Morgan State University
Department of Electrical & Computer Engineering

EEGR 415: Java Programing Applications

Project 1

Objective:

The ACME Fruit Company is in need of an automated system for detecting harvested fruit that is too ripe and may rot during shipping. After some analysis, two processes have proven to contribute the detection of over ripe fruit. The first (Process 1) is a chemical analysis of the gas produced by the fruit while processing. The second (Process 2) is a histogram and hue analysis of the color of the fruit. Given training and test data from these two processes, develop a Perceptron algorithm to classify the features of a given fruit as “good” or “too ripe”.

Step 1: Read and Display Data

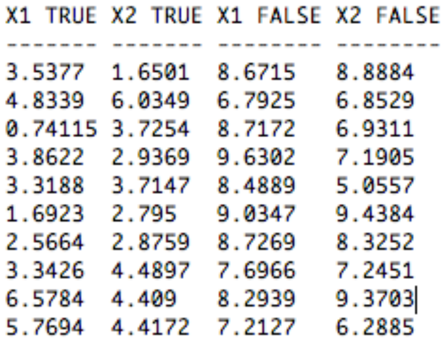
The test data to train your classifier is stored in the file ‘data.txt’. This is a text file with the following arrangement of data:

Rows	Description
1 - 10	Process 1 data (X1) for good fruit (True Class)
11 - 20	Process 2 data (X2) for good fruit (True Class)
21 - 30	Process 1 data (X1) for bad fruit (False Class)
31 – 40	Process 2 data (X2) for bad fruit (False Class)

This information will be stored in two double dimensioned arrays:

tcData – True Class data
fcData – False Class data

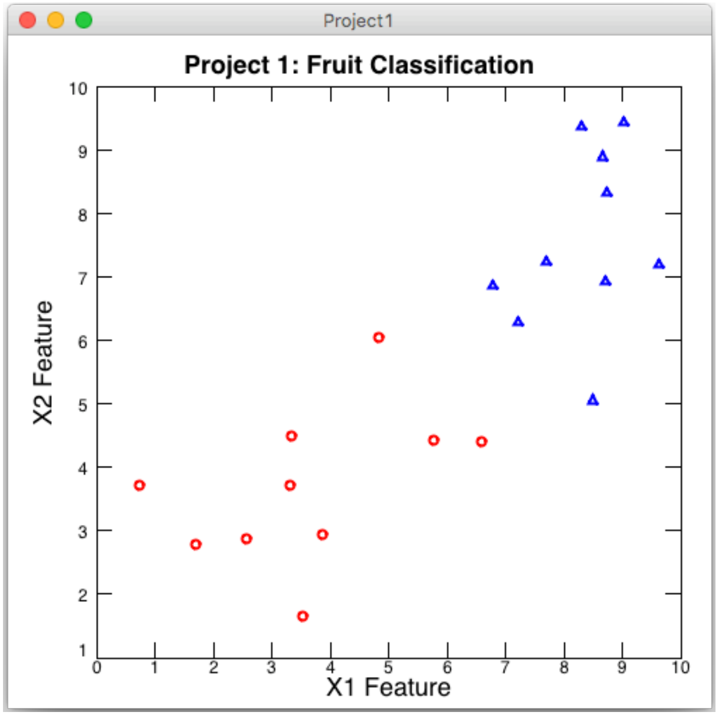
Instead of setting the first dimension of these arrays to 10 (for the number of samples), create a constant called MAX_SAMPLES, set that to 10, and then set the first dimension of the two arrays to MAX_SAMPLES, and the second to 2 (Process 1 data and Process 2 data). Loop through the contents of the file and store the data in the two arrays. Once complete, display the content of the two arrays as follows:



```
X1 TRUE X2 TRUE X1 FALSE X2 FALSE
------- ------- -------- --------
3.5377  1.6501  8.6715   8.8884
4.8339  6.0349  6.7925   6.8529
0.74115 3.7254  8.7172   6.9311
3.8622  2.9369  9.6302   7.1905
3.3188  3.7147  8.4889   5.0557
1.6923  2.795   9.0347   9.4384
2.5664  2.8759  8.7269   8.3252
3.3426  4.4897  7.6966   7.2451
6.5784  4.409   8.2939   9.3703
5.7694  4.4172  7.2127   6.2885
```

Step 2: Plot the Data

Plot the training data using JPlot2D as shown on the next page. In order to do this, you can create new single dimensioned arrays to store the X1 and X2 points for both classes separately. Plot X1 vs X2 for the true class using red circles, and X1 vs X2 for the false class using blue triangles. Refer to Lecture 4 to review the procedure for plotting.



*Hint: The graph properties, such as the marker colors and line style apply to the last points added. To add a new set of points, use the method **pl.addCurve(x2,y2)** (this is not in the slides).*

Step 3: Train Classifier

Implement the following Matlab script to train the weights of your classifier:

```
w = [0 0 0];
I = 100;
for i = 1:I
   for n=1:length(c1x1)
       % Weight adjusting for class 1
       if ( w * [1 c1x1(n) c1x2(n)]') < 0
           w = w + [1 c1x1(n) c1x2(n)];
       end

       % Weight adjusting for class 2
       if ( w * [1 c2x1(n) c2x2(n)]') >= 0
           w = w - [1 c2x1(n) c2x2(n)];
       end
   end
end
```

Array of 3 elements

Class 1, X1 data

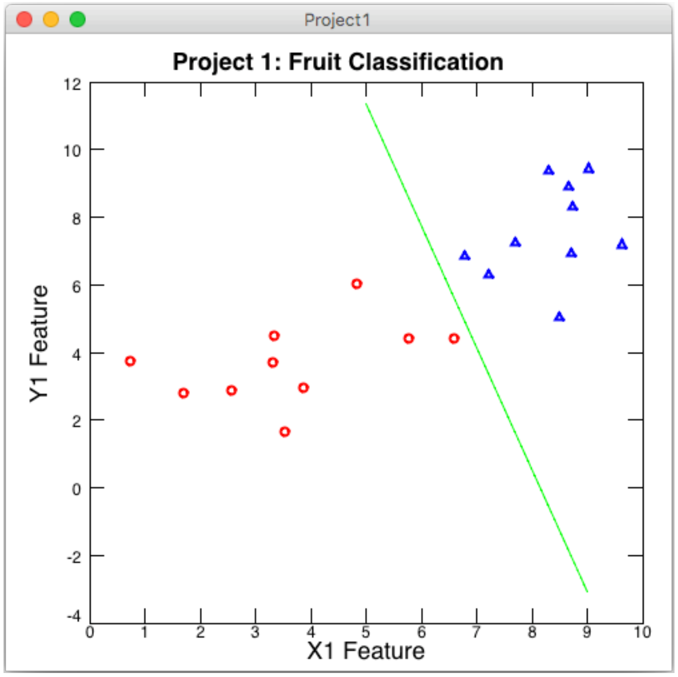
Loop from 1 to 100

Matrix multiplication

Class 1, X2 data

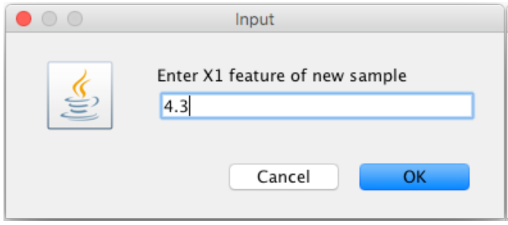
Note that in Matlab, vectors (arrays) start at position 1. Once the algorithm above completes, the **w** array should contain non-zero numbers. Extra Credit: Use the following formula to plot the resulting line, which shows the decision boundary of the classifier.

$$y = (w_0 + x(w_1))/-w_2$$



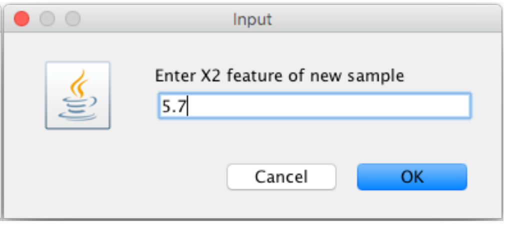
Step 4: Classifier a New Sample

Once you have confirmed that your classification boundary resembles the one shown above, you should be able to classify a new fruit as being 'good' or 'too ripe'. First, ask the user to enter the values for Process 1 and Process 2 for a fruit.



Enter X1 feature of new sample

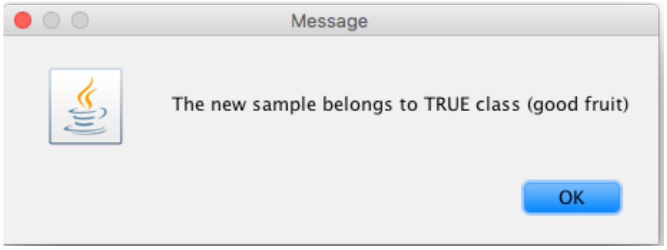
4.3



Enter X2 feature of new sample

5.7

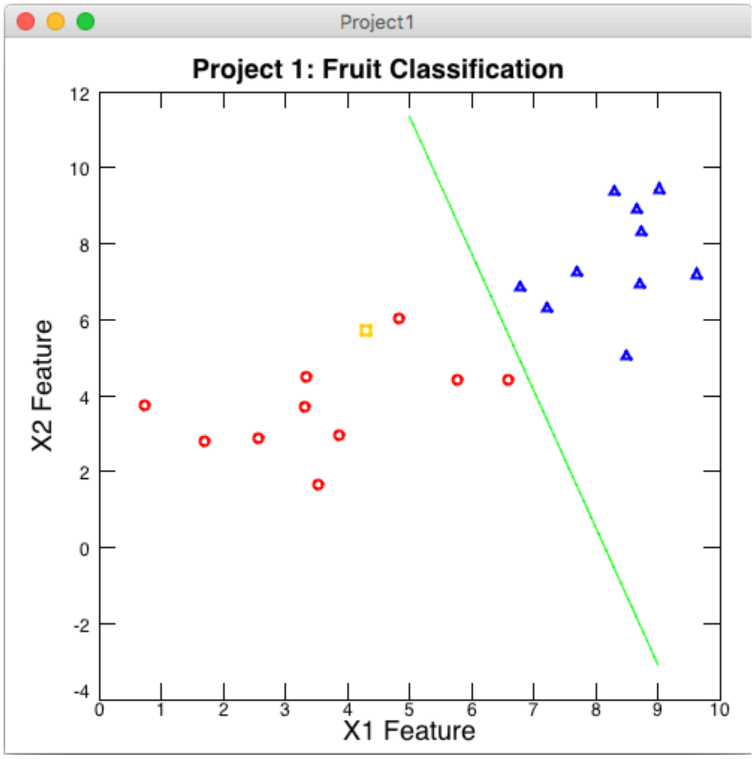
With the values provided, perform the dot product of the weights you obtained after training and the vector containing the new feature points for process 1 and 2 (similar to what was done during the training phase). If the result is greater than or equal to 0, then display the message below:



The new sample belongs to TRUE class (good fruit)

If the result is less than 0, display the message 'The new sample belongs to FALSE class (too ripe)'.

Then display the position on the plot where the new sample falls as an orange square.



Extra Credit (+20):

Display the classification boundary (green line) on the graph

Grading:

1. Use all appropriate variables (tcData,fcData) [10]
2. Able to read and store training data into tcData/fcData [10]
3. Able to display the training data in tabular form [10]
4. Able to produce a plot showing both true and false class [20]
5. Able to train the perceptron classifier correctly [20]
6. Able to display the new test point on plot [10]
7. Able to read new point using input message dialog boxes [10]
8. Able to correctly classifier a new point as belonging to true or false class [10]
9. Extra Credit: Able to display the classification boundary on the plot [+20]