Machine Learning - Assignment 1

— Shikhar Vashishth

Problem-1:

$P(X)$ is defined as:

$$P(X = x1) = 0.1 + 0.1 = 0.2$$

$$P(X = x2) = 0.2 + 0.2 = 0.4$$

$$P(X = x3) = 0.1 + 0.3 = 0.4$$

$P(Y)$ is defined as:

$$P(Y = y1) = 0.1 + 0.2 + 0.1 = 0.4$$

$$P(Y = y2) = 0.1 + 0.2 + 0.3 = 0.6$$

$P(X|Y)$ is defined as:

$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

$$P(X = x1|Y = y1) = 0.1/0.4 = 1/4$$

$$P(X = x2|Y = y1) = 0.2/0.4 = 1/2$$

$$P(X = x3|Y = y1) = 0.1/0.4 = 1/4$$

$$P(X = x1|Y = y2) = 0.1/0.6 = 1/6$$

$$P(X = x2|Y = y2) = 0.2/0.6 = 1/3$$

$$P(X = x3|Y = y2) = 0.3/0.6 = 1/2$$

$P(Y|X)$ is defined as:

$$P(Y|X) = \frac{P(X,Y)}{P(X)}$$

$$P(Y = y1|X = x1) = 0.1/0.2 = 1/2$$

$$P(Y = y2|X = x1) = 0.1/0.2 = 1/2$$

$$P(Y = y1|X = x2) = 0.2/0.4 = 1/2$$

$$P(Y = y2|X = x2) = 0.2/0.6 = 1/3$$

$$P(Y = y1|X = x3) = 0.1/0.4 = 1/4$$

$$P(Y = y2|X = x3) = 0.3/0.4 = 3/4$$

Problem-2:

$$P(x|\omega_1) \sim \mathcal{N}(0, I) = \frac{1}{\sqrt{(2\pi)^2 |I|}} \exp\left(-\frac{1}{2}(x-0)^T I (x-0)\right)$$

$$= \frac{1}{2\pi} \exp\left(-\frac{1}{2} x^T x\right)$$

$$P(x|\omega_2) \sim \mathcal{N}(\mu, I) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x-\mu)^T (x-\mu)\right) \quad ; \mu = [1, 1]^T$$

$$P(\omega_1|x) = \frac{P(x|\omega_1) P(\omega_1)}{P(x)}$$

$$P(\omega_2|x) = \frac{P(x|\omega_2) P(\omega_2)}{P(x)}$$

Bayes classifier will predict ω_1 for a given x if

$$P(\omega_1|x) > P(\omega_2|x)$$

$$\frac{P(x|\omega_1) P(\cancel{\omega_1})}{P(\cancel{x})} > \frac{P(x|\omega_2) P(\cancel{\omega_2})}{P(\cancel{x})} \quad [\because P(\omega_1) = P(\omega_2) = 0.5]$$

$$\frac{P(x|\omega_1)}{P(x|\omega_2)} > 1$$

$$\frac{\frac{1}{2\pi} \exp\left(-\frac{1}{2} x^T x\right)}{\frac{1}{2\pi} \exp\left(-\frac{1}{2}(x-\mu)^T (x-\mu)\right)} > 1$$

$$\exp\left(-\frac{1}{2} x^T x + \frac{1}{2}(x-\mu)^T (x-\mu)\right) > 1$$

$$\exp\left(-\frac{1}{2} \cancel{x^T x} + \frac{1}{2}(\cancel{x^T x} - 2x^T\mu + \mu^T\mu)\right) > 1$$

$$\exp\left(\frac{1}{2}\left(\|\mu\|^2 - 2x^T\mu\right)\right) > 1 \quad \left[\begin{array}{l} \|\mu\|^2 = 2 \\ \text{since } \mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{array}\right]$$

$$\exp\left(\frac{1}{2}(2 - 2x^T\mu)\right) > 1$$

$$\exp(1 - x^T\mu) > 1$$

$$\exp(1 - x^T\mu) > 1$$

$$\exp\left(1 - (x_1\ x_2)\begin{pmatrix}1\\1\end{pmatrix}\right) > 1$$

$$\exp(1 - x_1 - x_2) > 1$$

Hence, Bayes classifier will $\Rightarrow$

predict ω_1, if $\exp(1 - x_1 - x_2) > 1$

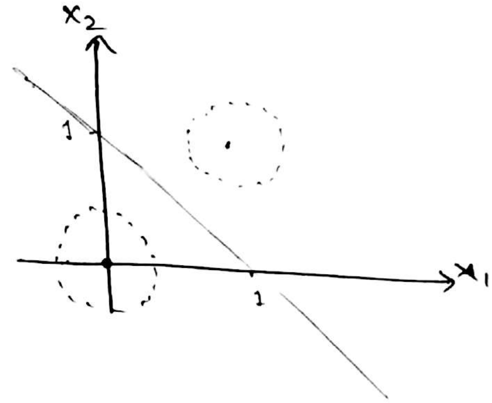
predict ω_2, otherwise

or $\exp(1 - x_1 - x_2) > 1$

$1 - x_1 - x_2 > 0$ [Taking log both sides]

$1 > x_1 + x_2$

$$h(x) = \begin{cases} \omega_1, & \text{if } x_1 + x_2 < 1 \\ \omega_2, & \text{otherwise} \end{cases}$$



Problem 3:

$$l_c(y,\hat{y}) = \begin{cases} c, & \text{if } y=-1, \hat{y}=1 \\ 1-c, & \text{if } y=1, \hat{y}=-1 \\ 0, & \text{if } y=\hat{y} \end{cases}$$

$$\eta(x) = P(y=1|x)$$

Bayes optimal classifier, at every x will make prediction s.t $l_c(y, h(x))$ is minimum.

So, for a given x, h^* (Bayes classifier) will predict '+1' if

$$\underset{(x,y)\sim D}{E}\left[l_c(y, h^*(x)=+1)\right] < \underset{(x,y)\sim D}{E}\left[l_c(y, h^*(x)=-1)\right]$$

$$l_c(y=1,\hat{y}=+1)\,P(y=1|x) + l_c(y=-1,\hat{y}=1)\,P(y=-1|x)$$

$$< l_c(y=1,\hat{y}=-1)\,P(y=1|x) + l_c(y=-1,\hat{y}=-1)\,P(y=-1|x)$$

$$(0)\,\eta(x) + (c)(1-\eta(x)) < (1-c)(\eta(x)) + (0)(1-\eta(x))$$

$$(c)(1-\eta(x)) < (1-c)(\eta(x))$$

$$c - \cancel{c\eta(x)} < \eta(x) - \cancel{c\eta(x)}$$

$$c < \eta(x)$$

Hence,

$$h^*(x) = \begin{cases} +1, & \text{if } c < \eta(x) \\ -1, & \text{otherwise} \end{cases}$$

New loss function:

$$l_c(y,\hat{y}) = \begin{cases} a, & \text{if } y=1, \hat{y}=-1 \\ b, & \text{if } y=-1, \hat{y}=1 \\ 0, & \text{if } y=\hat{y} \end{cases}$$

Bayes classifier (h^*) will predict '+1' for a given x if

$$E_{(x,y)\sim D}\left[l_c(y, h^*(x)=1)\right] < E_{(x,y)\sim D}\left[l_c(y, h^*(x)=-1)\right]$$

$$l_c(y=1,\hat{y}=1)P(y=1|x) + l_c(y=-1,\hat{y}=1)P(y=-1|x)$$

$$< l_c(y=1,\hat{y}=-1)P(y=1|x) + l_c(y=-1,\hat{y}=-1)P(y=-1|x)$$

$$(0)(\eta(x)) + (b)(1-\eta(x)) < (a)(\eta(x)) + (0)(1-\eta(x))$$

$$b - b\eta(x) < a\eta(x)$$

$$b < \eta(x)(a+b)$$

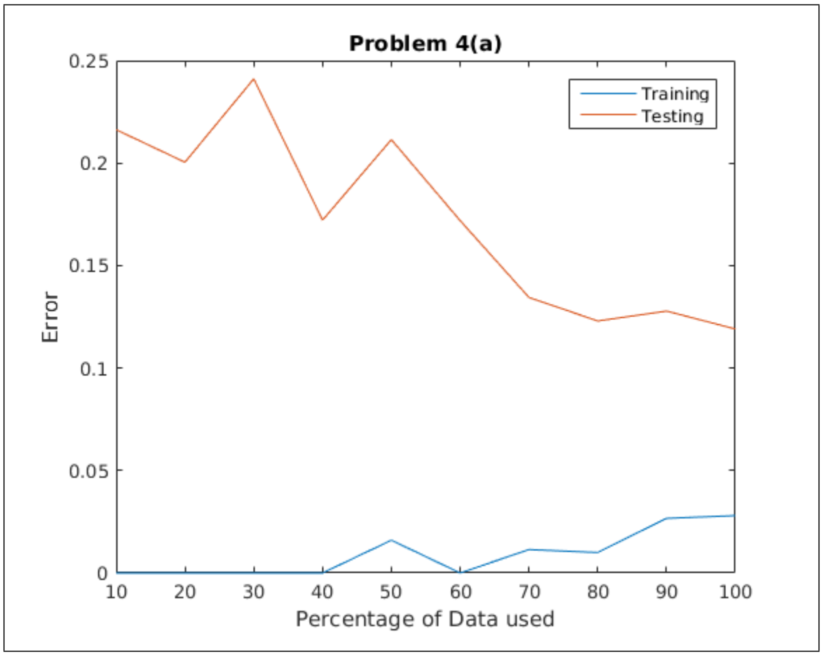
$$\eta(x) > \frac{b}{(a+b)}$$

Hence,

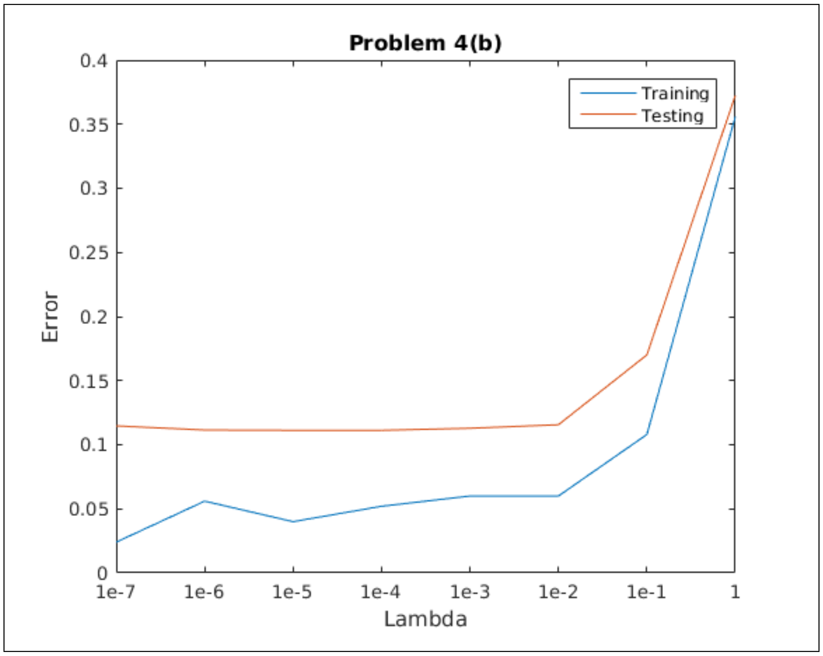
$$h^*(x) = \begin{cases} +1, & \text{if } \eta(x) > \frac{b}{(a+b)} \\ -1, & \text{otherwise} \end{cases}$$

Problem-4:

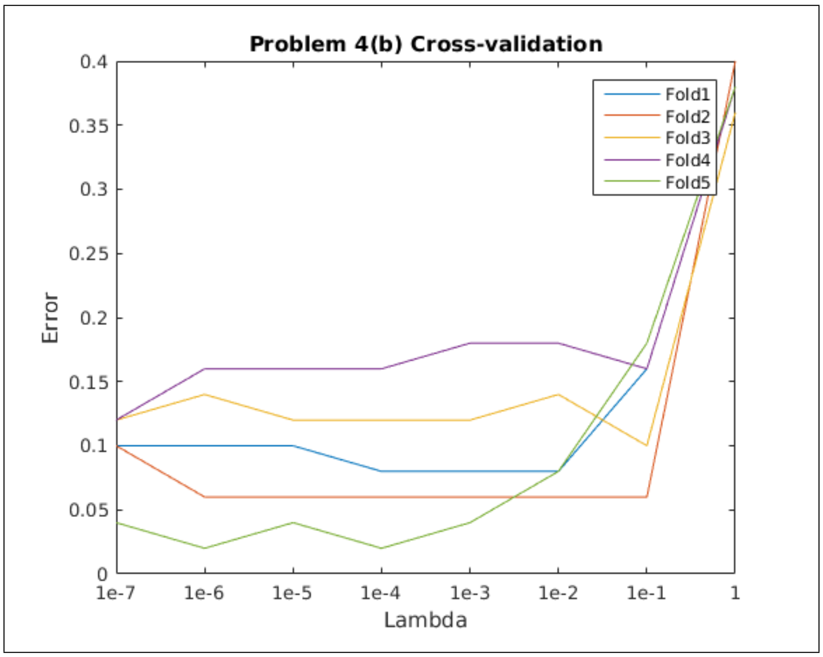
Part (a):



Part (b):



λ	Training Error	Test Error
1.0e-07	0.0240	0.1147
1.0e-06	0.0560	0.1115
1.0e-05	**0.0400**	**0.1112**
0.0001	0.0520	0.1112
0.0010	0.0600	0.1128
0.0100	0.0600	0.1156
0.1000	0.1080	0.1701
1	0.3560	0.3726



Cross-validation:

λ	Avg Training Error	Avg Test Error
1.0e-07	**0.0240**	**0.0240**
1.0e-06	0.0400	0.0400
1.0e-05	0.0380	0.0380
0.0001	0.0460	0.0460
0.0010	0.0510	0.0510
0.0100	0.0600	0.0600
0.1000	0.1160	0.1160
1	0.3730	0.3730

The cross-validation method gave us a different value of λ as $1.0e-07$ which is different from the λ value which we got before. The cross validation process doesn't select the right value of λ.

Problem-5:

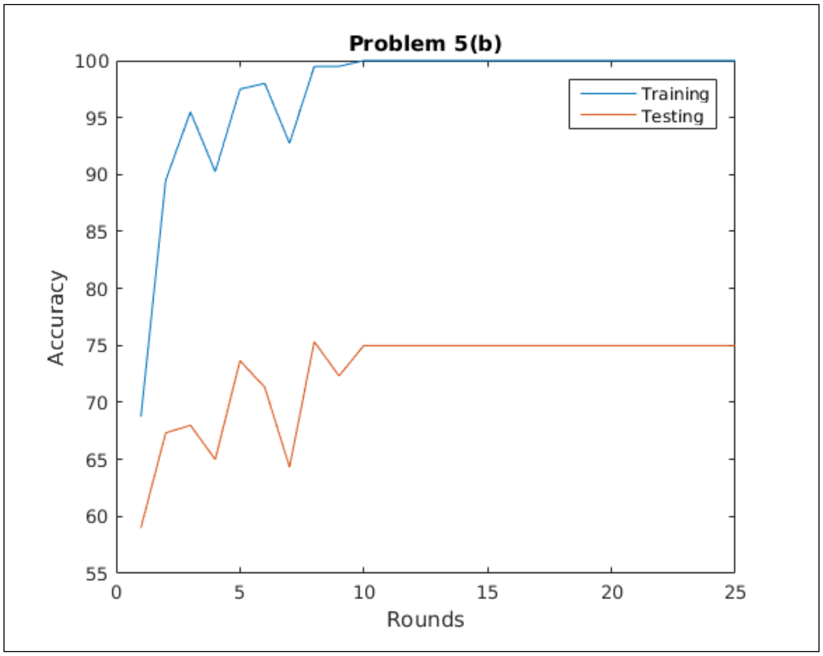
Part (a):

Training Accuracy: **99.75**
Testing Accuracy: **75.67**

Table 1: Confusion Matrix

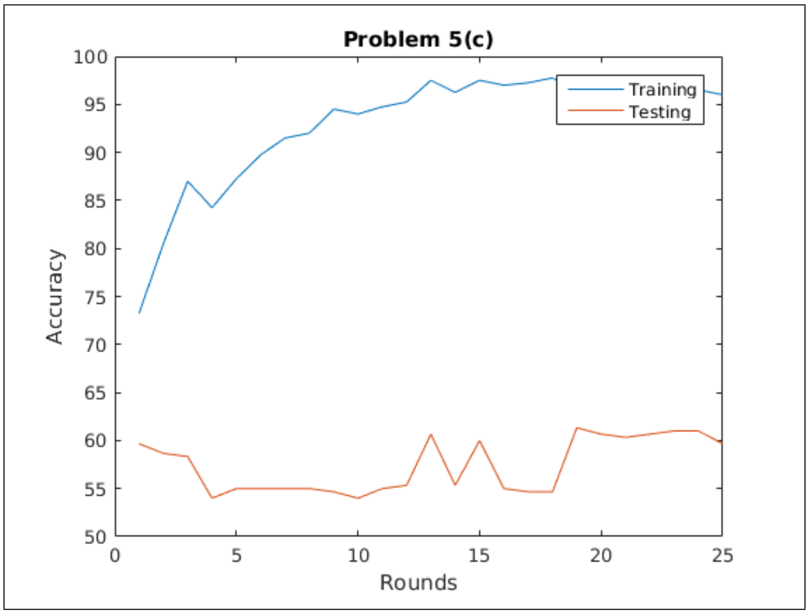
	$\hat{y} = +1$	$\hat{y} = +1$
$y = +1$	118	33
$y = -1$	40	109

Part (b):



Part (c):

η value taken as 0.25.

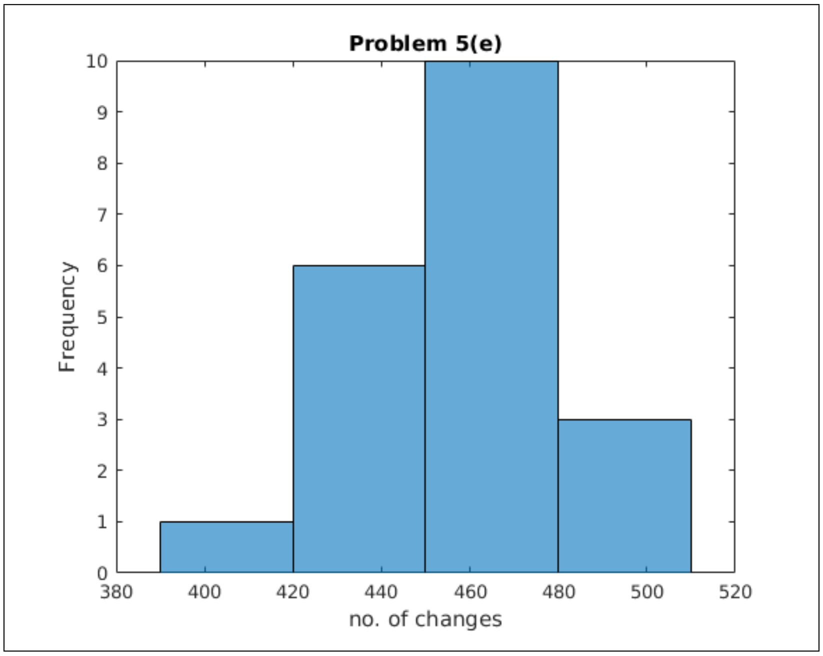


Part (d):

Top 10 influential features:

1. titanic
2. bad
3. no
4. they
5. story
6. nothing
7. seagal
8. would
9. bit
10. any

Part (e):



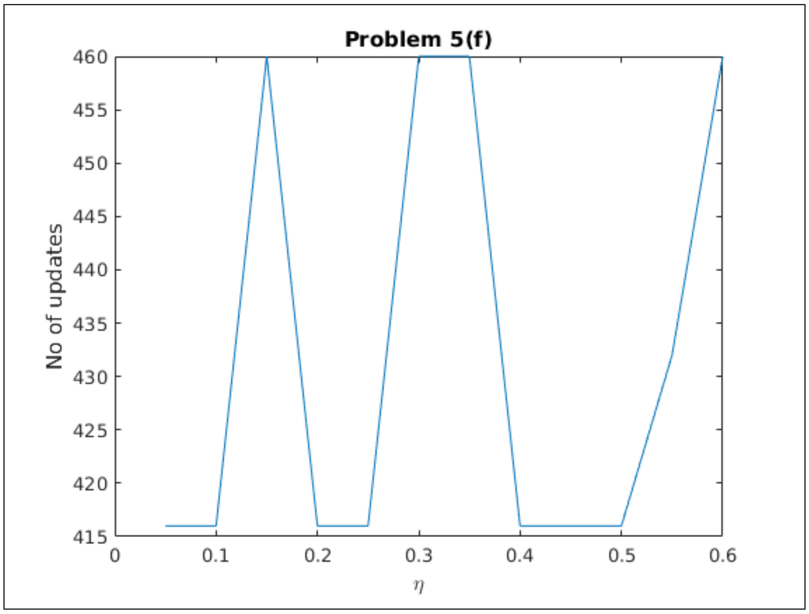
We can see from the table below that there exist a positive value of γ thus, the data is linearly separable. Also the upper bound from the following inequality is correct:

$$\gamma \leq \frac{R^2||w||^2}{M}$$

Gamma	Gamma upper bound
5	7.1022 e+07
10	6.9681 e+07
11	6.2964 e+07
1	6.6023 e+07
7	6.4864 e+07
7	6.5349 e+07
6	6.3785 e+07
10	5.8274 e+07
10	6.5096 e+07
4	6.5724 e+07
4	6.4086 e+07
8	6.4305 e+07
0	6.1728 e+07
16	6.9228 e+07
2	6.5416 e+07
11	6.4969 e+07
18	5.8436 e+07
5	6.6827 e+07
7	6.7941 e+07
0	6.7997 e+07

Part (f):

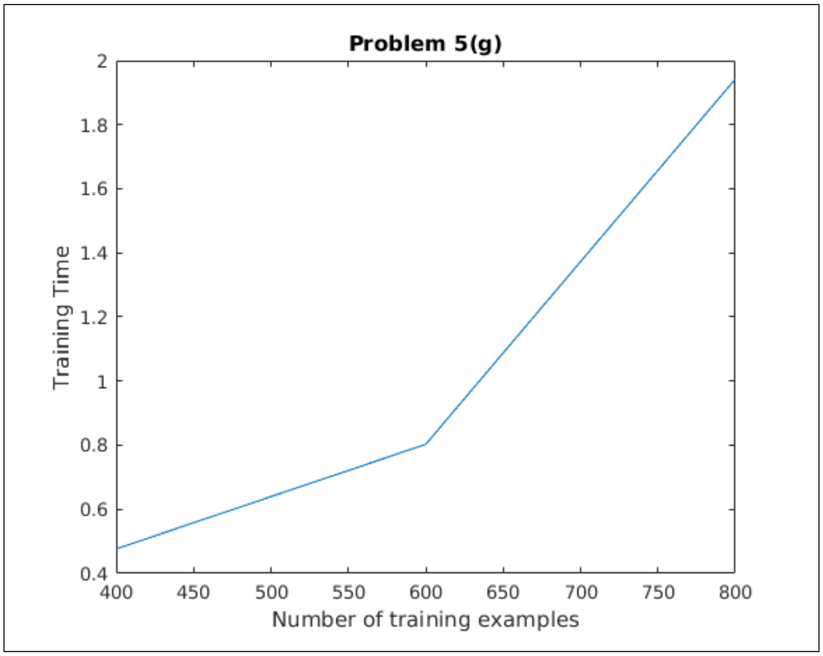
η	Updates	Training Acc.	Test Acc
0.05	416	100.0	75.0
0.10	416	100.0	75.0
0.15	460	100.0	75.0
0.20	416	100.0	75.0
0.25	416	100.0	75.0
0.30	460	100.0	75.0
0.35	460	100.0	75.0
0.40	416	100.0	75.0
0.45	416	100.0	75.0
0.50	416	100.0	75.0
0.55	432	100.0	71.3
0.60	460	100.0	75.0



Part (g):

Dataset	Training Time	Test Accuracy
small	0.4764	75.0
medium	0.8028	75.0
large	1.9427	75.3

As can be seen from the graph below, the runtime complexity of perceptron increases linearly with the number of training examples.


Problem 5(g)
Training Time
Number of training examples
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
400
450
500
550
600
650
700
750
800