**UNIVERSITY OF VICTORIA**

**Department of Electrical and Computer Engineering**

**ECE 403/503 Optimization for Machine Learning**

**LABORATORY REPORT**

Experiment No: 2

Title: Multi-Category Classification Using Binary Linear Classifiers

Date of Experiment: 11 June 2019

Report Submitted on: 4 June 2019

To: Jinlong Zhan - [jinlongzhan@uvic.ca](mailto:jinlongzhan@uvic.ca)

Laboratory Group No.: B01 - 6

Name(s):

Philippe Larocque - V00903595

Alexander Fung - V00904094

## Objectives

The objective of the lab is to classify three different iris plants using multi-category binary linear classification. [1]

## Introduction

In order to identify the individual flowers, a large data set D\_iris.mat of 5x150 elements containing the dimensions (lengths and widths) of each flower is used.

MATLAB software was used to train the machine by taking groups of 50 data points. A random 40 from each are selected for training and the remaining 10 from each are used for testing. The testing data is then classified into each category, and the error rate is calculated.

## Results

See MATLAB code in appedix A.

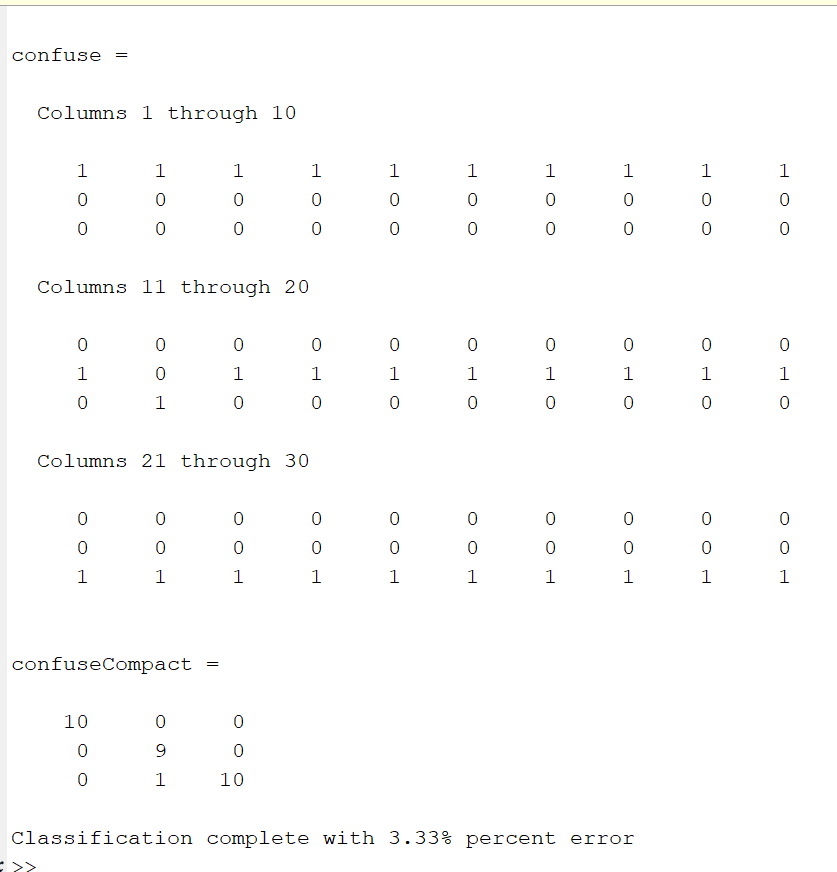


Figure 1 - Output results from code

## Discussion

The results of the MATLAB code produces the classifications of each flower. Row 1 corresponds to the program classifying it as Setosa, Row 2 corresponds to Versicolor, and Row 3 corresponds to Virginica. The first output, *confuse,* is a 3x30 matrix, and *confuseCompact* is a 3x3 summary matrix. The items on the diagonal correspond to the correctly classified flowers, and the anti-diagonal values represents the misclassification, and which flower it was misclassified as. The misclassified flower (versicolor) likely does not have similar characteristics to the other flowers of its type. It likely more closely resembles virginica, which it was predicted to be. Possibly with a larger training set, we may improve the accuracy of classification.

## Conclusion

The objective of this experiment was to classify three different iris plants using multi-category binary linear classification. Based on a testing set of 30 flowers, the confusion matrix was obtained, and an error of 3.33% was calculated, which is equal to one misclassification. We may be able to improve the accuracy with a larger training set.

## References

[1] Lu, Wu-Sheng. (May 2019). Experiment 1 - Laboratory Manual ECE 403/504 Optimization for Machine Learning. [Online]. Accessed May 2019.

<https://ece.uvic.ca/~wslu/403/403pass/Trans/LabManual-ECE403-503-2019.pdf>

## Appendix A - MATLAB code

%function call to generate each w-star and b-star

[ws1,bs1] = linear(Xtr1,Xtr2,Xtr3);

[ws2,bs2] = linear(Xtr2,Xtr1,Xtr3);

[ws3,bs3] = linear(Xtr3,Xtr1,Xtr2);

%group the testing data together

XteTotal = [Xte1 Xte2 Xte3];

%calculate the new y's and combine them

y1 = (ws1')\*XteTotal+bs1;

y2 = (ws2')\*XteTotal+bs2;

y3 = (ws3')\*XteTotal+bs3;

Y = [y1;y2;y3];

%identify the max of each column

[C,I]=max(Y);

%build the large confusion matrix

confuse = zeros(3,30);

for i=1:30

if I(i)==1

confuse(1,i)=confuse(1,i)+1;

elseif I(i)==2

confuse(2,i)=confuse(2,i)+1;

else

confuse (3,i)=confuse(3,i)+1;

end

end

%build the small confusion matrix

confuseCompact=zeros(3,3);

for i=1:3

confuseCompact(i,1)=sum(confuse(i,1:10));

confuseCompact(i,2)=sum(confuse(i,11:20));

confuseCompact(i,3)=sum(confuse(i,21:30));

end

%calculate the error

sumErr = confuseCompact;

for i=1:3

sumErr(i,i)=0;

end

%display everything to console

confuse

confuseCompact

percentErr = 100\* sum(sumErr,'all')/30;

%print error percentage results to screen

fprintf('Classification complete with %0.2f%% percent error\n',percentErr);

%function to output w-star and b-star

function [w\_Star, b\_Star] = linear(D1,D2,D3)

y = [ones(40,1); -ones(80,1)]; %initialize the y

p = D1; %set the positive

n = [D2 D3]; %set the negative

xHat = [[p n]' ones(120,1)]; %calculate x-hat

wHat = ((xHat'\*xHat)^-1)\*(xHat')\*y;%calculate w-hat from eq. E2.2 [1]

w\_Star = wHat(1:4,:); %calculate w-star

b\_Star = wHat(5,1); %calculate b-star

end