Fileri Farhan Witjaksono

Algorithm for Machine Learning and Inference

Final Exam

1) **Linear Regression**

$$L(g) = \sum_i \left(g(x_i) - y_i \right)^2, \quad L(h) = \sum_i (h(x_i) - y_i)^2$$

$$g(x) = ax + b, \quad h(x) = cx^2 + dx + f$$

Answer :

(a)

The squared loss is :

$$\mathcal{L}(g) = \sum_i (ax_i + b - y_i)^2, \quad \mathcal{L}(h) = \sum_i (cx_i^2 + dx_i + f - y_i)^2$$

Assume that we put the parameters in a vector w, then the loss function $\mathcal{L}(g)$ becomes

$$\mathcal{L}(g) = \sum_i \left(w^T x_i - y_i \right)^2, \quad w = \begin{bmatrix} b \\ a \end{bmatrix}$$

Then, we collect x_i and y_i in a vector X and y, we will then have

$$\mathcal{L}(g) = (wX - Y)^T (wX - Y), \quad X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

then, taking partial derivatives with respect to each parameter, we will have

$$\frac{\partial \mathcal{L}(g)}{\partial w} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_0} \\ \frac{\partial \mathcal{L}}{\partial w_1} \end{bmatrix}$$

Setting the partial derivatives to zero $\frac{\partial \mathcal{L}}{\partial w} = 0$,

$$\frac{\partial}{\partial w} \left((wX - y)^T (wX - y) \right) = \frac{\partial}{\partial w} \left(y^2 + X^T X w^T w - X^T w^T y - X w y^T \right)$$

$$0 = 2X^T X w - 2X^T y$$

$$X^T X w = X^T y$$

Multiplying each side with $(X^TX)^{-1}$ gives

$$(X^TX)^{-1}(X^TXw) = (X^TX)^{-1}X^Ty$$

$$\hat{w} = (X^TX)^{-1}X^Ty$$

Then, inserting the values that we have from the problems, we will have

$$w = \begin{bmatrix} b \\ a \end{bmatrix}, \quad X = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 5.7 \\ 8 \\ 10.3 \\ 11.6 \end{bmatrix}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}$$

$$\hat{w} = (X^TX)^{-1}X^Ty$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 5.7 \\ 8 \\ 10.3 \\ 11.6 \end{bmatrix}$$

$$= \left(\begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \right)^{-1} \begin{bmatrix} 38.6 \\ 10.2 \end{bmatrix} = \frac{1}{50} \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 38.6 \\ 21.8 \end{bmatrix}$$

$$= \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 38.6 \\ 21.8 \end{bmatrix}$$

$$\hat{w} = \begin{bmatrix} 7.72 \\ 2.18 \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$$

Hence, the model that we have is

$$g(x) = \hat{w}x = ax + b =$$

$$2.18x + 7.72$$

For the second loss, we can construct the loss function in matrix/vectors form as below

$$\mathcal{L}(h) = \sum_i \left(cx_i^2 + dx_i + f - y_i\right)^2$$

Similar with the first loss function, we construct the loss function in a matrix/vector form :

$$w = \begin{bmatrix} f \\ d \\ c \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & \vdots & \vdots \\ 1 & \vdots & \vdots \\ 1 & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \quad y = \begin{bmatrix} 3 \\ 5.7 \\ 8 \\ 10.3 \\ 11.6 \end{bmatrix}$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \end{bmatrix}$$

Estimating the model parameters,

$$\hat{w} = \left(X^T X\right)^{-1} X^T y$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 5.7 \\ 8 \\ 10.3 \\ 11.6 \end{bmatrix}$$

$$= \left(\begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix} \right)^{-1} \begin{bmatrix} 38.6 \\ 21.8 \\ 74.4 \end{bmatrix}$$

$$\hat{w} = \begin{bmatrix} 0.486 & 0 & -0.143 \\ 0 & 0.1 & 0 \\ -0.143 & 0 & 0.071 \end{bmatrix} \begin{bmatrix} 38.6 \\ 21.8 \\ 74.4 \end{bmatrix} = \begin{bmatrix} 8.12 \\ 2.18 \\ -0.29 \end{bmatrix} = \begin{bmatrix} f \\ d \\ c \end{bmatrix}$$

Hence, the model that we have is

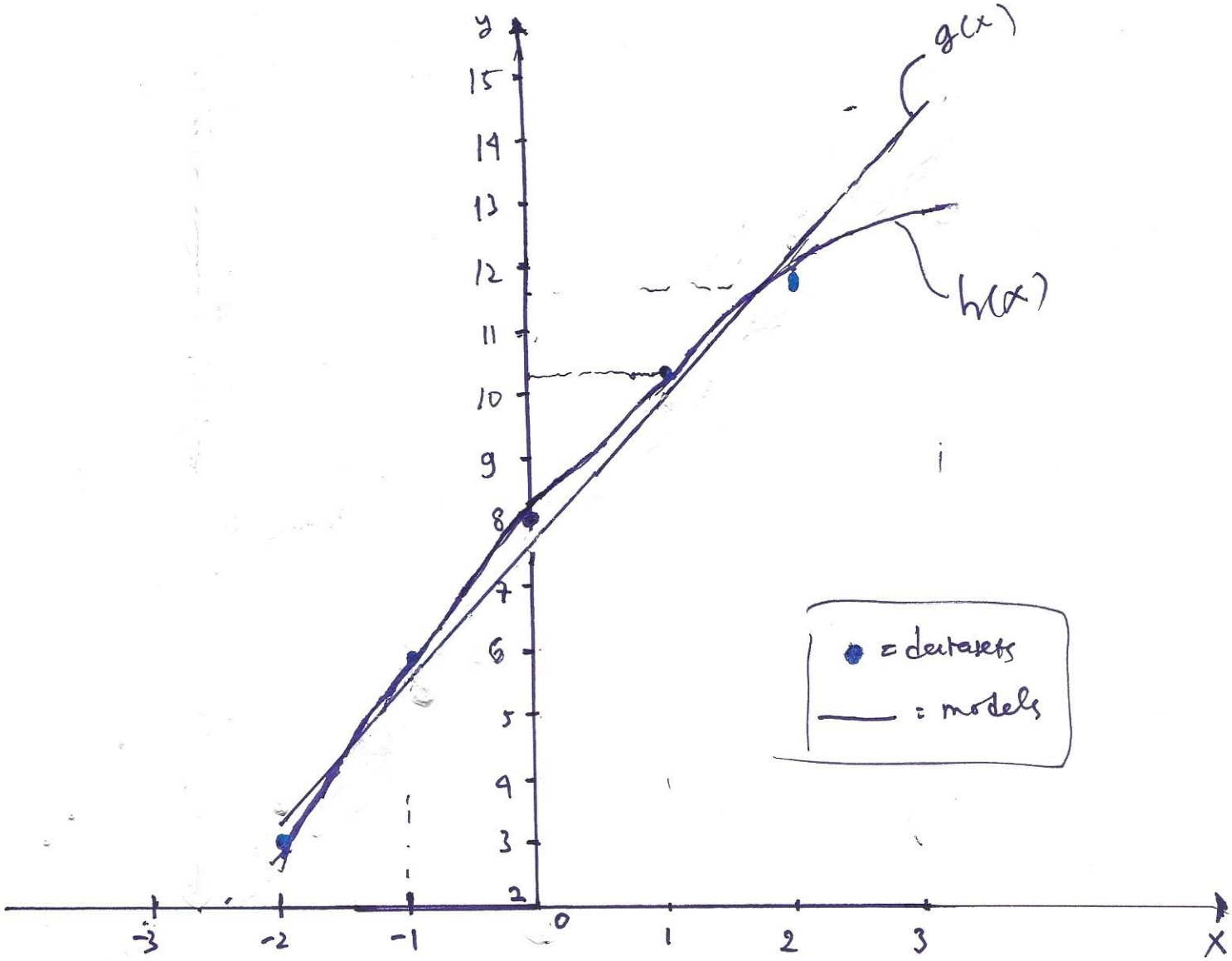
$$h(x) = cx^2 + dx + f = -0.29x^2 + 2.18x + 8.12$$

Finding the points to plot according to the models.

table 1. $\alpha = 0$

g(x)	h(x)	x_i	x_i^2	y	$L(g, x, y)$	$L(h, x, y)$
1.18	-0.22	-3	9	NA	NA	NA
3.36	2.96	-2	4	3	0.1296	0.0016
5.54	5.74	-1	1	5.7	0.0256	0.0016
7.72	8.12	0	0	8	0.0784	0.0144
9.90	10.1	1	1	10.3	0.16	0.040
12.08	11.68	2	4	11.6	0.2304	0.0064
14.26	12.86	3	9	NA	NA	NA

Hence, plotting both model in a single curve together with the dataset gives



b) Let $\alpha = 6$. then re-calculating the parameters estimation, for $g(x)$

$$w = \begin{bmatrix} b \\ a \end{bmatrix}, X = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, y = \begin{bmatrix} 3 \\ 5.7 \\ 14 \\ 10.3 \\ 11.6 \end{bmatrix}, X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}$$

$$\hat{w} = \underbrace{(X^T X)^{-1}}_{\text{fixed term}} \underbrace{X^T y}_{\text{changing terms}}$$

$$= \frac{1}{50} \begin{bmatrix} 10 & 0 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 5.7 \\ 14 \\ 10.3 \\ 11.6 \end{bmatrix}$$

$$= \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 44.6 \\ 21.8 \end{bmatrix} = \begin{bmatrix} 8.92 \\ 2.18 \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$$

Hence, the model that we have become

$$g(x) = \hat{w}x = ax + b = 2.18x + 8.92$$

For the second model $h(x)$, we calculating the parameter estimation

$$w = \begin{bmatrix} f \\ d \\ c \end{bmatrix}, X = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, y = \begin{bmatrix} 3 \\ 5.7 \\ 14 \\ 10.3 \\ 11.6 \end{bmatrix}$$

$$\hat{w} = \underbrace{(X^T X)^{-1}}_{\text{fixed term}} \underbrace{X^T y}_{\text{changing term}}$$

$$= \left(\begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 5.7 \\ 14 \\ 10.3 \\ 11.6 \end{bmatrix}$$

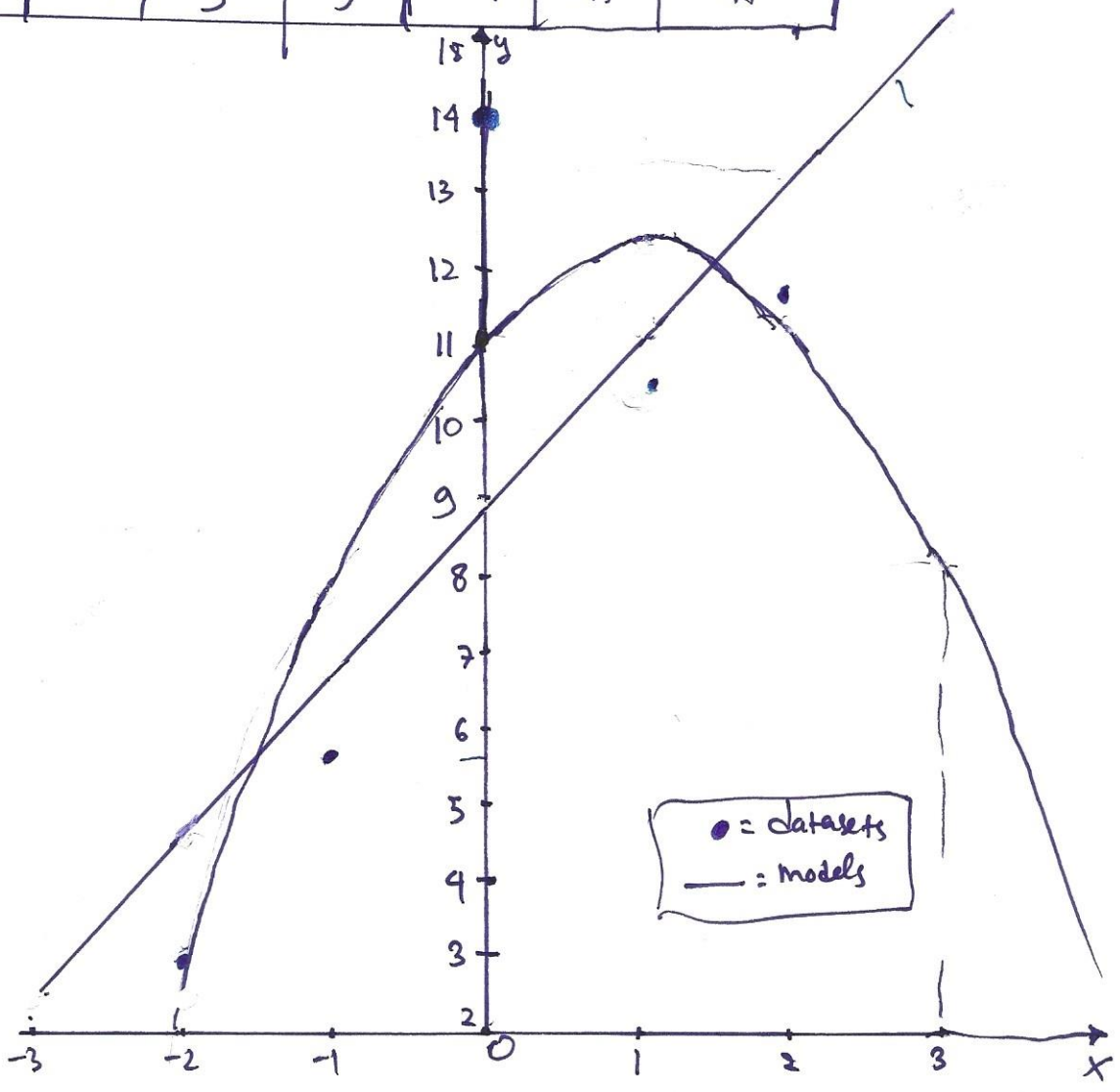
$$\begin{bmatrix} 0.486 & 0 & -0.143 \\ 0 & 0.1 & 0 \\ -0.143 & 0 & 0.071 \end{bmatrix} \begin{bmatrix} 44.6 \\ 21.8 \\ 74.4 \end{bmatrix} = \begin{bmatrix} 11.03 \\ 2.18 \\ -1.06 \end{bmatrix} = \begin{bmatrix} f \\ d \\ c \end{bmatrix}$$

The model $h(x)$ becomes $h(x) = cx^2 + dx + f = -1.06x^2 + 2.18x + 11.03$

Finding the points to plot according to the models

table 2. $\alpha = 6$

$g(x)$	$h(x)$	x_i	x^2	y	$L(g, x, y)$	$L(h, x, y)$
2.38	-5.05	-3	9	NA	NA	NA
4.56	2.43	-2	4	3	2.43	0.325
6.74	7.79	-1	1	5.7	1.08	4.87
8.92	11.03	0	0	14	25.81	8.82
11.1	12.15	1	1	10.3	0.64	3.42
13.28	11.15	2	4	11.6	2.82	0.203
15.46	8.03	3	9	NA	NA	NA



We could observe that both models becomes underfit. and the loss becomes considerable larger. In previous question, we could easily determine that the quadratic model is the best model. However, here as we add $\alpha = 6$, it becomes harder to determine the best model. This is due to the fact that at $x = 0$, data is evilod from the other data. Hence, fitting becomes hard.

c) To show that we always have $\min \mathcal{L}(g,x,y) \geq \min \mathcal{L}(h,x,y)$,

Answer: In the first model with $\alpha = 0$,

we could observe that the minimum loss of the linear model is

$$\min \mathcal{L}(g,x,y) = 0.0256$$

for $\alpha = 6$,

$$\min \mathcal{L}(g,x,y) = 0.64$$

In the second model with $\alpha = 0$,

$$\min \mathcal{L}(h,x,y) = 0.0016$$

For $\alpha = 6$,

$$\min \mathcal{L}(h,x,y) = 0.203$$

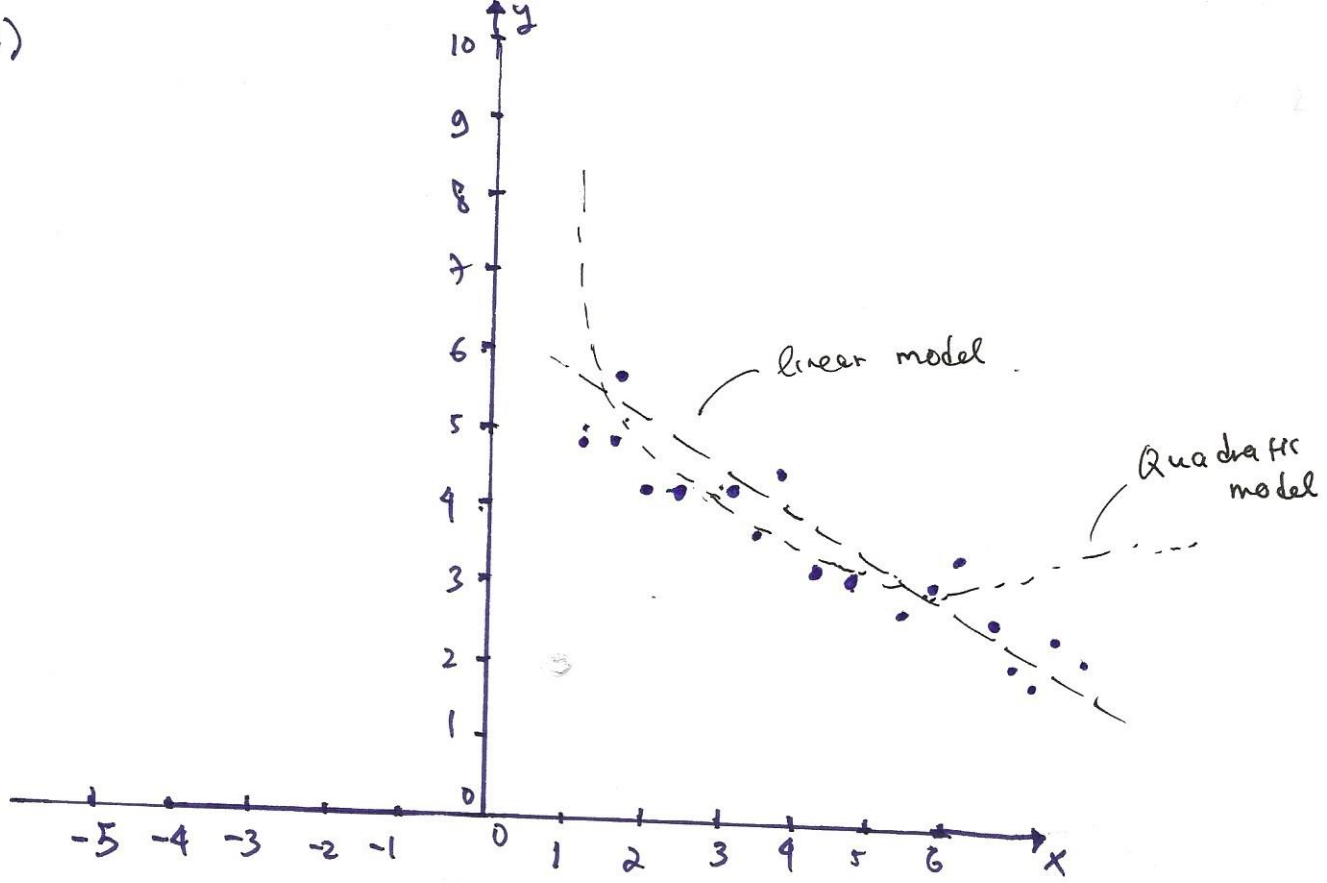
From these two datasets, we could conclude that we always have $\min \mathcal{L}(g,x,y) \geq \min \mathcal{L}(h,x,y)$

Theoretically, we could also say that since loss always decreases as the model is made more complex, we will always have

$$\min \mathcal{L}(g,x,y) \geq \min \mathcal{L}(h,x,y)$$

due to the fact that model $h(x)$ has higher complexity than model $g(x)$

e)



In this kind of datas, where the data is V almost perfectly linearly decreasing : trend combined with the fact that the

variance is small / where datas is more compact to each other, is better fitted using linear regression model.

It is shown that using quadratic model, the data is underfitted whereas using linear model, the datas were fitted well.

3.) Classification

Answer:

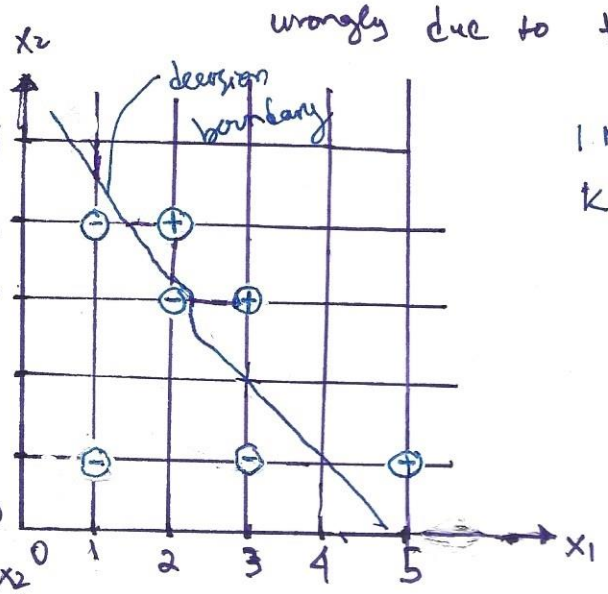
(a) The figure with decision boundary 1 belongs to 1-nearest neighbor classifier.

This is due to the fact that we have an island as a result of overfitting. We could also say that for 1-nearest neighbor, it is heavily influenced by noise exemplified by the island.

The figure with decision boundary 2 → k-nearest neighbor ($k>2$)

The figure will be smoother than with 1-nearest neighbor. However, there is one outlier (class +) which should have been classified as class (−), however classified wrongly due to the decision boundary.

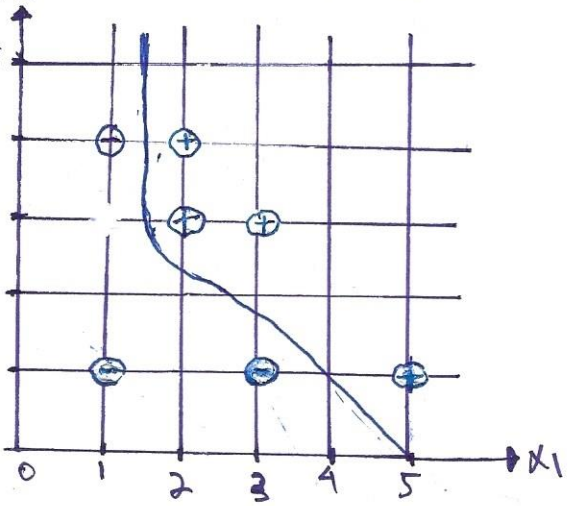
(b) a.)



1 nearest neighbor → ? is nearest neighbor

$k=1$ → ? would be classified as ⊖

b.)

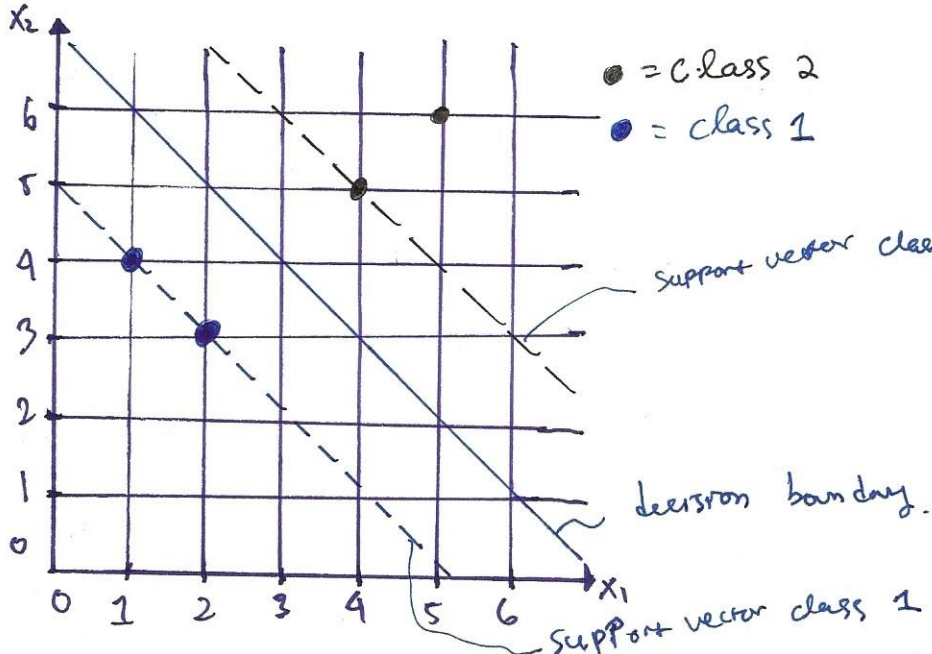


3 nearest neighbor

→ { 2 (+) class
  { 1 (−) class

$k=3$ → ? would be classified as ⊕

c)



The plot for the 4 training points could be illustrated as above. From the plot, we could say the following:

- the decision boundary could be expressed mathematically by linear equation as: $x_2 = -x_1 + 7$
- the support vector for class 1, could be expressed mathematically as: $x_2 = -x_1 + 5$
- The support vector for class 2 could be expressed as: $x_2 = -x_1 + 9$

Then, we know that the general equation for decision boundary is:

$$w_1x_1 + w_2x_2 + b = 0$$

$$x_2 = -x_1 + 7 \longleftrightarrow x_1 + x_2 - 7 = 0$$

Hence, the weights and biases are

$$w_1 = 1, \; w_2 = 1, \; b = -7$$

d) Limitations of linear hard SVM

1) In linear hard SVM, as it can only work well when data is completely linearly separable without any errors in the form of noise or outliers. If there is an error, since a single outlier can determine the boundary, the hard margin SVM would not work well. Hence, it possibly overfits to a particular dataset and therefore cannot generalize well.

2) linear hard SVM is not very suited for large datasets.

3) In case of when we have more features for each data point than the number of traning data sample, we will have bad performance of SVM.

(e). Assume $p_i = p_j \; \forall i, j$, what is the probability of the document "turn left"?

Answer:

A multinomial distribution is given by:

$$P(x_1, \dots, x_k \mid p_1, \dots, p_k) = d! \frac{(p_1)^{x_1} \times \dots \times (p_k)^{x_k}}{(x_1)! \times \dots \times (x_k!)} = \frac{d!}{\prod_{t=1}^{d} x_t!} \prod_{t=1}^{d} p_t^{x_t}$$

Assuming random variables $x_1, \dots, x_k$ such that $\sum_{k=1}^{k} x_k = d$.

Consider the vocabulary

$$V = \{\text{left}, \text{motorway}, \text{the}, \text{leave}, \text{turn}, \text{and}, \text{Gothenburg}\}$$

$$|V| = d = 7.$$

Let d^M is the multinomial feature vector, then we would have

$$d^M = \{1, 0, 0, 0, 1, 0, 0\}^T$$

Assuming $p_{\text{our}} = 1$ since $p_i = p_j$, Then, the probability of the document "turn left" is

By Multinomial Naive-Bayes $\longrightarrow$ $P(x_1, x_2 \mid C) = \underbrace{\hat{P}(C)}_{\text{we have only single class}} \cdot \prod_i \hat{P}(x_i \mid C) = 1 \cdot \frac{1}{7} \cdot \frac{1}{7} = \frac{1}{49}$

6) A multinomial distribution is given by

general

$$P(x_1,\dots,x_k \mid p_1,\dots,p_k) = d! \frac{(p_1)^{x_1} \times \dots \times (p_k)^{x_k}}{(x_1)! \times \dots \times (x_k)!}$$

$$= \frac{d!}{\prod_{i=1}^{k} x_i!} \prod_{i=1}^{k} p_i^{x_i}$$

$$= \binom{d}{x_1, x_2, x_k} \cdot p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$$

Changing the variable x_i with f_{dk},

$$\frac{d!}{\prod_{i=1}^{k} f_{dk}!} \cdot \prod_{i=1}^{k} p_i^{f_{dk}} = \binom{d}{f_{d1}, f_{d2}, f_{dk}} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$$

Then, taking the log-likelihood

$$\ell(p_1, p_2, \dots, p_k) = \log d! + \left(\sum_{k=1}^{k} (f_{dk} \times \log p_i)\right) - \sum_{k=1}^{k} \log f_{dk}!$$

the arrow shows the relationship before and after log-likelihood calculation.

4.) Deep Learning

a) the convolution operation of the weights (2x2 filters) on the image (2xn) size will result in output of size (1 x n-1) is as shown below

$$\text{Image} \odot \text{Filter} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \end{bmatrix} \odot \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}$$

$$= \begin{bmatrix} z_1 & z_2 & \cdots & z_{n-1} \end{bmatrix}$$

Let's say that the output has an index m, therefore we could generalize the operation with the following equation

$$z_m = \sum_{i=1}^{2} \sum_{j=1}^{2} w_{i,j+m-1} x_{i,j+m-1} + b$$

where index $m = 1, \cdots, n-1$, $b \rightarrow$ bias

b) Given $\frac{dE}{dz_i} \rightarrow$ gradients of loss function E wrt outputs z_i

$$\frac{dE}{dw_{i,j}} =$$

c) For $k > 2$ (multi class classification),

to design the network, we need to take several common measures:

1. Define the output label as a one-hot encoded vector such that when the output is predicted, the value of node corresponding to output should be 1 whereas the other nodes should be 0.

2. Make sure to use Softmax and cross-entropy function, Softmax is used due to the fact that it takes a vector as input as well as it produces an output between 0 and 1. Hence, it is beneficial to use it in multi-classification problem. Regarding cross entropy loss function, it is showing a very good performance for multiclass classifications hence, it is common to use it.

3. Define the feed forward phase. Mind the dimension of the nodes.

4. Design the Backward propagation wisely.

5. Choose the best initialization method. (e.g. He initialization)

6. Interpreting result of multiclass classification is better by using quantitative metrics (e.g. confusion matrix) rather than visual metric since it is easier to evaluate.

7. Tune the network architecture by increasing network complexity, checkpoints/early stopping or learning rate scheduler, dropout, batch normalization etc.

8. choose the best optimizer which performs best for multiclass classification.

d) $\hat{p}_j(x) \longrightarrow$ probability computed by model for input x in class j

$y_i \in \{1, \ldots, k\} \longrightarrow$ corresponding class labels

$i = 1, \ldots, N$

$X_i \longrightarrow$ training dataset consisting of N samples

Answer:

By using Maximum likelihood principle, the likelihood of input x_i is in class y_i could be written as:

$$P(\hat{y}_1, \ldots, \hat{y}_N | x_i) \prod_{i=1}^{N} \hat{p}_{y_i}(x_i)$$

the negative log-likelihood could be expressed as

$$\log\left(P(\hat{y}_1, \ldots, \hat{y}_N | x_i)\right) = -\log \prod_{i=1}^{N} \hat{p}_{y_i}(x_i)$$

The maximization of the negative log-likelihood wrt y_i will results in

$$\mathcal{L}(\hat{y}_i, x_i) = -\sum_{i=1}^{N} \log \hat{p}_{y_i}(x_i)$$

Averaging loss over the whole dataset will give us the Cross Entropy (CE) Loss as

$$CE = -\frac{1}{N} \sum_{i=1}^{N} \log\left(\hat{p}_{y_i}(x_i)\right)$$

5.) Mixture Models and EM algorithm

a) .. $\Theta = \{\pi_i ; \lambda_1 ; \lambda_2\}$ be the set of parameters

- exponential distribution with parameter λ has the density function

$$f(x,\lambda) = \lambda e^{-\lambda x}$$

- The distribution of inaction times is a mixture of 2 exponential distributions:

$$p(X=x) = \pi_i \lambda_1 e^{-\lambda_1 x} + (1-\pi_i)\lambda_2 e^{-\lambda_2 x}$$

- $z_k \longrightarrow$ cluster variable
- Inactivity times samples x_n

3	5	2

Answer:

$$p(x_n \mid z_1=1, \Theta) = \ldots ?$$

Since we have latent variable, thus only one element of z is equal to 1 and all other elements are zero.

$$z_k \in \{0,1\}, \quad \sum_k z_k = 1$$

Then, we know that $z_1 = 1$,

Hence $\boxed{z_2 = 0}$

the marginal (prior) distribution over z is specified in terms of the mixing coefficients π_k.

$$p(z_k=1) = \pi_k$$
$$\updownarrow$$
$$p(z_1=1) = \pi_1$$

for $p(x)$ to be a proper distribution

$$\sum_{k=1}^{z} \pi_k = 1 \text{ and } \pi_k \geq 0, 1 \leq k \leq z$$

therefore, the conditional distribution of the sample x_n given a particular value of z is

$$p(x_n \mid z_1=1, \pi_1, \lambda_1, \lambda_2) = \pi_1 \lambda_1 e^{-\lambda_1 x_n} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_n}$$

$$p(x_1 \mid z_1=1, \pi_1, \lambda_1, \lambda_2) = \pi \lambda_1 e^{-3\lambda_1} + (1-\pi_1)\lambda_2 e^{-3\lambda_2}$$

$$p(x_2 \mid z_1=1, \pi_1, \lambda_1, \lambda_2) = \pi \lambda_1 e^{-5\lambda_1} + (1-\pi_1)\lambda_2 e^{-5\lambda_2}$$

$$p(x_3 \mid \ldots) = \ldots e^{-2\lambda_1} + (1-\pi_1)\lambda_2 e^{-2\lambda_2}$$

Find full data. log. likelihood

b) $P(x_1, x_2, x_3 \mid \Theta)$ in terms of different distributions and parameters?

$X = (x_1, x_2, x_3)$, $N = 3$,

Answer:

The full data log likelihood could be expressed as.

$$\ln P(X \mid \pi_1, \lambda_1, \lambda_2) = \sum_{n=1}^{N} \ln \{ \pi_1 (\lambda_1 e^{-\lambda_1 x_n}) + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_n}) \}$$

$$= \ln \underbrace{\{\pi_1 (\lambda_1 e^{-\lambda_1 x_1}) + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_1})\}}_{a} + \ln \underbrace{[\pi_1 (\lambda_1 e^{-\lambda_1 x_2}) + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_2})]}_{b}$$

$$+ \ln \underbrace{[\pi_1 (\lambda_1 e^{-\lambda_1 x_3}) + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_3})]}_{c}$$

By the identity $\ln(a) + \ln(b) + \ln(c) = \ln(abc)$, we get

$$= \ln [(\pi_1 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_1})(\pi_1 (\lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_1}))$$
$$(\pi_1 (\lambda_1 e^{-\lambda_1 x_3}) + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_3})]$$

$$= \ln [(\pi_1 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_1}) \{\pi_1^2 (\lambda_1^2 e^{-\lambda_1 (x_1+x_3)}) + (1-\pi_1)^2 (\lambda_2^2 e^{-\lambda_2 (x_1+x_3)})$$
$$+ (1-\pi_1)\pi_1 (\lambda_1 \lambda_2 e^{-\lambda_2 x_1 - \lambda_1 x_3})$$
$$+ (1-\pi_1)(\pi_1)(\lambda_1 \lambda_2 e^{-\lambda_1 x_1 - \lambda_2 x_3})]$$

$$= \ln [(\pi_1 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)(\lambda_2 e^{-\lambda_2 x_1}) \{\pi_1^2 (\lambda_1^2 e^{-\lambda_1 (x_1+x_3)} - \lambda_2^2 e^{-\lambda_2 (x_1+x_3)}) + (1-\pi_1)\pi_1 \lambda_1 \lambda_2$$
$$e^{-\lambda_2 (x_1+x_3)} - \lambda_1 (x_1+x_3)$$
$$+ \lambda_2^2 e^{-\lambda_2 (x_1+x_3)}]$$

Inserting $(x_1, x_2, x_3) = \{3, 5, 2\}$

$$= \ln [(\pi_1 \lambda_1 e^{-3\lambda_1} + (1-\pi_1)(\lambda_2 e^{-3\lambda_2}))(\pi_1^2 (\lambda_1^2 e^{-8\lambda_1} - \lambda_2^2 e^{-8\lambda_2}) + (\pi_1 - \pi_1^2)\lambda_1 \lambda_2 e^{-5(\lambda_1+\lambda_2)}$$
$$+ \lambda_2^2 e^{-5\lambda_2}]$$

$$= \ln [(\pi_1^3 (\lambda_1^3 e^{-11\lambda_1} - \lambda_1 \lambda_2^2 e^{-3\lambda_1 - 8\lambda_2}) + (\pi_1^2 - \pi_1^3)\lambda_1^2 \lambda_2 e^{-8\lambda_1 - 5\lambda_2} + \pi_1 \lambda_1 \lambda_2^2 e^{-3\lambda_1 - 5\lambda_2}$$
$$+ (1-\pi_1)\pi_1^2 (\lambda_1^2 e^{-8\lambda_1} - \lambda_2^2 e^{-8\lambda_2})(\lambda_2 e^{-3\lambda_2}) + (1-\pi_1)\lambda_2^3 e^{-8\lambda_2}$$
$$+ (1-\pi_1)(\pi_1 - \pi_1^2)\lambda_1 \lambda_2^2 e^{-5\lambda_1 - 8\lambda_2}]$$

$= \cdots$

$$\gamma(z_{22}) = \frac{\pi_2 \lambda_1 e^{-\lambda_1 x_2} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_2}}{0.0426}$$

$$= \frac{\left((0.6)(2)e^{-10} + (0.4)(3)e^{-15}\right)/0.6}{0.0426}$$

$$= 7.72 \times 10^{-7}$$

$$\gamma(z_{32}) = \frac{\pi_2 \lambda_1 e^{-\lambda_1 x_3} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_3}}{0.0426}$$

$$= \frac{\left((0.6)(2)e^{-9} + (0.4)(3)e^{-6}\right)/0.6}{0.0426}$$

$$= 0.35$$

d) $$N_k = \sum_{n=1}^{3} \gamma(z_{nk})$$

$$N_1 = \gamma(z_{11}) + \gamma(z_{21}) + \gamma(z_{31})$$

$$= 0.052 + 0.019 + 0.18 = 0.249$$

Re-estimating the mixing coefficient as

$$\pi_1 = \frac{N_1}{N} = \frac{0.249}{3} = 0.083$$

$$N_2 = \gamma(z_{12}) + \gamma(z_{22}) + \gamma(z_{32})$$

$$= 0.044 + 7.7\times10^{-7} + 0.35 = 0.394$$

Re-estimating the mixing coeffs.

$$\pi_2 = \frac{N_2}{N} = \frac{0.394}{3} = 0.131$$

the new parameters are

$$\theta_1 = \{\pi_1 = 0.083, \lambda_1 = 2, \lambda_2 = 3, \pi_2 = 0.131\}$$

c) $\Theta_0 = \{\pi_1, \lambda_1, \lambda_2 \mid \lambda_1 = 2, \lambda_2 = 3\}$, $x_1 = 3, x_2 = 5, x_3 = 2$

calculate responsibilities

$\gamma(z_{nk}) = p(z_k = 1 \mid x_n, \Theta_0)$ for all $n \in \{1,2,3\}$, $k = \{1,2\}$

$\sum_{k=1} \pi_k = 1$, $\pi_1 = 0.4 \longrightarrow \pi_2 = 1 - 0.4 = 0.6$

Answer:

$$\gamma(z_{nk}) = p(z_k = 1 \mid x_n, \Theta_0) = \frac{p(z_{nk} = 1)\, p(x_n \mid z_{nk} = 1)}{\sum_{j=1}^{k} p(z_{nj} = 1)\, p(x_n \mid z_{nj} = 1)}$$

$$\gamma(z_{11}) = \frac{(\pi_1 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_1})(0.4)}{\{(\pi_1 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_1}) + (\pi_1 \lambda_1 e^{-\lambda_1 x_2} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_2}) + (\pi_1 \lambda_1 e^{-\lambda_1 x_3} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_3})\}(0.4) + \{(\pi_2 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_1}) + (\pi_2 \lambda_1 e^{-\lambda_1 x_2} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_2}) + (\pi_2 \lambda_1 e^{-\lambda_1 x_3} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_3})\}(0.6)}$$

$$= \frac{(0.4)(2)e^{-6} + (0.6)3e^{-9}}{\{((0.4)(2)e^{-6} + (0.6)3e^{-9}) + ((0.4)(2)e^{-15} + (0.6)(3)e^{-15}) + ((0.4)(2)e^{-9} + (0.6)(3)e^{-6}) + ((0.6)(2)e^{-6} + (0.4)(3)e^{-15}) + ((0.6)(2)e^{-10} + (0.4)(3)e^{-6}) + ((0.6)(2)e^{-9} + (0.4)(3)e^{-6})\}}$$

$$= \frac{2.2 \times 10^{-3}}{\underbrace{(0.004) + (0.002) + (0.019)}_{\times 0.4} + \underbrace{(0.003) + (0.003) + (0.025)}_{\times 0.6}}$$

$$= \frac{0.0022}{0.024 + 0.0186} = \frac{0.0022}{0.0426} = 0.052$$

$$\gamma(z_{21}) = \frac{0.4(\pi_1 \lambda_1 e^{-\lambda_1 x_2} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_2})}{0.0426} = \frac{((0.4)(2)e^{-6} + (0.6)(3)e^{-15})0.4}{0.0426}$$

$$= 0.019$$

$$\gamma(z_{31}) = \frac{\pi_1 \lambda_1 e^{-\lambda_1 x_3} + (1-\pi_1)\lambda_2 e^{-\lambda_2 x_3}}{0.0426} = \frac{((0.4)(2)e^{-9} + (0.6)(3)e^{-6})0.4}{0.0426}$$

$$= 0.18$$

$$\gamma(z_{12}) = \frac{\pi_2 \lambda_1 e^{-\lambda_1 x_1} + (1-\pi_2)\lambda_2 e^{-\lambda_2 x_1}}{0.0426}$$

$$= \frac{((0.6)(2)e^{-6} + (1-0.6)(3)e^{-9})0.6}{0.0426} = 0.049$$