Neural Network Assignment 03 (Tutor Redion)

Priyanka Upadhyay (2581714) - s8prupad@stud.uni-saarland.de
Gopal Bhattrai (7013547) - gobh00001@stud.uni-saarland.de

1.1 $Y_i = B_0 + B_1 X_{i1} + B_2 \ln(X_{i2}) + B_3 X_{i3}$

$Y_i = 10 + 10 X_{i1} + 0.5 \ln(X_{i2}) - 5 X_{i3}$ — (1)

(i) → X_1 = changes from 1 unit

$Y_{i+1} = 10 + 10 X_{i+1} + 0.5 \ln(X_{i2}) - 5 X_{i3}$

$Y_{i+1} - Y_i = [10 + 10 X_{i+1} + 0.5 \ln(X_{i2}) - 5 X_{i3}]$

$- [10 + 10 X_{i1} + 0.5 \ln(X_{i2}) - 5 X_{i3}]$

$\Rightarrow 10 X_{i+1} - 10 X_{i1}$

$[Y_{i+1} - Y_i] \Rightarrow 10 (X_{i+1} - X_{i1})$

So $X_{i+1} - X_i = 1$

$\downarrow$

$Y_{i+1} - Y_i = 10 X_1 = 10$

So if 1 change in X gives 100% change in Y So Now 10 unit change in $(Y_{i+1} - Y_i)$ gives total $10 \times 100\% = 1000\%$ total change in Response.

= TRUE

(ii) X_2 = 1 unit change

$[Y_{i+1} - Y_i] = 0.5 [\ln(X_{i2+1}) - \ln(X_{i2})]$

$= 0.5 \ln [X_{i2+1} - X_{i2}]$

So 0.5 x 100 % = 50 % total change in Y

= TRUE

(iii) 100 % change in x_{i2} → x_{i2} → $2x_{i2}$

So $y_{i+1} - y_i = 0.5 \ln[2x_{i2} - x_{i2}]$

$= 0.5 \ln[x_{i2}]$

⇒ 0.5 x 100 % = 50 % total change in Y

= TRUE

(iv) Higher debt implies lower future stock values

as Y and x_{i3} has inverse relation because of the (-ve) coefficient, we can say that when x_{i3} has the largest value, Y would become very small.

= TRUE

(v) As per the machine Learning → we know that Bias term gives us an educated guess of the Responce when we dont Have any knowledge about Input variables.

So when I/P = 0

$Y = B_0 = 10$ → without any input we still Have some Responce value.

1.2 Part b $\quad J(w) = MSE_{train} + \lambda w^T w$

w = weights , λ = Regularization Parameter

Output : $W = \left[(x^{Train})^T x^{Train} + \lambda I \right]^{-1} x^{Train^T} y^{Tr}$

Writing - $X = x^{Train}$, $Y = y^{train}$

we minimize the distance b/w $J(w)$

$$\therefore J(w) = \frac{1}{2} \| xw - y \|^2 + \lambda \times \frac{1}{2} \| w \|^2$$

OR $\quad J(w) = \frac{1}{2} (xw - y)^T (xw - y) + \frac{\lambda}{2} w^T w$

$$\Rightarrow \frac{1}{2} (x^T w^T - y^T)(xw - y) + \frac{\lambda}{2} w^T w$$

$$\Rightarrow \frac{1}{2} \left[x^T x w^T w - \underbrace{x^T w^T y - y^T x w}_{y^T x = x^T y} + y^T y \right] + \frac{\lambda}{2} w^T w$$

$$\Rightarrow \frac{1}{2} \left[x^T x w^T w - 2 w^T x^T y + \underbrace{y^T y}_{X^T X} \right] + \frac{\lambda}{2} w^T w$$

$$\Rightarrow w^T \times \frac{1}{2} \left[2 x^T x w - 2 x^T y \right] + \frac{\lambda}{2} w^T w$$

for closed form

So ~~$\frac{\partial [J(w)]}{\partial w} = 0$~~ put $\frac{\partial J(w)}{\partial w} = 0$

~~(Resulting matrix is full Ran~~

$$\frac{w^T}{2} \left[2 x^T x w - 2 x^T y \right] + \frac{w^T}{2} \lambda w = 0$$

$$\Rightarrow 2 x^T x w - 2 x^T y + \lambda w = 0$$

$$\Rightarrow w (2 x^T x + \lambda) - x^T y = 0$$

$$w = \frac{x^T y}{[x^T x + \lambda]}$$

$$w = [x^T x + \lambda I]^{-1} x^T y$$

putting back again $X = X^{train}$, $y = y^{train}$

Weights $$w = \left[X^{Train^T} X^{Train} + \lambda I \right]^{-1} X^{Train^T} Y^{Train}$$

Answer

[Reference! - [redacted]
Ridge Regression, Wessel N. van → Page 8

Problem - 4 Logistic Regression

1 → Logistic Regression peforms when our Responce variable is categorical, predictors variable can by anything continuous or categorical. Since Linear Regression work on a very Strong assumptions which is that the outcome variable is continuous & using Linear Regression such case violates the strong assumption & therefore can not be used.

2 → Logistic Regression is supervised learning method.

3 → We can not use because then cost function would become non-convex function which is not our desired goal.

5 → for a simple classifier, the outcome Y has a category 0 or 1 / yes or No but nothing in the middle of these values. But this is Not true of all various Input values X. So in the Logistic Regression the output can be seen as probability since it matches to training distribution.

4 → Logistic Regression is trained by feeding on input data and a binary class to which this data belongs

Problem-5 Eigen Decomposition:-

$$M = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

→ Symmetric $X = X^T$ and M is a symmetric matrix. It implies that matrix is orthogonal so M^{-1} could be replaced by M^T which is much easier than calculating M^{-1}

→ Yes, M is Singular. It implies that matrix is diagonalizable

→ As we have also solved it in the Assignment 02:-

$$MV = \lambda V$$

$$[M-\lambda]\, v = 0$$

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0$$

$$\Rightarrow |M-\lambda| = 0$$

$$= \begin{bmatrix} 1-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 1-\lambda \end{bmatrix} = 0$$

$$\Rightarrow (1-\lambda)\left[(2-\lambda)(1-\lambda) - 1\right] + (-1)\left[(1-\lambda)\right] + 0$$

$$\Rightarrow (1-\lambda)(2-\lambda)(1-\lambda) - (1-\lambda) - (1+\lambda) = 0$$

$$\Rightarrow \lambda = 3, 1, 0$$

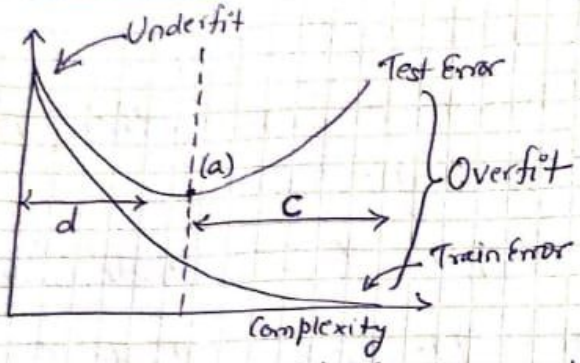
$$\lambda_1 = 3, \quad v_1 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$\lambda_2 = 1 \quad = \quad v_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_3 = 0, \quad v_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Ans

2. ⇒



* Point (a) represents the ideal point where we should operate ML algorithm Because after that test error starts to go Up. and train Error down ⟶ Overfitting.

• After the Point (a), the range shown by C is Overfitting. the model became highly complex, we should reduce down its complexity.

Like ⇒ Minimize the degree of Polynomial.
Manipulate Regularization Const. (i.e. increase the value of it)

• The Point where both train and test error is Very high the model is Underfitting, because model is not Complex enough to grab the pattern of Underlying dataset.

(b) Ridge Regression ⇒ $\Rightarrow \|\theta^T X - b\|_2^2 + \lambda \|\theta\|_2^2$

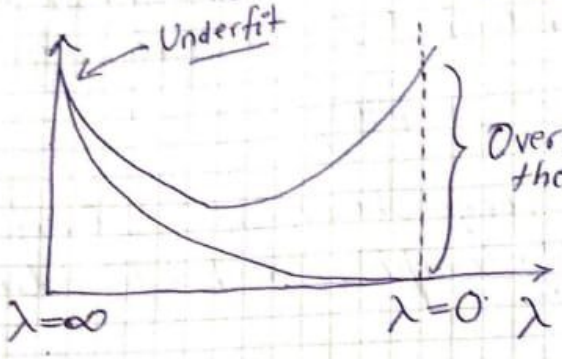
Here ⇒ $\theta \in \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$, $X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \in \mathbb{R}^{p+1}$

If I increase λ ⇒

Our Job is to minimize the loss f^n above, If I increase λ, then θ has to be Small, then and Only then the loss f^n will go down.

If λ = Very high $\rightsquigarrow \theta \approx 0$, Model Underfits

λ = Very Small $\rightsquigarrow$ No restriction on θ, Model Overfits



2.3 ⟹ here K = #Instances. i.e. Training on n-1 instances and predicting on 1 instance.

So we trained on n-1 instances and then we calculated the score on one remaining instance we got MSE. Here I assume that error 35 we got was on validation points. Not the average of $[RMSE_1, RMSE_2 \ldots RMSE_n]$. So when I reshuffle and again do LOOCV test then two Case might happen ⇒

1. We again selected the same Validation point in that case I'll get 35 again.
2. In the Case when I selected another

Point as the Validation point.:

Since we used n-1 instances for training, and we assume it captures almost all of the Variance of the data. The MSE that we'll get on the Validation point will be more likely close to 35. (Assuming its not an Outlier.)

3. $L = \underset{\lambda}{\operatorname{argmax}}\ P(x)$

$L = \log \underset{\lambda}{\operatorname{argmax}}\ \frac{e^{-\lambda}\lambda^{x}}{x!}$

$L = \log \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$ Because R.V are i.i.d.

$L = \sum_{i=1}^{n} \log(e^{-\lambda}\lambda^{x_i}) - \log(x_i!)$

$L = \sum_{i=1}^{n} \log(e^{-\lambda}) + \log(\lambda^{x_i}) - \log(x_i!)$

$L = \sum_{i=1}^{n} -\lambda + x_i \cdot \log\lambda - \log(x_i!)$

$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{n}\left[-1 + \frac{x_i}{\lambda}\right] = 0$

$-n + \frac{1}{\lambda}\cdot\sum_{i=1}^{n} x_i = 0$

$\frac{1}{\lambda}\sum_{i=1}^{n} x_i = n$ Hence,

$$\left[\lambda = \frac{1}{n}\sum_{i=1}^{n} x_i\right]$$

(b) Since we now assumed $X_i = 0$ or $X_i > 0$

So by Part (a) $\Rightarrow$ $\lambda = \frac{1}{n} \sum_{i=1}^{n} x_i$ $\quad \because x_i > 0$ or $x_i = 0$

If I want to Interpret λ in terms of Y's, I can $\Rightarrow$

$$\lambda = \frac{1}{n} \sum_{i=1}^{n} y_i \quad ; \quad y_i = 1 \rightarrow x_i > 0$$
$$y_i = 0 \rightsquigarrow x_i = 0$$

So Say All the $X_i > 0 \ \forall i$ then

$$[\lambda = 1]$$

if all $X_i = 0 \ \forall i$: $\boxed{\lambda = 0}$

Because while doing Maximum Likelihood Estimation we took all the random variables into account. and now by getting the final expression, we can find value of λ based on cond^n on Random Variables.