Lab 1: Bayesian Decision Theory

**Objective**

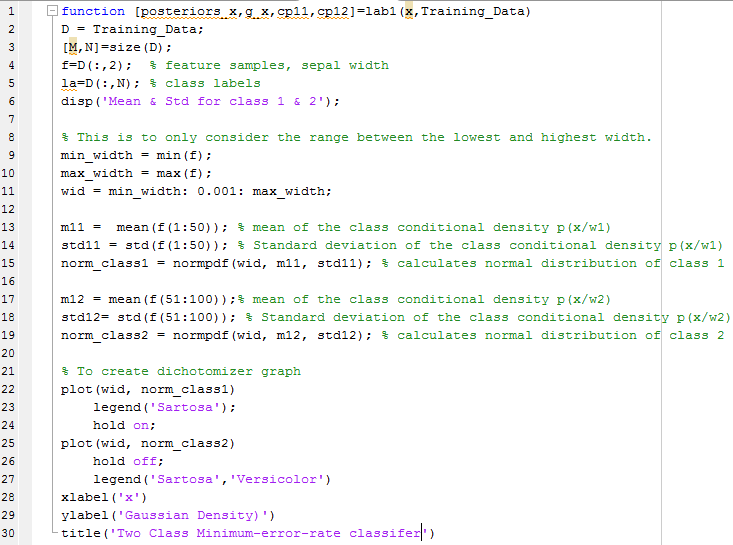
*https://www.byclb.com/TR/Tutorials/neural_networks/Ch_4_dosyalar/image002.gif*Main objective of this lab was to use Bayesian Decision rule to perform simple classifications. This theory is used when probability structure underlying the categories is well known. In this lab the following categories were used: Iris Setosa and Iris Versicolour. Bayes formula can be expressed as:

This formula is used to find Posterior probability of class (wj) given the feature (x).

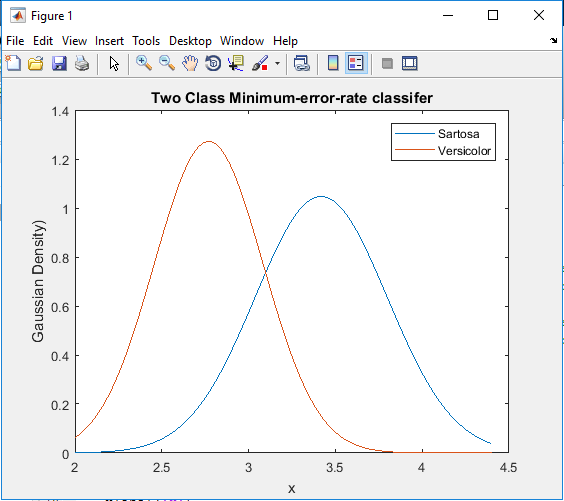
**Discussion**

1. Using a single discriminant function g(x2), design a 2 class minimum-error-rate classifier (dichotomizer) from the given data, to classify IRIS samples into either Iris Setosa or Iris Versicolour, according to the feature: *sepal width.*

* The sepal width data was extracted from the training set by extracting column 2 of the data. There were 100 entries within column 2, where the entries was evenly divided into two class. The first 50 rows pertained to Setosa while the last 50 rows pertained to Versicolour.
* To create a dichotomizer, the Normal distribution was computed and plotted for each class. This was achieved by calculating the mean and standard deviation for the first 50 rows (class Setosa), and last 50 rows (class Versicolor). Upon calculating the mean and standard deviations, the normal distributions were calculated and plotted.
* The following snippet function demonstrates the blocks of code that:
  + Extracted the appropriate columns for the classes of interest.
  + Calculated the mean and standard deviation for each class.
  + Plotted the normal distributions of each class.

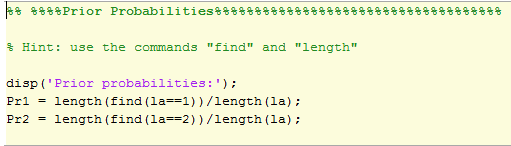


* The following normal distribution graph was generated as a result of the previous code snippet:

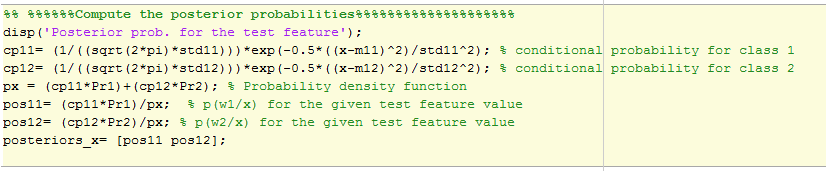


1. Using the shell programlab1.m, write a program that will take an individual sample value as the input and will return the posterior probabilities and the value of g(x2)

* To calculate the posterior probabilities, the prior probabilities were determined first. This was computed by the following code snippet:

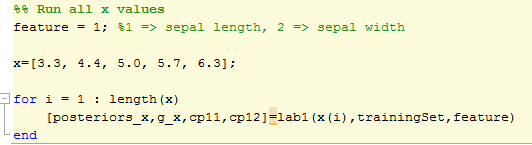


* The posterior probabilities were then computed using the following code snippet:



* The values of the returned posterior probabilities are: [**0.7657 0.2343]**
* The value of the returned g(x2) is: **0.5314**

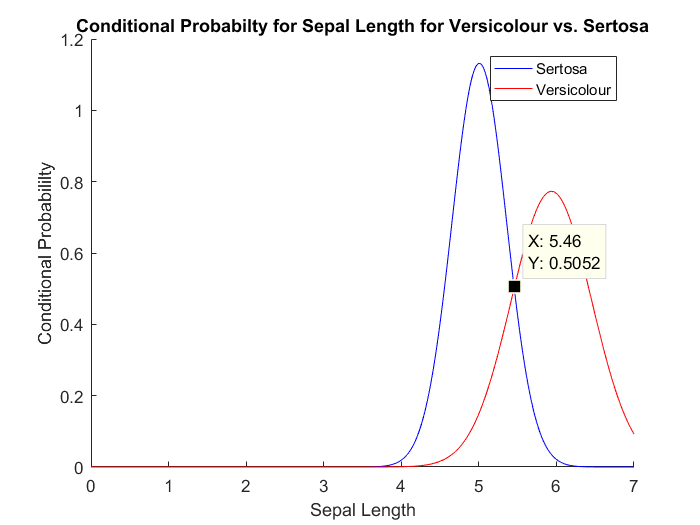
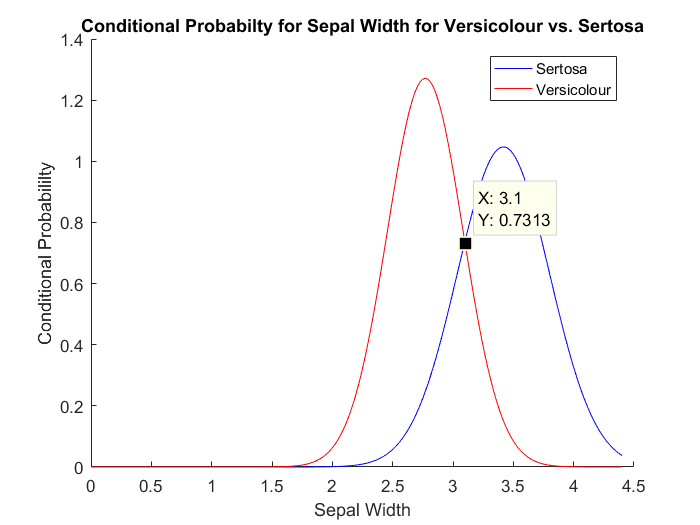
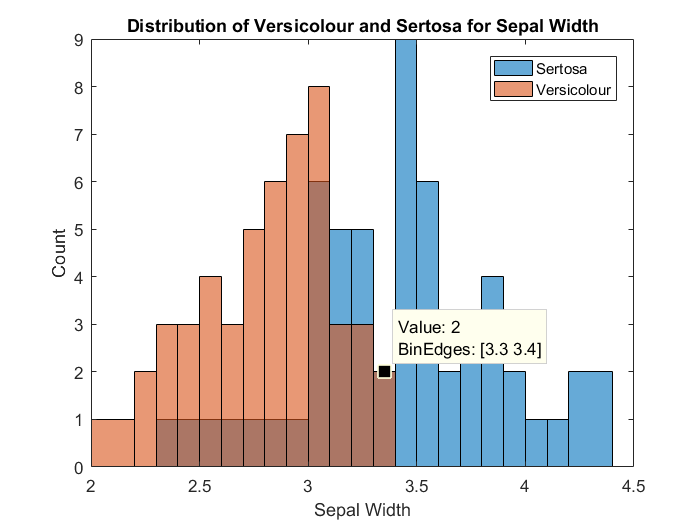
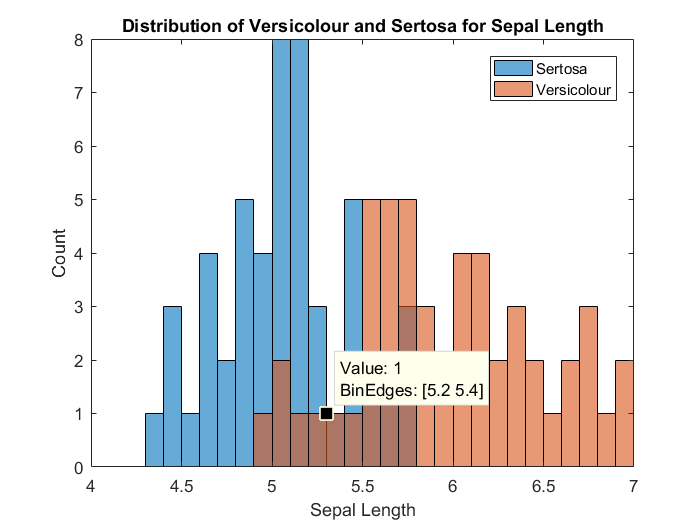
3. Identify the class labels for the feature values using your program, and indicate their respective posterior probabilities and discriminant function values: x1= [3.3, 4.4, 5.0, 5.7, 6.3]



The code above was used to find the posterior probabilities and discriminant function values. Class labels, posterior probabilities and discriminant function values are given in the following table using feature sepal width (x2):

|  |  |  |  |
| --- | --- | --- | --- |
| Values | Test Feature (x2) Results | Posterior Probabilities | Discriminant Function value |
| 3.3 | Sertosa | 0.7657  0.2343 | 0.5314 |
| 4.4 | Sertosa | 1.0000  0.0000 | 1 |
| 5.0 | Sertosa | 1.0000  0.0000 | 1 |
| 5.7 | Sertosa | 1.0000  0.0000 | 1 |
| 6.3 | Sertosa | 1.0000  0.0000 | 1 |

4. Arrive at an optimal threshold (Th1) that separates classes w1 and w2 (theoretically or experimentally). Justify your result.

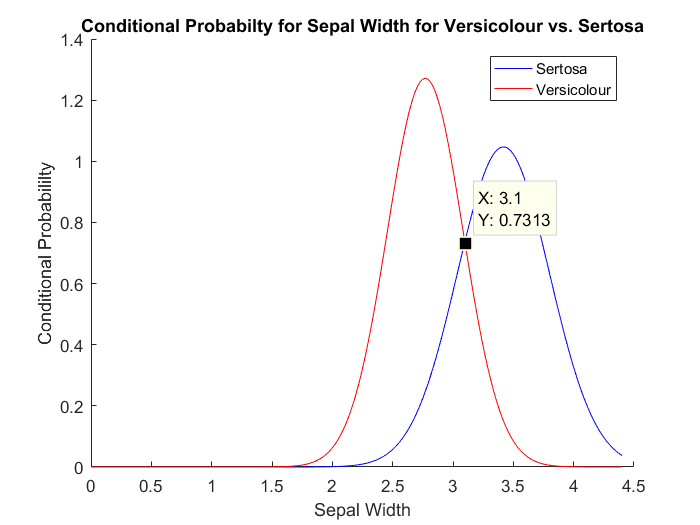


Since then, the equation for minimal error optimum threshold becomes

Instead of solving the mathematical equation, it was easier to graph the conditional probabilities against each other and find the P.O.I. For Sepal Width the optimum threshold was found to be x=3.1 and for Sepal Length, the optimum threshold was found to be x=5.4. These thresholds can also be confirmed on the histogram distribution graphs.

5. Suggest how Th1 would be a effected if a higher penalty is associated with classifying class w2 as class w1 – show with experiment.

Essentially, we would have to shift g(x) to the left to avoid the high penalty associated with classifying versicolor as sertosa if we were to use sepal width as our feature (see below).



Moving g(x) to the left from it’s previous location of x=3.1 (depending on the cost), will compensate for the higher penalty of misclassification but will also misclassify more sertosas as versicolor. This is the trade off with higher penalties.

6. Adjust your program to accept Sepal Length as the discriminating feature g(x1). Suggest which of the two features (x1,x2) might be a better choice for separating the two classes w1 and w2. Justify.

The code above in question 3 was used to find the posterior probabilities and discriminant function values using feature sepal length (x1).

|  |  |  |  |
| --- | --- | --- | --- |
| Values | Test Feature (x1) Results | Posterior Probabilities | Discriminant Function value |
| 3.3 | Satosa | 0.8457  0.1533 | 0.6935 |
| 4.4 | Satosa | 0.9655  0.0345 | 0.9310 |
| 5.0 | Satosa | 0.8834  0.1166 | 0.7669 |
| 5.7 | Versicolor | 0.1897  0.8103 | -0.6207 |
| 6.3 | Versicolor | 0.0022  0.9978 | -0.9956 |

Both features work well as the discriminating features separately but not together as one can see in the figure below, they do no intersect. It would be a better choice to use Sepal Width if there was a higher penalty with classifying versicolour as sertosa as versicolour has a higher conditional probability than sertosa. And the opposite would true for Sepal Length as sertosa has a higher conditional probability than versicolour.

