MACM 316 – Computing Assignment 6

- Read the *Guidelines for Assignments* first.
- Submit a one-page PDF report to Canvas and upload you Matlab scripts (as m-files). Do not use any other file formats.
- Keep in mind that Canvas discussions are open forums.
- You must acknowledge any collaborations/assistance from colleagues, TAs, instructors etc.

Machine learning

In this assignment, you will be use least-squares fitting to teach your computer to distinguish between red and blue points in 2D. This is an example of linear discriminant analysis.

First, download the file *dataset.mat*. This contains two tall matrices, *training_set* and *test_set*. Your goal is to train your computer using the training set and then use the test set to see how well it has learned.

Description of the dataset

Each dataset has three columns. The training set has 2000 rows (i.e. 2000 red and blue points) and the test set has 400 points. Let

```
X = training_set(:,[1 2]); y = training_set(:,3);
```

Each row of $\boldsymbol{X}$ is the (x, y) coordinates of a point in 2D. The corresponding entry in $\boldsymbol{y}$ is either 0 or 1, with 1 representing the colour blue and 0 representing the colour red.

Use Matlab's *load* command to import the datasets. Next, assemble $\boldsymbol{X}$ and $\boldsymbol{y}$ as above and use the *plot* command to reproduce the following plot:

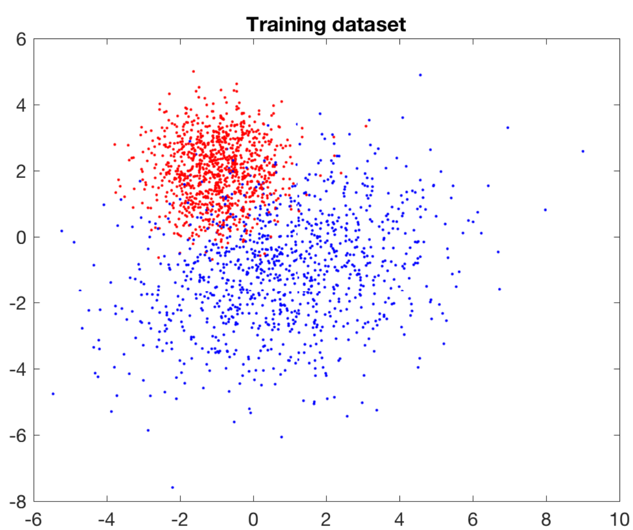

Figure 1: training dataset

Training phase

Define the matrix $\boldsymbol{A}$ as follows:

```
A = [ ones(2000,1) X ];
```

Note that $\boldsymbol{A}$ is a 2000×3 matrix and $\boldsymbol{y}$ is a 2000×1 vector. The objective of the training to find a vector $\boldsymbol{\beta}$ such that $\boldsymbol{A\beta} \approx \boldsymbol{y}$. Using least-squares fitting (implemented via backslash), compute such a vector, and call it $\hat{\boldsymbol{\beta}}$. Use this to compute the RSS (Residual Sum of Squares) value

$$\text{RSS} = \left\| \boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{\beta}} \right\|^2.$$

Record this value in your report. This gives an idea of how good $\hat{\boldsymbol{\beta}}$ is at predicting a value in $\boldsymbol{y}$ given the corresponding row in $\boldsymbol{A}$.

Next, in Figure 1 add a black line given by the following equation

$$\hat{\beta}_1 + \hat{\beta}_2 x_1 + \hat{\beta}_3 x_2 = 1/2. \tag{1}$$

What is this line is saying about the red and blue points? Explain your answer, and include both the figure and your explanation in the report.

Testing phase

It is now time to see how well the computer has learned. Let

```
z = test_set(:,3);
B = [ ones(400,1) test_set(:,[1 2]) ];
```

Note that $\boldsymbol{z}$ is a 400×1 vector and $\boldsymbol{B}$ is a 400×3 matrix similar to $\boldsymbol{A}$. Write a few lines of code to compute the vector $\boldsymbol{v} = \boldsymbol{B}\hat{\boldsymbol{\beta}}$ and the vector $\boldsymbol{z}$ defined by

$$\hat{z}_j = \begin{cases} +1 & \text{if } v_j \geq 1/2 \\ 0 & \text{if } v_j < 1/2 \end{cases}.$$

Compute the error of the prediction as follows:

$$\text{Err} = \frac{1}{400} \sum_{j=1}^{400} |z_j - \hat{z}_j|.$$

Record this value in your report, and discuss. Also, generate a plot of the test dataset where the points are labelled according to the prediction $\hat{\boldsymbol{z}}$ along with the discriminant line defined in equation (1) above.

Bonus! (2 additional marks)

You will hopefully have seen that linear discriminant analysis can do quite well for training a computer to distinguish certain datasets. Indeed, it is a powerful technique. However, it does not work well for all datasets. Devise a dataset for which you expect it would not work so well, and run your code on this dataset to demonstrate this observation.