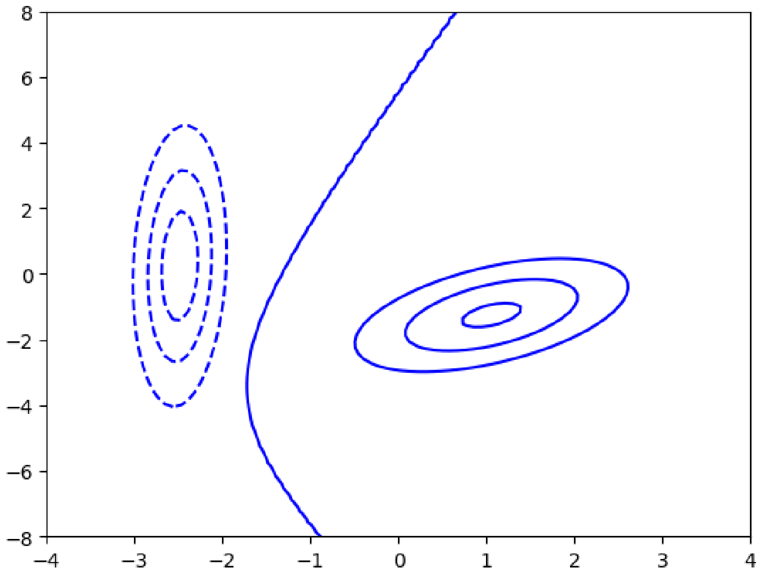

Figure 4: Discriminant and iso-probability contours for different variances and arbitrary values.

- The first three cases give linear separating boundary with finite number of misclassifications. However, the fourth case gives a non-linear boundary which gives better results since, the misclassifications become 0.

Please refer the IPython notebook for the detailed results.

3 Wage Dataset

- We have been given a dataset. We try to fit a polynomial for studying the effect of features on the Wage.

- For an n^{th} degree polynomial, we use the fact that every data point should satisfy the fitting polynomial equation. This way, with some error, we can find the coefficients of the fitting polynomial using system of equations as $\mathbf{A} \cdot b = c$, where c is the array containing the required polynomial weights. $\mathbf{A}, b$ represent the matrix and array respectively, corresponding to the given feature and label values.

- For Wage vs Age, a $6th$ order polynomial is fit, with coefficients as follows (highest to lowest degree)

  $[2.58{\times}10^{-12}, -4.94{\times}10^{-10}, 2.31{\times}10^{-07}, 8.11{\times}10^{-04}, -1.65{\times}10^{-01}, 1.01{\times}10^{+01}, -7.45{\times}10^{+01}]$

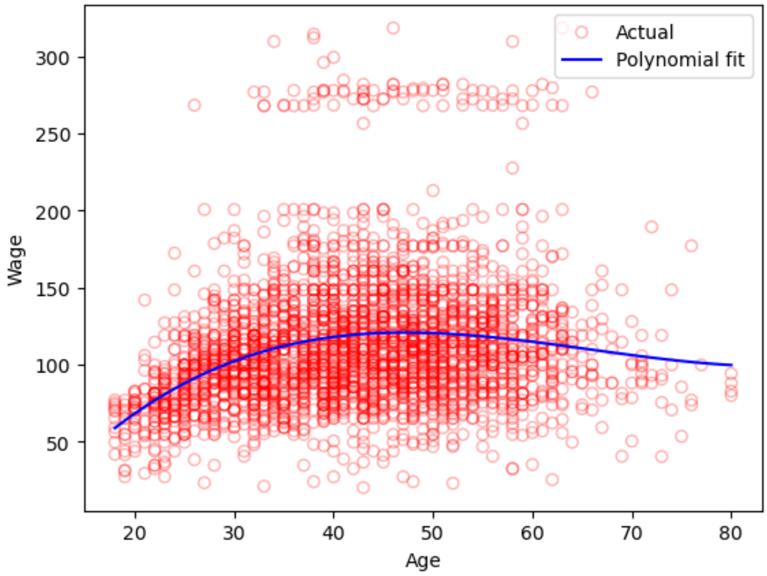

Figure 5: Actual data and polynomial fit for Wage vs Age.

- For Wage vs Year, a $3rd$ order polynomial is fit, with coefficients as follows (highest to lowest degree)

$$[-7.17 \times 10^{-20}, -1.88 \times 10^{-01}, 7.58 \times 10^{+02}, -7.62 \times 10^{+05}]$$

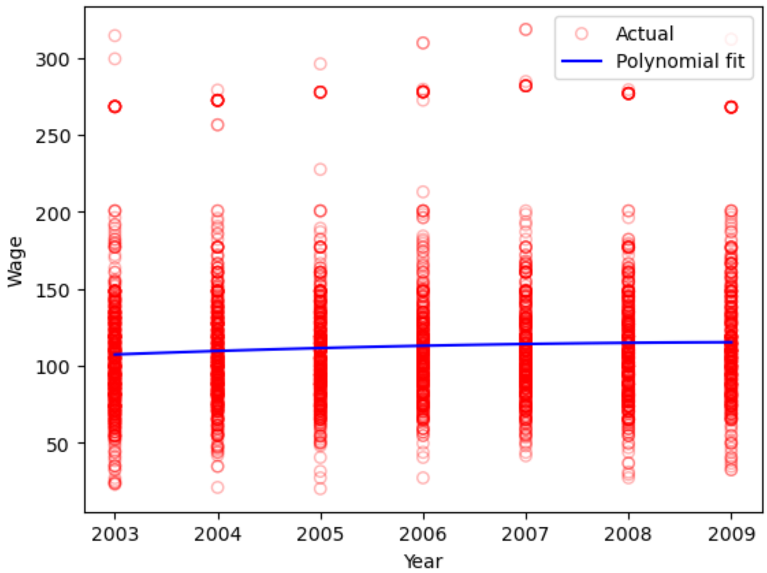

Figure 6: Actual data and polynomial fit for Wage vs Year.

- For Wage vs Education, a $4th$ order polynomial is fit, with coefficients as follows (highest to lowest degree)

$$[2.96 \times 10^{-02}, 4.38 \times 10^{-01}, -3.22 \times 10^{+00}, 1.78 \times 10^{+01}, 6.90 \times 10^{+}01]$$

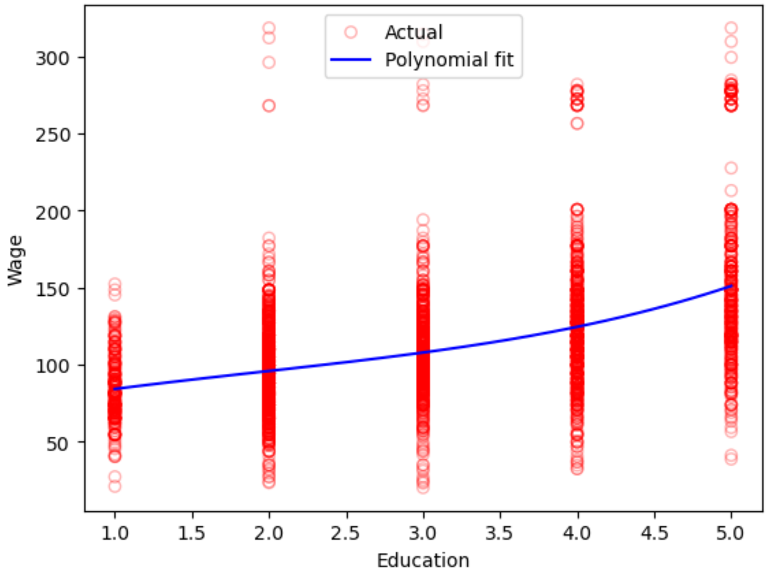

Figure 7: Actual data and polynomial fit for Wage vs Education.

- We tried to fit polynomials of various degrees for each of the 3 features, i.e. age, year and education. As seen from the obtained polynomial coefficients, we can get an idea of upto what degree of polynomial would fit best the data.
- In the case of Wage vs Age, we tried plotting for 6th degree polynomial, but the coefficients of x^6, x^5, x^4 are really small. Hence, we could say that a cubic polynomial would have been enough for the fit.
- Similarly for Year and Education, the polynomial coefficients are insignificant for x^3 and x^5 onwards respectively. Hence, a biquadratic polynomial would fit Wage vs Education and a quadratic polynomial fits Wage vs Year.
- Even though we are able to fit the data with some regression, we can see from the plots that the results are not good. This is because Wage depends on multiple features and only one feature is not enough to completely specify the Wage with good accuracy. Hence, we need to consider all the features together and go for some other regression technique.

Please refer the IPython notebook for the detailed results.

■■■