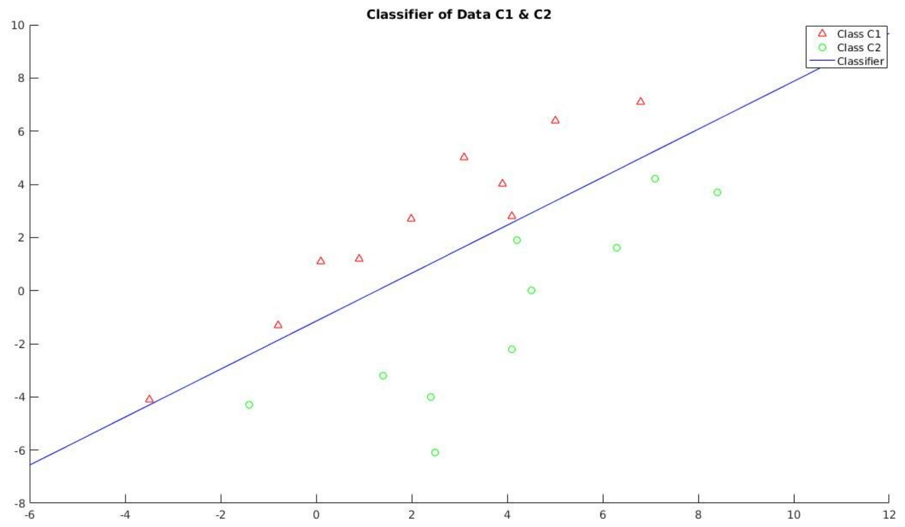
Aneel Balda(20162035)

Problem 1: Perceptron Learning

1

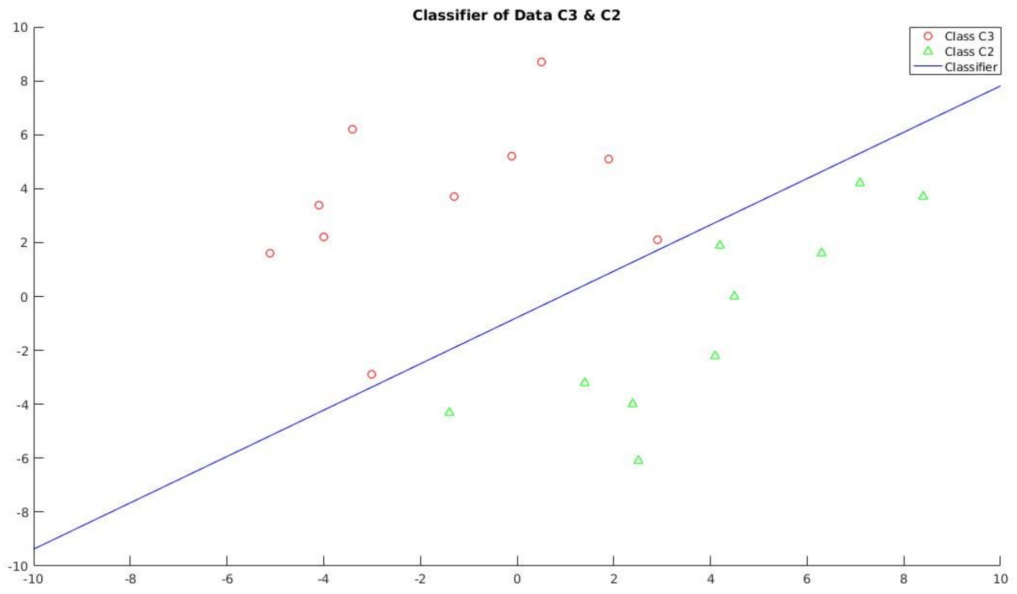
Initial is W=[0.0 0.0 0.0], the classifier found is W=[-0.1020 0.1130 0.1300] .
class C1 has been depicted with rad triangle, and that for class C2 is green dots. Classifier is shown in the following diagram.



The number of iteration required to reach this to this classifier is 9.

2:

Initial is W=[0.0 0.0 0.0], the classifier found is W=[-0.0055 0.0064 0.0050] . class C3 has been depicted with rad dot, and that for class C2 is green triangle. Classifier is shown in the following diagram.



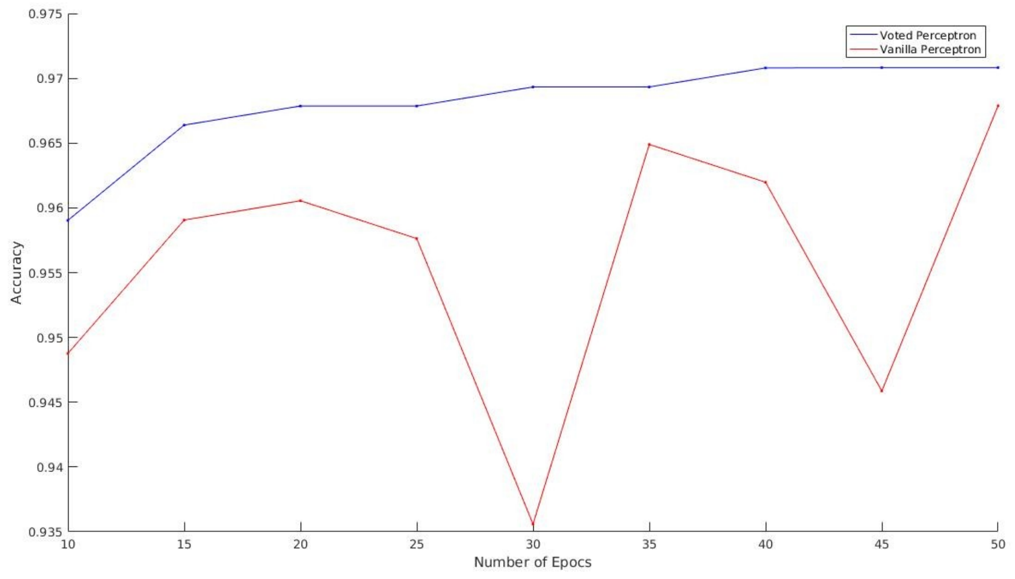
The number of iteration required to reach this to this classifier is 5.

3:

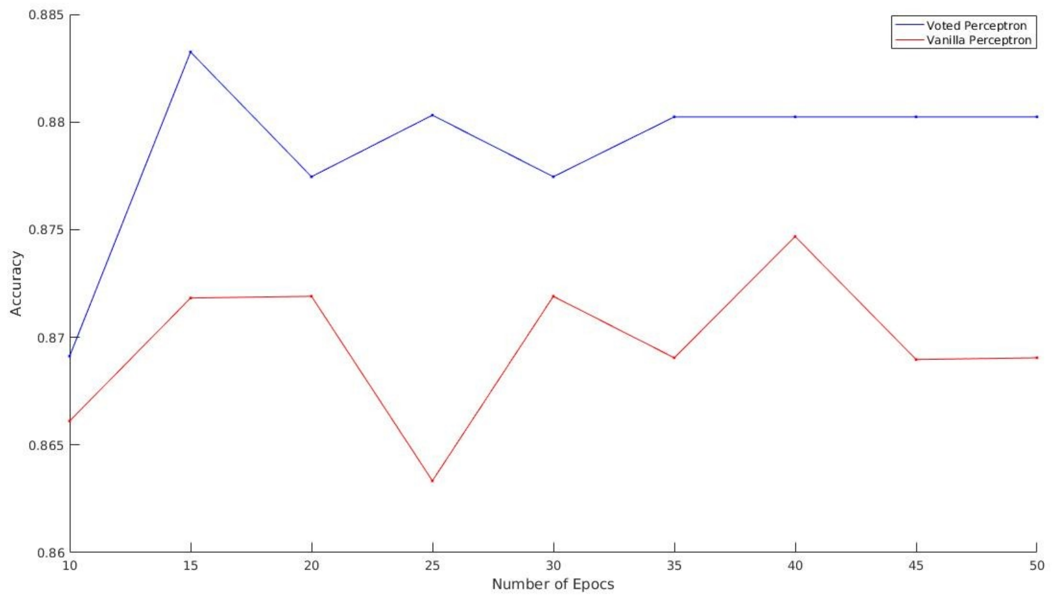
Convergence of perceptron depends on margin γ (separation between two classes). Let the perceptron converges after K steps, then, it can be proved that $k \propto 1/\gamma^2$. Since $k \propto 1/\gamma^2$, k will be smaller for larger value of γ. Thus, data of set 2 converges converges quickly.

Problem 2 : Voted Perceptron

<u>Dataset 1: Breast Cancer Dataset</u>



<u>Dataset 2: Ionosphere Dataset</u>

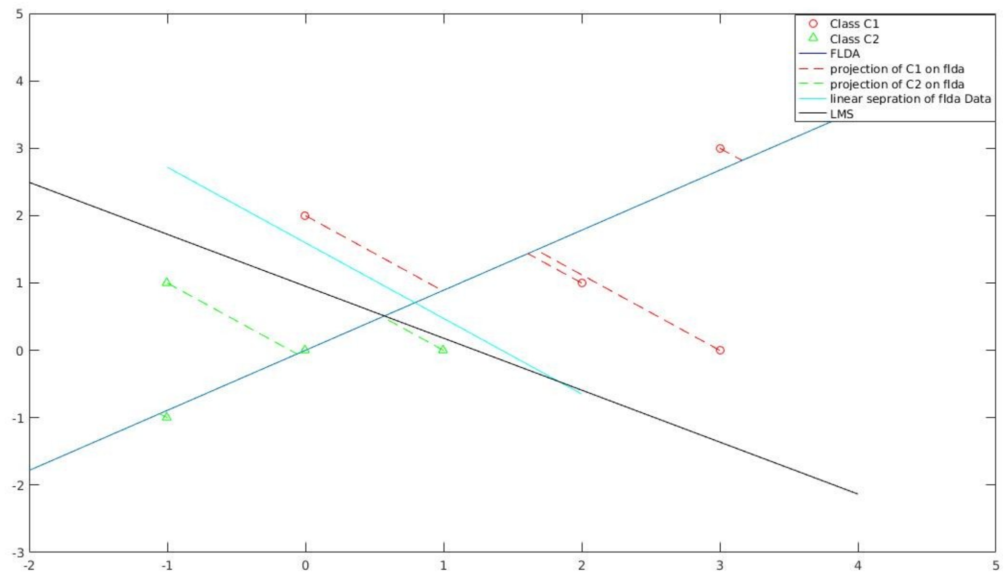


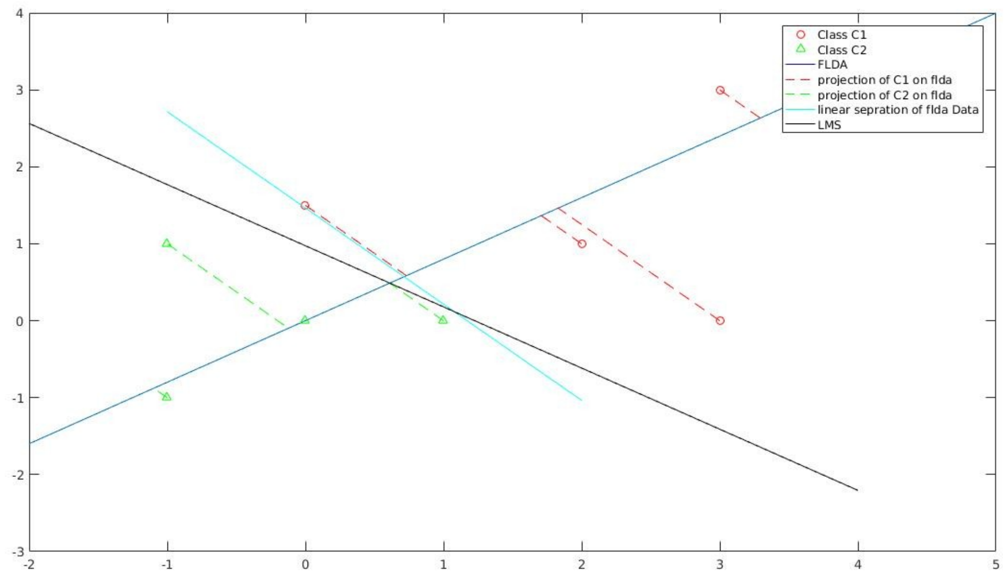
Performance of ***Voted algorithm*** is better when compared to the ***Vanilla algorithm*,**it is duly noted that the classification computational cost is very high. Also that there requires greater storage space. The voting Perceptron stores every single weight vector computed.
If I is the number of iteration to train weight vector and S is size of weight vector then space complexity id O(I*S) and in case of vanilla O(S).
Similarly, the Perceptron can make predictions in linear time in the size of the weight vector; the voting Perceptron only makes predictions in time linear in I∗ S which can be much larger.

Since in Voted Perceptron , the corresponding vote count has been taken under consideration , the classifier which classify more number of sample properly gets higher significance in deciding the final prediction. Thus, from epoch-accuracy (obtained using 10-fold cross validation) graph we see, for a set of data, the Voted Perceptron saturates after a certain value of epoch, where accuracy of Vanilla Perceptron fluctuates which does not guarantee better performance with the increase in number of epochs.

Problem 3: Least Square Approach

Dataset -1





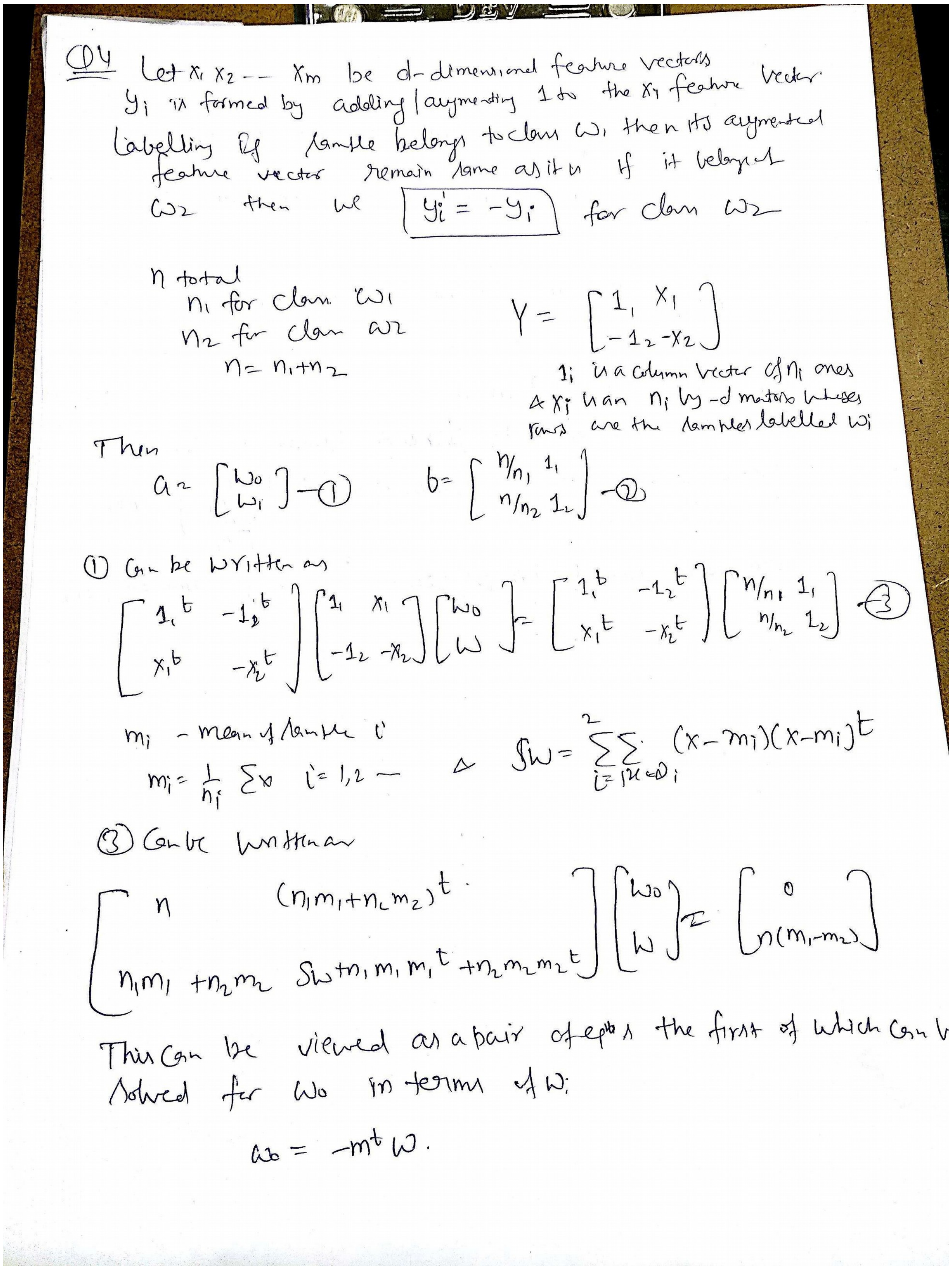
Dataset -2

Comparison:

<u>Fisher's Linear Discriminant</u> provides much clearer separation of data. Since it increases the inter-class scatter and hence projected points are guaranteed to be linearly separable. Upon finding the projected data, we applied Perceptron algorithm on projected data to find linear classifier – this separates the projected data.

<u>LMS</u> on the other hand tries to reduce the min square of the error which is kind of distance of the point from the classifier, so it might mis-classify the points lying very near to the classifier.

Problem 4:



Q4 Let $x_1\ x_2 - - x_m$ be d-dimensional feature vectors
y_i is formed by adding / augmenting 1 to the x_i feature vector.
Labelling if sample belongs to class ω_1 then its augmented feature vector remain same as it is. If it belongs to ω_2 then we $\boxed{y_i' = -y_i}$ for class ω_2

n total
n_1 for class ω_1
n_2 for class ω_2
$n = n_1 + n_2$

$$Y = \begin{bmatrix} 1_1 & X_1 \\ -1_2 & -X_2 \end{bmatrix}$$

1_i is a column vector of n_i ones & X_i is an n_i by $-d$ matrix whose rows are the samples labelled ω_i

Then

$$a = \begin{bmatrix} w_0 \\ w_i \end{bmatrix} - ① \qquad b = \begin{bmatrix} n/n_1\ 1_1 \\ n/n_2\ 1_2 \end{bmatrix} - ②$$

① Can be written as

$$\begin{bmatrix} 1_1^t & -1_2^t \\ X_1^t & -X_2^t \end{bmatrix} \begin{bmatrix} 1_1 & X_1 \\ -1_2 & -X_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} = \begin{bmatrix} 1_1^t & -1_2^t \\ X_1^t & -X_2^t \end{bmatrix} \begin{bmatrix} n/n_1\ 1_1 \\ n/n_2\ 1_2 \end{bmatrix} ③$$

m_i - mean of sample i

$$m_i = \frac{1}{n_i} \sum x \quad i = 1,2 - \qquad \& \quad S_W = \sum_{i=1}^{2} \sum_{x \in D_i} (x - m_i)(x - m_i)^t$$

③ Can be written as

$$\begin{bmatrix} n & (n_1 m_1 + n_2 m_2)^t \\ n_1 m_1 + n_2 m_2 & S_W + n_1 m_1 m_1^t + n_2 m_2 m_2^t \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ n(m_1 - m_2) \end{bmatrix}$$

This can be viewed as a pair of eq^ns the first of which can b
solved for w_0 in terms of w;

$$w_0 = -m^t w.$$

Substituting it in second equation

$$\Rightarrow \left[\frac{1}{n} S_W + \frac{n_1 n_2}{n^2} (m_1 - m_2)(m_1 - m_2)^t \right] w = m_1 - m_2$$

Since the vector $(m_1 - m_2)(m_1 - m_2)^t w$ is in the direction of $(m_1 - m_2)$, we can write

$$\frac{n_1 n_2}{n^2} (m_1 - m_2)(m_1 - m_2)^t w = (1 - \alpha)(m_1 - m_2)$$

α - scalar

$$\therefore \quad w = \alpha n . S_W^{-1} (m_1 - m_2)$$

This is identical to fisher's Linear Discri.

Hence if we take classes as label $\begin{bmatrix} \frac{n}{n_1} \\ -\frac{n}{n_2} \end{bmatrix}$, then

Classifier learnt through least square is same as FLD