

|  |  |
| --- | --- |
| **Course Title:** | **Intelligent Systems** |
| **Course Number:** | **ELE 888** |
| **Semester/Year (e.g.F2016)** | **W2018** |

**Dr. Ling Guan**

**Instructor:**

|  |  |
| --- | --- |
| *Assignment/Lab Number:* | **Lab 2** |
| *Assignment/Lab Title:* | **Linear Discriminant Functions** |

|  |  |
| --- | --- |
| *Submission Date:* | **Feb-27-2018** |
| *Due Date:* | **Feb-27-2018** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student**  **LAST Name** | **Student**  **FIRST Name** | **Student Number** | **Section** | **Signature\*** |
| **RIAZ** | **NAUMAN** | **73729** | **3** | **NR** |
| **Rodov** | **Ariel** | **63601** | **1** | **AR** |
|  |  |  |  |  |

\*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:

**Objective:**

To find a linear combination of features which characterizes or separates two or more classes of objects or events, a linear discriminant function is used. This function is used in statistics, pattern recognition, and machine learning. These functions are linear in either the component of the feature vector x or in some given set of function of x.

A linear discriminant function g(x) can be written as:

|  |  |
| --- | --- |
| g(x) = wT x + w0 | (1) |

where w is the weight vector and w0 is the threshold. If there are two categories we decide w1 by observing the feature vector x if g(x) > 0 or w2 if g(x) < 0.

|  |  |
| --- | --- |
| A generalized discriminant function can be represented as: |  |
| g(x) = aT y | (2) |

where y is the augmented feature vector and a to be the augmented weight vector.

The objective of this lab was to classify the Iris datasets obtained from "archive.ics.uci.edu/ml/" using linear discriminant functions and to compute the weight vector using the training data samples. The perceptron criterion function Jp(a) and its respective gradient ∇Jp can be represented by:

|  |  |
| --- | --- |
| p (a) = ∑ y€Y (-aTy) | (3) |
| ∇Jp = ∑ y€Y (-y) | (4) |

**Observation:**

**Class A and Class B**

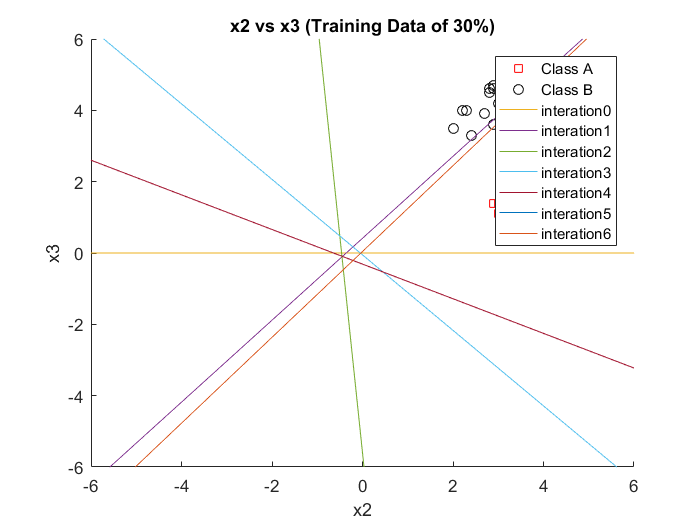
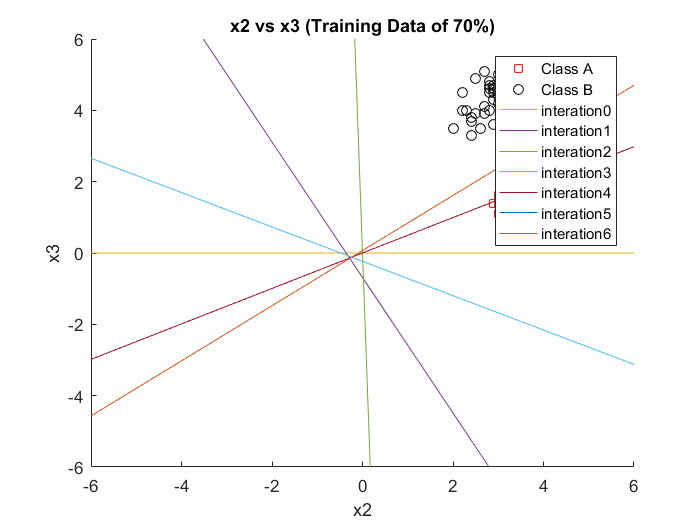


Figure 1: Data split with 30% training Figure 2: Data Split with 70% training

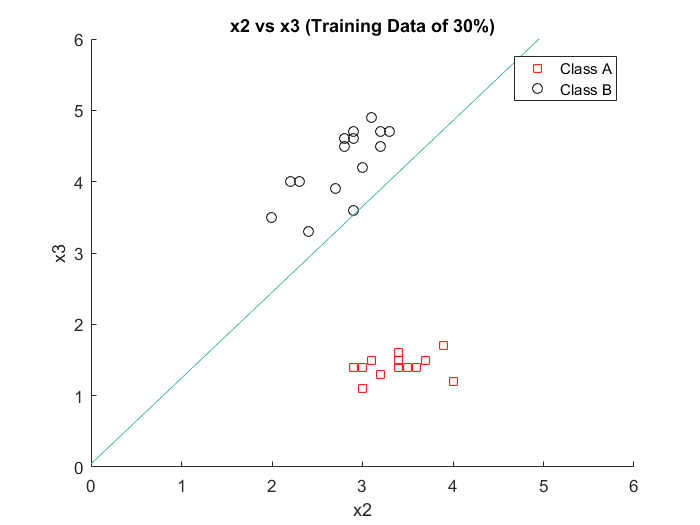


Figure 3: 30% training with decision boundary Figure 4: 70% training with decision boundary

By looking at the feature space plot (figures 1-4) of versicolor and setosa, we can identify that these two classes are linearly separable. By comparing these figures and changing the training set size provides more accuracy for the linear discriminate functions. This is because the more observations we take into account, the more accurate our solution can be.

**Numerical Values:**

Accuracy is:

|  |  |  |
| --- | --- | --- |
| Class A | Class B | Training Set Percentage |
| 100% | 97.22% | 30 |
| 100% | 100% | 70 |

Number of iterations: 6

Linear Discriminant Function (Vector):

[0.01 0.28 -0.23] at 30%.

[0.06 0.68 -0.89] at 70%.

**Class B and Class C**

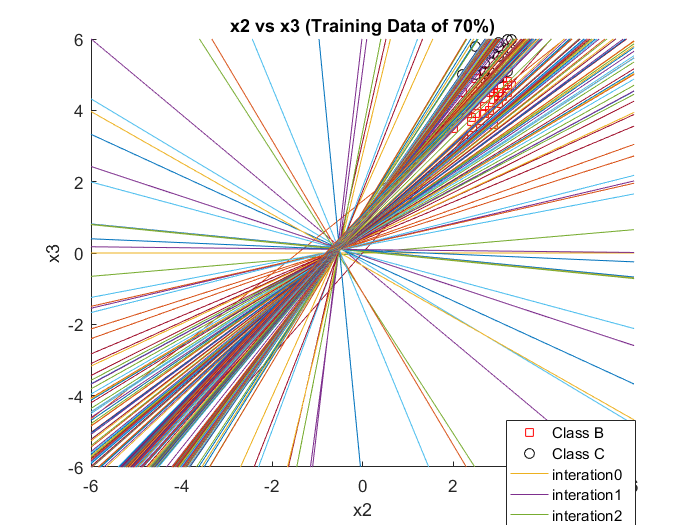
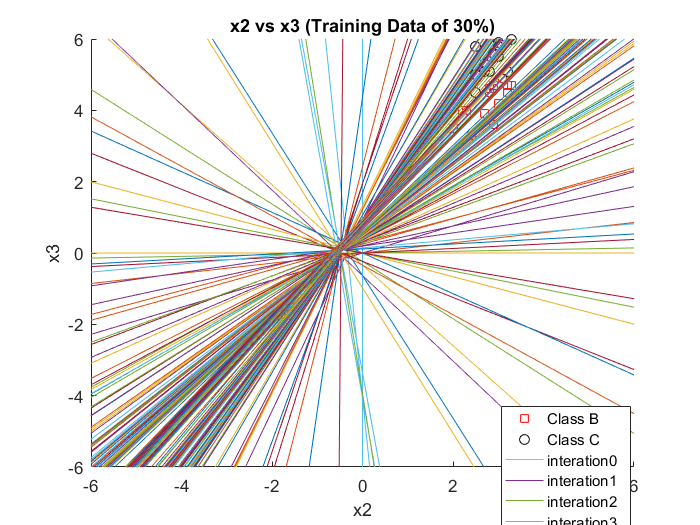


Figure 5: Data split with 30% trainingFigure 6: Data split with 70% training

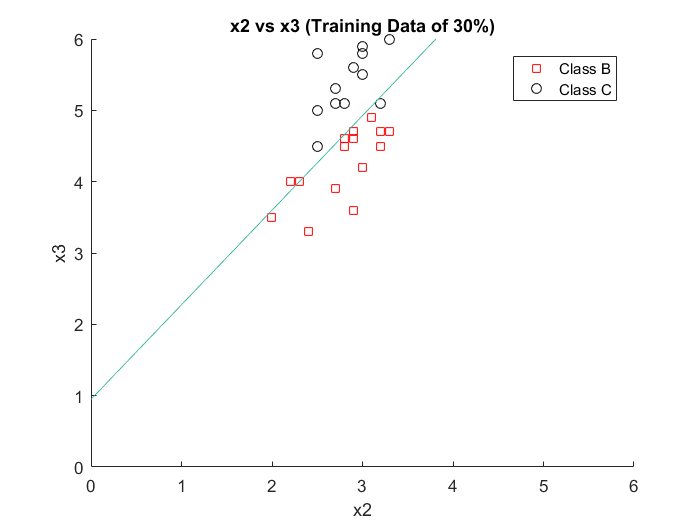
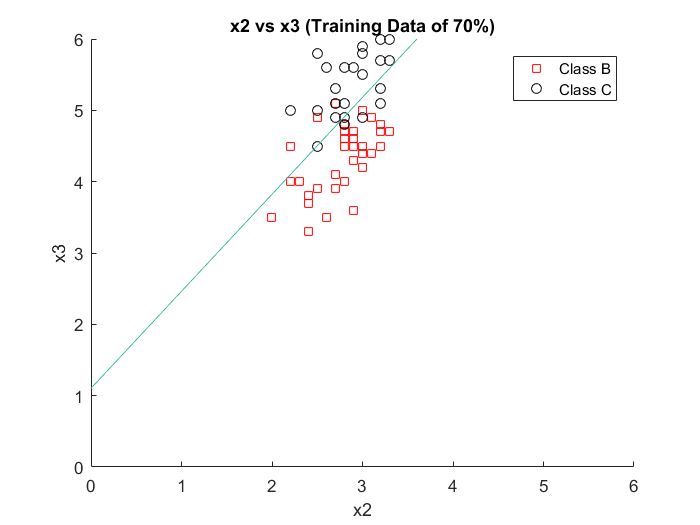


Figure 7: 30% training with decision boundary Figure 8: 70% training with decision boundary

For figures 5-8, we compare versicolor and verginica. Versicolor and verginica are not linearly separable. This shows us that increasing the training set does not improve the accuracy of the linear discriminant function. Also, vesicolor and verginica are not separable and this is why it took 300 iteration for the linear discriminant vector to be identified. This is because the perceptron criterion does not stop until all sample are correctly classified by a linear Function which is very hard to do practically.

**Numerical Values:**

Accuracy:

|  |  |  |
| --- | --- | --- |
| Class B | Class C | Training Set Percentage |
| 77.14% | 91.43% | 30 |
| 93.33% | 67% | 70 |

Number of iterations: 300

Linear Discriminant Function (Vector):

[2.30 3.22 -2.44] at 30%.

[5.62 6.92 -5.09] at 70%.

**Discussion:**

Following initial values were used for the figures above:

Learning rate: 0.01

Threshold: 0

Initial Weight vector: [0; 0; 1]

Maximum Iterations: 300

When the percentage of the training data was increased from 30% to 70%, the decision boundary for testing was much more distinct between the two classes. A larger will be more useful as the program will have more material to learn from. This is the reason why better decision boundary was seen for testing when training data was 70% and not 30%.

Accuracy can be affected by the learning rate. When the learning rate is small, training speed is slower and iteration increases. This can be seen in figures 7 and 8. On the other hand, when the learning rate is large, training speed is faster and iteration decreases. This can be seen in figures 3 and 4. Weight vector is another factor that determines the orientation of decision hyper plane. It can be seen in tables 1 and 2 how accuracy changes when training set is changed, the accuracy is affected for two different learning rate values of 0.006 and 0.03.

|  |  |  |
| --- | --- | --- |
| **Dataset**  **(Training/Testing)** | **Accuracy** | **Weight Vector a** |
| AB (30/70) | 91.4% |  |
| AB (70/30) | 100% |  |

Table 1: Results for datasets run with learning rate η(k) of 0.006

|  |  |  |
| --- | --- | --- |
| **Dataset**  **(Training/Testing)** | **Accuracy** | **Weight Vector a** |
| AB (30/70) | 94.3% |  |
| AB (70/30) | 100% |  |

Table 2:Results for datasets run with learning rate η(k) of 0.03

Just like other factors, the learning rates can have a significant effect on accuracy. A small learning rate will have slow training speed and more iteration. Opposite will happen when learning rate is large. To minimize the criterion function, the linear discriminate function is used. Higher training sample also increased the accuracy as it can be seen from the figures above. Lower sample made the program inaccurate. Overfitting could be a cause of this.

**Conclusion**

Major changes were observed when different sizes of data were used for testing and training. Higher number of training data led to more accurate criterion function for the set and the usage of more testing data made it possible to determine the accuracy of the criterion function. Based on the observations, the accuracy can be manipulated by the size of the training set. Larger training sets give different accuracies than small training sets. Furthermore, the perceptron discriminant function can also affect the learning rate. As the learning rate increases, it took less iteration for the results to converge. In conclusion, the lab was a success in classifying the Iris datasets using linear discriminant functions.