FOUNDATIONS OF MACHINE LEARNING (AI2000)
ASSIGNMENT 4

MURARISETTY ADHVIK MANI SAI
(AI20BTECH11015)

NOVEMBER 2021

---

1 Non-Uniform Weights in Linear Regression: (6 marks)

You are given a dataset in which the data points are denoted by $(x_n, t_n), n = 1, ..., N$. Each data point is associated with a non-negative weighting factor $g_n > 0$. The error function is thus modified to:

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N} g_n \left(t_n - w^T \Phi(x_n)\right)^2 \tag{1}$$

Solution: Given error function as:

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N} g_n \left(t_n - w^T \Phi(x_n)\right)^2 \tag{2}$$

Let us consider,

$$G = \begin{bmatrix} \sqrt{g_1} \\ \sqrt{g_2} \\ . \\ . \\ . \\ \sqrt{g_n} \end{bmatrix}, T = \begin{bmatrix} t_1 \\ t_2 \\ . \\ . \\ . \\ t_n \end{bmatrix} \text{ and } \Phi = \begin{bmatrix} \Phi(x_1)^T \\ \Phi(x_2)^T \\ . \\ . \\ . \\ \Phi(x_n)^T \end{bmatrix} \tag{3}$$

(a) (3 marks) Find an expression for the solution w^* that minimizes the above error function.

Solution: We can rewrite the above error function as,

$$E_D(w) = \frac{1}{2}\|G^T(T - \Phi\vec{w})\|^2 \tag{4}$$

As the above function is a convex function, to get minima we have equate its derivative wrt $\vec{w}$ to zero.

$$\frac{\partial E_D(w)}{\partial w} = \frac{1}{2}\frac{\partial \|G^T(T - \Phi\vec{w})\|^2}{\partial w} \tag{5}$$

$$= \frac{1}{2}\frac{\partial}{\partial w}\left[(G^T(T - \Phi\vec{w}))^T(G^T(T - \Phi\vec{w}))\right] \tag{6}$$

$$= \frac{1}{2}\frac{\partial}{\partial w}\left[T^T GG^T T - 2w^T\Phi^T GG^T T + w^T\Phi^T GG^T \Phi w\right] \tag{7}$$

$$\frac{\partial E_D(w)}{\partial w} = \Phi^T GG^T \Phi w - \Phi^T GG^T T \tag{8}$$

For minima equate $\frac{\partial E_D(w)}{\partial w}$ to zero.

$$\frac{\partial E_D(w)}{\partial w} = 0 \tag{9}$$

$$\implies \Phi^T G G^T \Phi w^* = \Phi^T G G^T T \tag{10}$$

$$\implies w^* = \left(\Phi^T G G^T \Phi\right)^{-1} \Phi^T G G^T T \tag{11}$$

$$\therefore w^* = \left(\Phi^T G G^T \Phi\right)^{-1} \Phi^T G G^T T$$

(b) (3 marks) Give two alternative interpretations of the above weighted sum-of-squares error function in terms of: (i) data-dependent noise variance and (ii) replicated data points.

Solution: Alternate interpretations for the error function:

(i) Data-dependent noise variance: If the output of the data has gaussian noise:

$$t_i \sim \vec{w} x_i + \mathbf{N}(0, \sigma^2) \tag{12}$$

$$\implies t_i \sim \mathbf{N}(\vec{w} x_i, \sigma_i^2) \tag{13}$$

Maximising the log likelihood is same as minimising the negative log likelihood:

$$argmax_\theta \ log\left(L(\theta|x)\right) = argmax_{(\mu,\sigma_i^2)} \ \log\left(\prod P_{(\mu,\sigma_i^2)}(x_i)\right) \tag{14}$$

$$= argmax_{(\mu,\sigma_i^2)} \ \sum_{i=1}^{n} \left[-\frac{1}{2}\log(2\pi\sigma_i^2) - \frac{(x_i-\mu)^2}{2\sigma_i^2}\right] \tag{15}$$

$$= argmax_{(\mu,\sigma_i^2)} \ g_i(\mu, \sigma_i^2) \tag{16}$$

Maximise μ and σ_i^2,

$$\mu = \frac{\sum x_i}{n} \tag{17}$$

$$\sigma_i^2 = \frac{\sum (x_i - \mu)^2}{n} \tag{18}$$

$$\implies argmax_\theta \ log\left(L(\theta|x)\right) = \sum_{i=1}^{n} \frac{\left(\vec{w} x_i - t_i\right)^2}{2\sigma_i^2} \tag{19}$$

$\therefore$ By setting $\frac{1}{\sigma_i^2}$ as g_i will results in the given expression.

(ii) Replicated data points:
If we replicate our data such that ith element is replicated g_i times then the error function will be,

$$error\_function = \frac{1}{2}\sum_{n=1}^{N} g_n \left(t_n - w^T \Phi(x_n)\right)^2 \tag{20}$$

2 Bayes Optimal Classifier: (2 marks)

Let there be 5 hypotheses h_1 through h_5 that could guide a robot to move either Forward(F) or Left(L) or Right(R):

Compute the MAP estimate and Bayes optimal estimate using the data provided in the table. Are they the same? Justify your answer.

$P(h_i\|D)$	$P(F\|h_i)$	$\|P(L\|h_i)$	$P(R\|h_i)$
0.4	1	0	0
0.2	0	1	0
0.1	0	0	1
0.1	0	1	0
0.2	0	1	0

Solution: **Bayes optimal estimate:**

$$P(F) = \sum P(F|h_i)P(h_i|D) \tag{21}$$
$$= 1(0.4) + 0(0.2) + 0(0.1) + 0(0.1) + 0(0.2) \tag{22}$$
$$\implies P(F) = 0.4 \tag{23}$$

$$P(L) = \sum P(L|h_i)P(h_i|D) \tag{24}$$
$$= 0(0.4) + 1(0.2) + 0(0.1) + 1(0.1) + 1(0.2) \tag{25}$$
$$\implies P(L) = 0.5 \tag{26}$$

$$P(R) = \sum P(R|h_i)P(h_i|D) \tag{27}$$
$$= 0(0.4) + 0(0.2) + 1(0.1) + 0(0.1) + 0(0.2) \tag{28}$$
$$\implies P(R) = 0.1 \tag{29}$$

$\implies$ Bayes optimal classifier suggest to move Left.

MAp estimate:

$$h_{MAP} = argmax_{h \in H} P(h|D) \tag{30}$$

By observing in the table, and using the above definition, As for h_1, $P(h_i|D) is max$. So, The MAP estimate suggest to move Forward as only $P(F|h_i) is 1$.

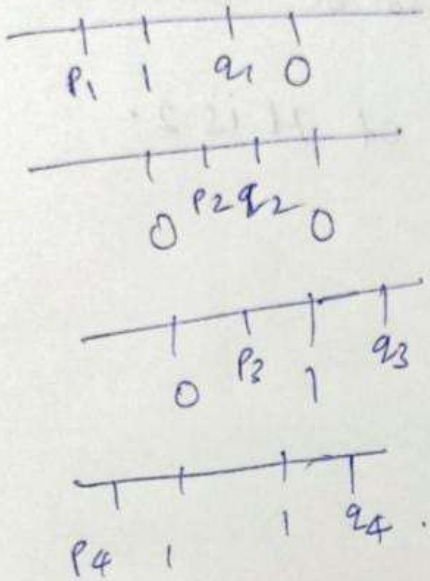
$\therefore$ They are **not same** as **bayes optimal classifier** suggest to move **Left** whereas **MAP** is suggesting to move **Forward**.

3 VC-Dimension: (2 marks)

Consider a data setup of one-dimensional data $\in \mathbf{R}^1$ where the hypothesis space **H** is parametrized by $\{p, q\}$ where x is classified as 1 iff $p < x < q$. Find the VC-dimension of **H**.

3)

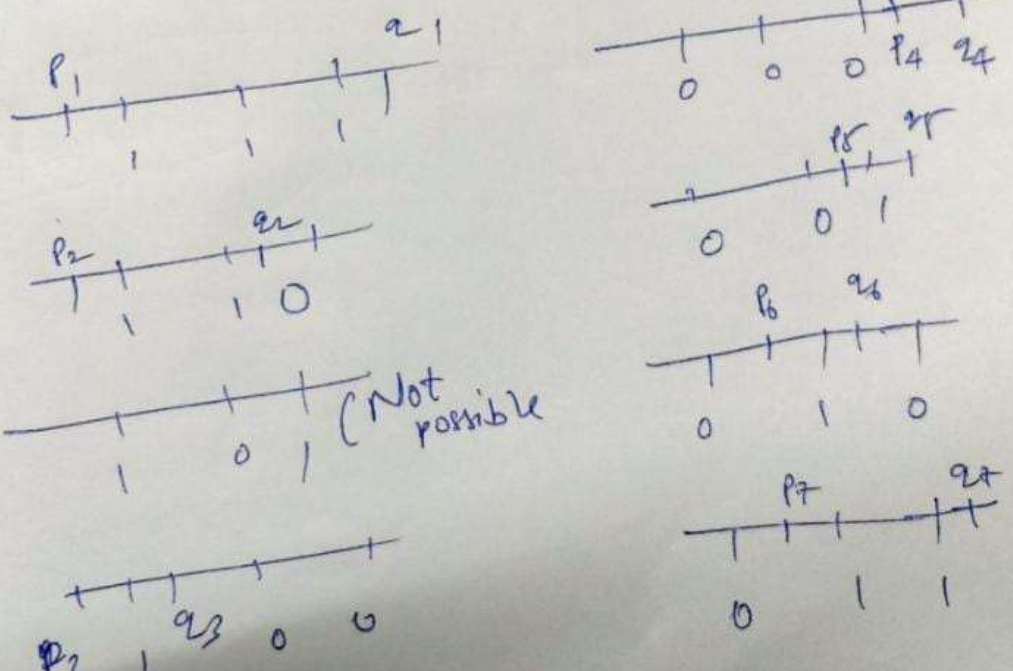
Sol:- Consider 2 points in $\mathbb{R}^1$, The no. of possible labellings are 4.

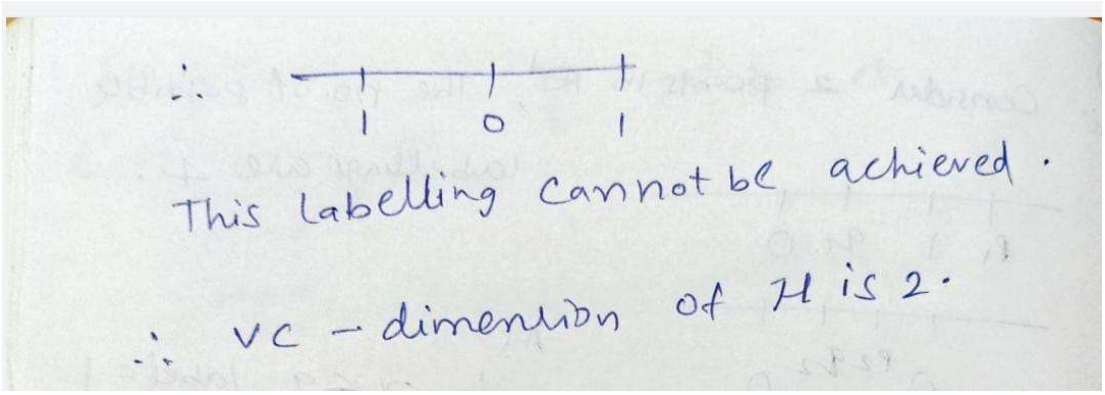


As
if $p < x < q$, label = 1
else, label = 0.

∴ All possible labellings are classified by hypothesis Space H. ∴ VC dimension is 2 or more

Now consider 3 points in $\mathbb{R}^1$, no. of possible labellings are 8.





4 Regularizer: (4 marks)

Given D-dimensional data $x = [x_1, x_2, .., x_D]$, consider a linear model of the form:

$$y(x, w) = w_0 + \sum_{k=1}^{D} w_k x_k \tag{31}$$

Now, for N such data samples with their corresponding labels $(x_i, t_i), i = 1, 2, .., N$, the sum-of-squares error (or mean-squared-error) function is given by:

$$E(w) = \frac{1}{2} \sum_{i=1}^{N} (y(x_i, w) - t_i)^2 \tag{32}$$

Now, suppose that Gaussian noise ϵ_k $\mathbf{N}(0, \sigma^2)$ (i.e. zero mean and variance σ^2) is added independently to each of the input variables x_k. Find a relation between: minimizing the above sum-of-squares error averaged over the noisy data, and minimizing the standard sum- of-squares error (averaged over noise-free input data) with a $\mathbf{L}_2$ weight-decay regularization term, in which the bias parameter w0 is omitted from the regularizer.

Solution:

4) $y(\underline{x}, \underline{w}) = w_0 + \sum_{k=1}^{D} w_k x_k$

$$E(\underline{w}) = \frac{1}{2} \sum_{i=1}^{N} (y(\underline{x}_i, \underline{w}) - t_i)^2.$$

On adding gaussian noise to the input variable x_k, then

$$y'(\underline{x}, \underline{w}) = w_0 + \sum_{k=1}^{D} w_k (x_k + \varepsilon_k)$$

$$y'(\underline{x}_i, \underline{w}) = w_0 + \sum_{k=1}^{D} w_k (x_{ik} + \varepsilon_{ik})$$

$$= y(x_i, w) + \sum_{k=1}^{D} w_k \varepsilon_{ik}.$$

Now, $$E'(\underline{w}) = \frac{1}{2} \sum_{i=1}^{N} \left(y(x_i, w) - t_i \right)^2 + 2(y(x_i, w) - t_i)(\Sigma w_k \varepsilon_{ik}) + (\Sigma w_k \varepsilon_{ik})^2 \Big)$$

$$E(E'(\underline{w})) = \frac{1}{2} \sum_{i=1}^{N} (y(x_i, w) - t_i)^2 + 2(y(x_i, \underline{w}) - t_i)(\Sigma w_k E[\varepsilon_{ik}]) + E\left[(\Sigma w_k \varepsilon_{ik})^2\right] \Big)$$

Let us assume,

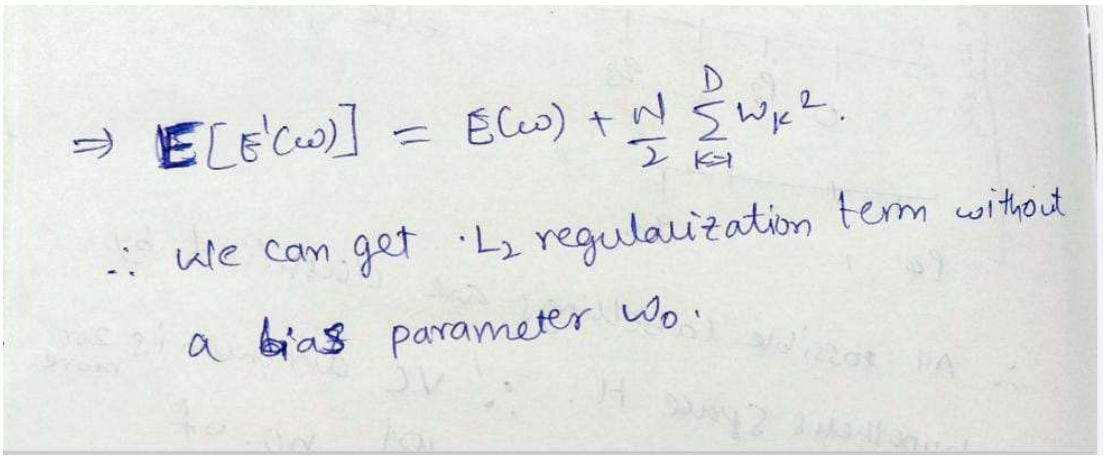
$$E[\varepsilon_i] = 0. \quad \& \quad E[\varepsilon_i \varepsilon_j] = \begin{cases} \sigma^2, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

then we will get,

$$E\left[(\Sigma w_k \varepsilon_{ik})^2\right] = E\left[\sum_{k=1}^{D} \sum_{k'=1}^{D} w_k w_{k'} (\varepsilon_{ik} \varepsilon_{ik'})\right]$$

$$= \sum_{k=1}^{D} \sum_{k'=1}^{D} w_k w_{k'} E[\varepsilon_{ik} \varepsilon_{ik'}]$$

$$= \sum_{k=1}^{D} w_k^2.$$



5 Logistic Regression: (7 marks)

Weights after one iteration are: [-1.01899756 1.53210518 0.51181202]
Final weights: [-18.07774125 26.47820683 6.58909946]
(i) Final logistic model $P(\hat{y} = 1|x_1, x_2)$ and its cross entropy function

$$\vec{w} = [-18.0777412526.478206836.58909946]$$

$$P(\hat{y} = 1|x_1, x_2) = \frac{1}{1 + e^{-(-18.07774125+26.47820683x_1+6.58909946x_2)}}$$

$$P(\hat{y} = 0|x_1, x_2) = 1 - \frac{1}{1 + e^{-(-18.07774125+26.47820683x_1+6.58909946x_2)}}$$

$$J(\vec{w}) = \sum_i y_i log(P(\hat{y}_i = 1|\vec{x}_i)) + (1 - y_i)log(P(\hat{y}_i = 0|\vec{x}_i))$$

where

$$\vec{x} = [1, x_1, x_2]$$

(ii) Use gradient descent to update $\theta_0, \theta_1, \theta_2$ for one iteration. Write down the updated logistic regression model.
After one iteration, weights:

$$\vec{w} = [-1.01899756, 1.53210518, 0.51181202]$$

Logistic regression model:

$$P(\hat{y} = 1|x_1, x_2) = \frac{1}{1 + e^{-(-1.01899756+1.53210518x_1+0.51181202x_2)}}$$

$$P(\hat{y} = 0|x_1, x_2) = 1 - \frac{1}{1 + e^{-(-1.01899756+1.53210518x_1+0.51181202x_2)}}$$

$$J(\vec{w}) = \sum_i y_i log(P(\hat{y}_i = 1|\vec{x}_i)) + (1 - y_i)log(P(\hat{y}_i = 0|\vec{x}_i))$$

where

$$\vec{x} = [1, x_1, x_2]$$

(iii) At convergence of gradient descent, use the model to make predictions for all the samples in the test dataset. Calculate and report the accuracy, precision and recall to evaluate this model.

$$Prediction : [110111]$$

$$Actual : [000111]$$

Accuracy : 0.6666666666666666

Precision : 0.6

recall : 1.0

6 Kaggle - Taxi Fare Prediction: (9 marks)

Submission and Description	Private Score	Public Score	Use for Final Score
sub2.csv 25 minutes ago by AI20BTECH11015 add submission details	3.50645	3.50645	☐
sub1.csv 25 minutes ago by AI20BTECH11015 add submission details	3.92551	3.92551	☐
sub0.csv 25 minutes ago by AI20BTECH11015 add submission details	3.47243	3.47243	☐

I have used LGBM, XGBRegressor and random forest.
XGBRegressor and random forest performed well on the test data.
Brief description:
Best Model:
RandomForestRegressor($n\_estimators = 1000$)
Ensemble model: Bagging.
Trains 1000 trees, employs bootstrapping, uses only sqrt(features) for training the trees.
Second Best Model:
XGBRegressor($n\_estimators = 200, learning\_rate = 0.23, max\_depth = 5, random\_state = 0$)
Ensemble model:Boosting.
Trains 1000 trees, employs bootstrapping, uses only sqrt(features) for training the trees.
For comparison sake, all models have been trained on the same dataset size(2e5).
XGBRegressor and Random Forest being an ensemble methods, trains a multiple weak learners, but can guarantee reasonably good accuracy. If we increase dataset size, the run time increases but accuracy will also increase.