EE559- Mathematical Pattern Recognition

Homework #4

Sourabh Tirodkar
3589406164

Results:

Problem 1:

Data: Synthetic 1

Error Percentage:

Error Rate for training data= 2
Error Rate for testing data= 2

Final Weights:

w1= -32.2943
w2= 29.7455

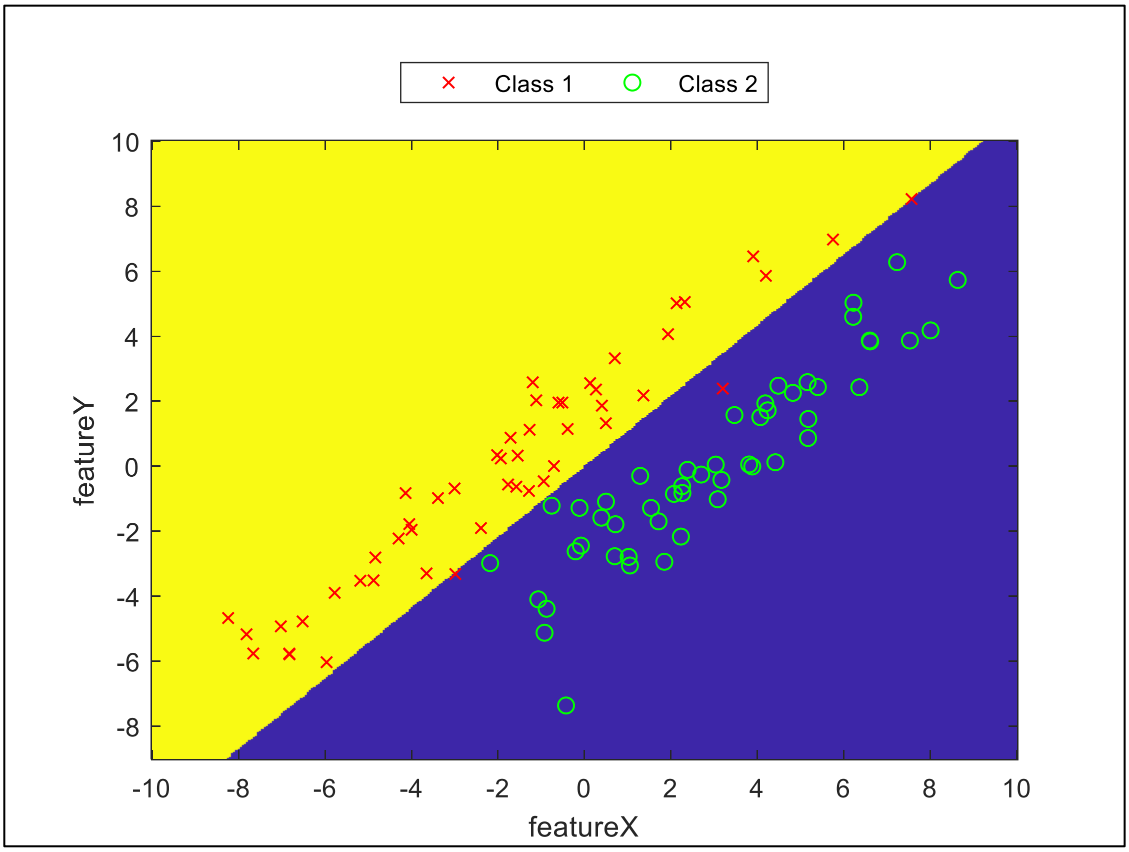

Fig. 1 Training Data (Synthetic 1) Plot

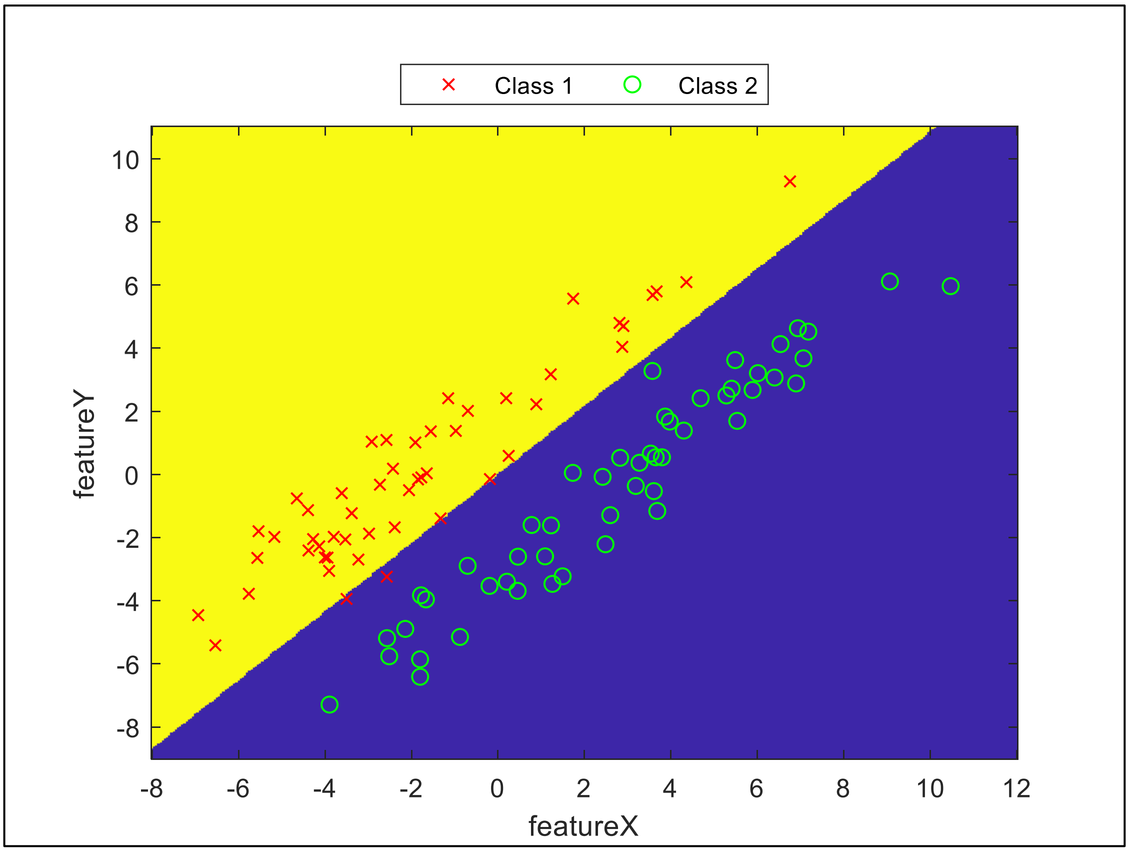

Fig. 2 Testing Data (Synthetic 1) Plot

Data: Synthetic 2

Error Percentage:

Error Rate for training data= 2

Error Rate for testing data= 3

Final Weights:

w1= -2.3046

w2= 17.4572

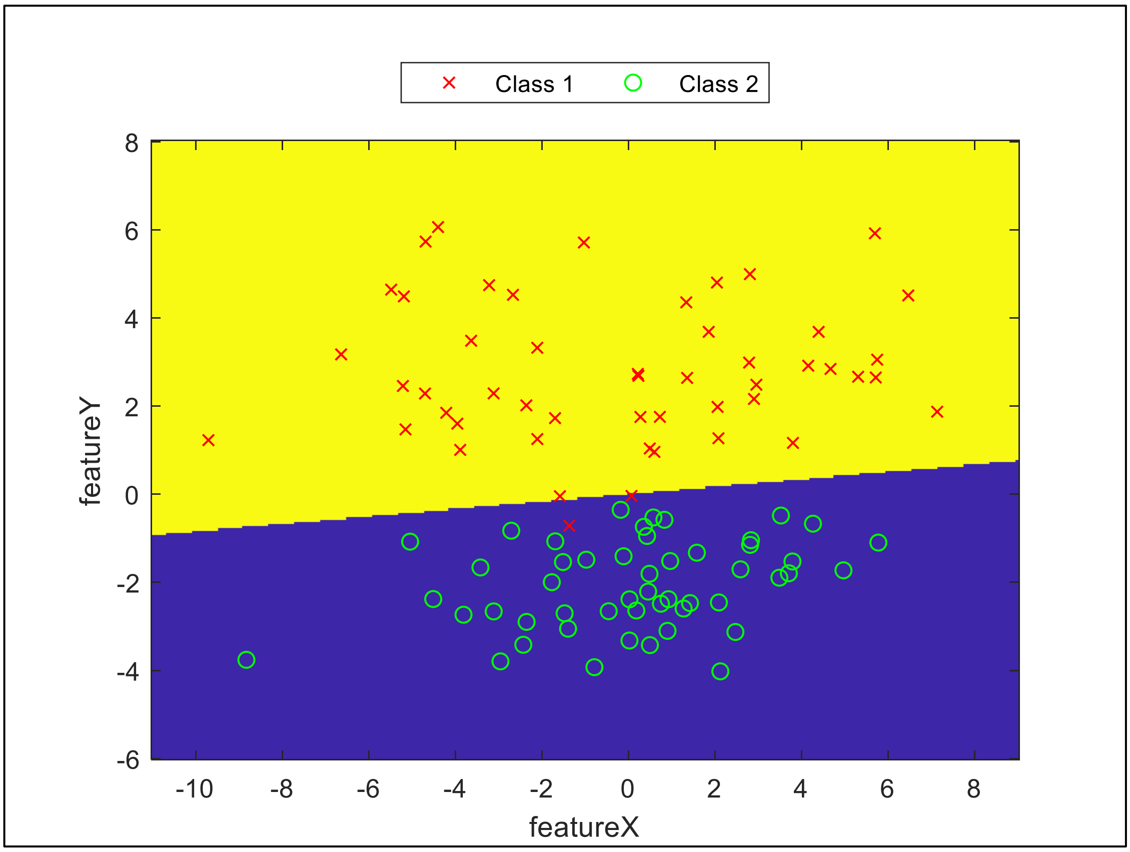

Fig. 3 Training Data (Synthetic 2) Plot

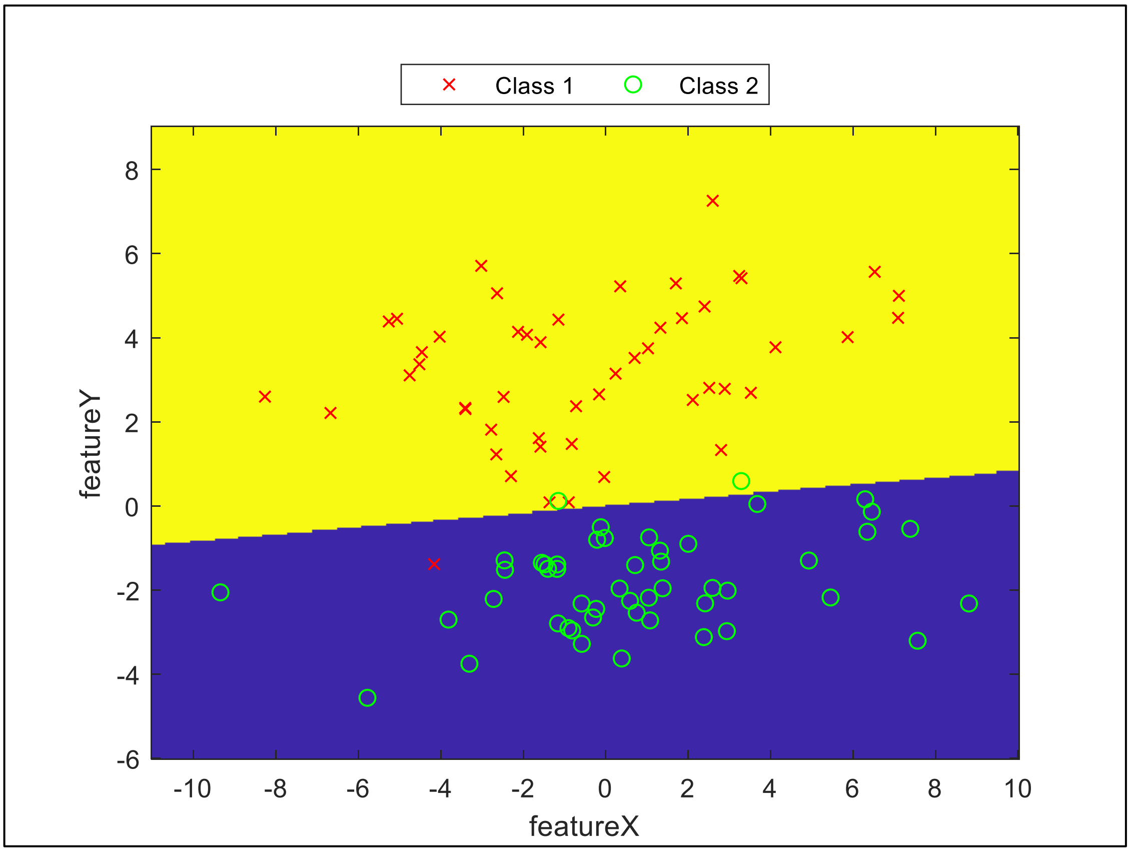

Fig. 4 Testing Data (Synthetic 2) Plot

<u>Data: Synthetic 3</u>

Error Percentage:

Error Rate for training data= 0

Error Rate for testing data= 1

Final Weights:

w1= -9.9596

w2= 8.2223

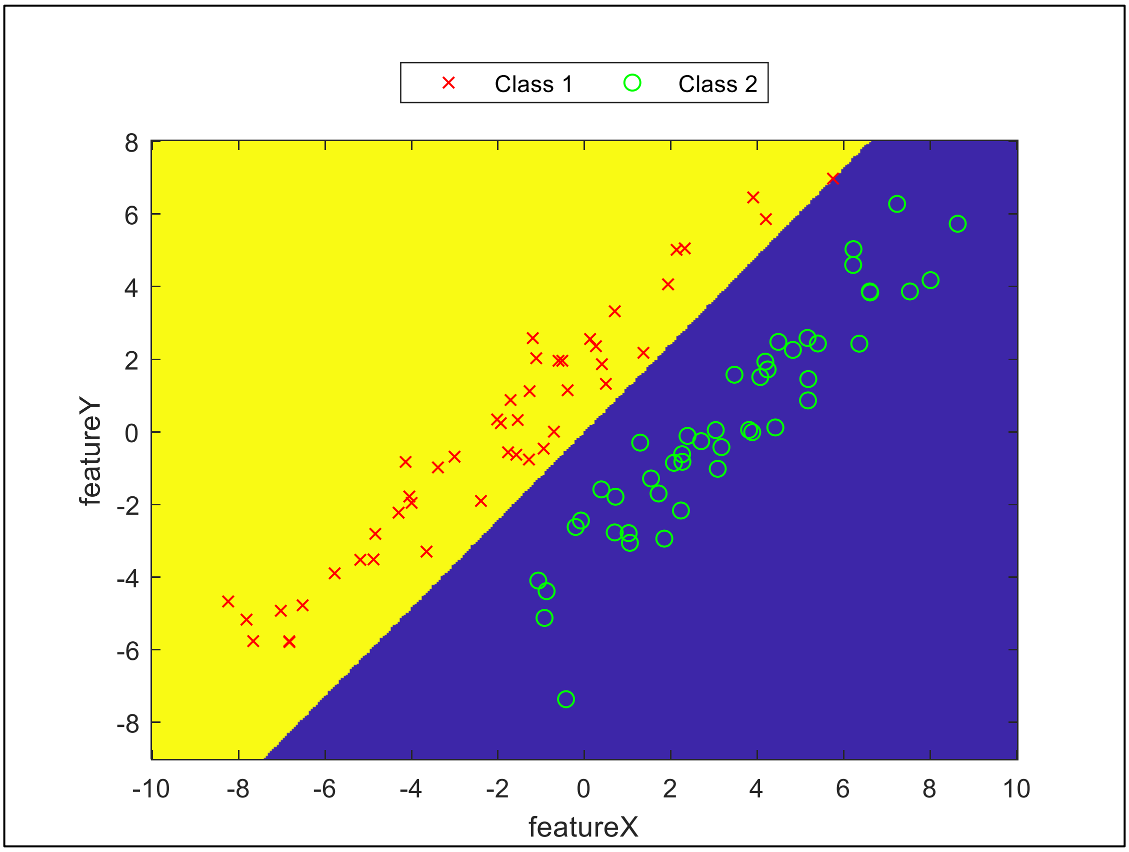

Fig. 5 Training Data (Synthetic 3) Plot

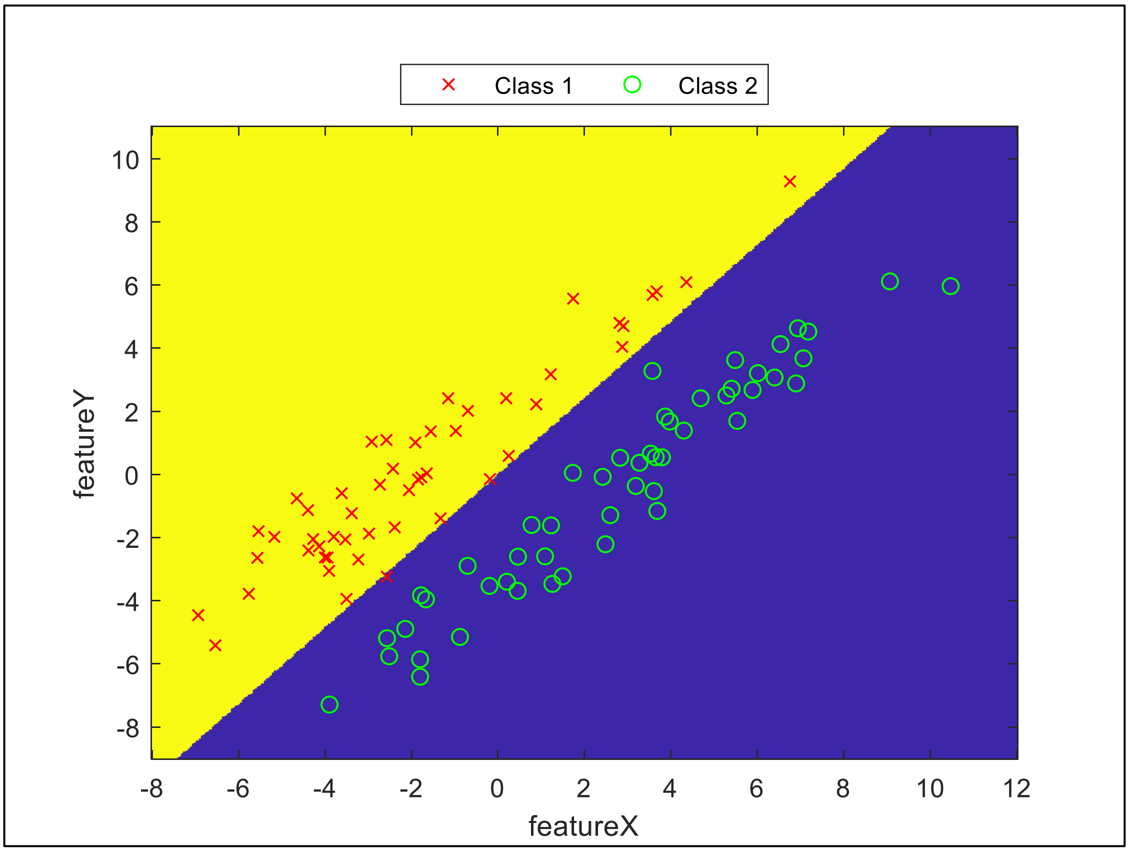

Fig. 6 Testing Data (Synthetic 3) Plot

Observations:

The 2-class perceptron classifier was coded and the results with plots are shown above.

We have used Synthetic Dataset 1 and 2 from Homework1 so that we can compare error rate of Nearest Mean Classifier and Perceptron Classifier.

Comparison	**Synthetic 1**		**Synthetic 2**	
	Nearest Mean	**Perceptron**	**Nearest Mean**	**Perceptron**
Training	21	2	3	2
Testing	24	2	4	3

As seen, there is lot of difference in the error rates with perceptron and nearest mean. We can conclude that Perceptron classifier performs better than Nearest Mean and classifies more accurately. In perceptron, we update the weight after each iteration giving better results which is not in the case of Nearest Mean where weight is calculated only once.

Sourabh Tirodkar EE559

HW # 4

Q. 2) $\Delta \underline{w}(i) = \underline{w}(i+1) - \underline{w}(i)$

& $\underline{w}(i) = \underline{w}_0 \forall i$

$J(\underline{w}) = \sum_{n=1}^{N} J_n(\underline{w})$

↳ criterion func^n

$\eta(i) = \eta$ = constant.

a) For stochastic gradient descent variant 2.

$E\{\Delta \underline{w}(i)\} = ?$

$\underline{w}(i+1) - \underline{w}(i) = -\eta \nabla_w J_n(\underline{w}(i))$

$= \Delta \underline{w}(i)$

$\therefore \Delta \underline{w}(i) = -\eta \nabla_w J_n(\underline{w}_0)$

$\therefore E(x) = \sum x_i \, p(x_i)$

expected value.

n is randomly choosen from $\{1, 2, \ldots, N\}$ with uniform probability.

$$P[\Delta w_n(i)] = 1/N.$$

$$\therefore E\{\Delta \underline{w}(i)\} = \sum_{n=1}^{N} \Delta w(i)\, P(\Delta w(i))$$

$$= \frac{1}{N} \sum \Delta \underline{w}(i)$$

$$= \frac{1}{N} \sum_{n=1}^{N} -\eta\, \nabla_w J_n(\underline{w}(i))$$

$$= \frac{-\eta}{N} \sum \nabla_w J_n(\underline{w}(i))$$

$$= \frac{-\eta}{N} \nabla_w J(\underline{w}(i))$$

$$\therefore \boxed{E\{\Delta \underline{w}(i)\} = \frac{-\eta}{N} \nabla_w J(w_0)}$$

b) ~~For~~ batch gradient descent

$$\underline{w}(i+1) = \underline{w}(i) - \eta \nabla_w J(\underline{w}(i))$$

$$\Delta w(i) = -\eta \nabla_w \sum_{n=1}^{N} J_n(w_0)$$

Now $E\{\Delta w(i)\} = \frac{-\eta}{N} \nabla_w J(w_0)$

Batch ~~Stochastic~~ GD variat z.

$$= \frac{-\eta}{N} \nabla_w \sum_{n=1}^{N} J_n(w_0)$$

1) In Batch ~~stochastic~~ gradient update, the $E(x)$ is mean of weight update.

2) Since in batch descent, we randomize data with replacement & update the weights.

It will have batches of training set.

Weights changes after one epoch.

3) In stochastic G.D., we choose each data point randomly with replacement to update weights. Weight updation takes place after each data point.

4) Thus, it makes sense to compare the expected value of stochastic & batch G.D.

5) Since E(x) of stochastic G.D. variant 2 is average of weight updates in

batch G.D.