CS5063: Foundations of Machine Learning

Assignment 4

Submitted By:
Vishal Singh Yadav – CS20MTECH01001

1.

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

(a) Let G be a diagonal matrix containing weighting coefficients.

$\Rightarrow G = diag(g_1, g_2, \dots, g_N)$.

We can now rewrite error function as,

$$E_D(w) = \frac{1}{2}(\phi w - t)^T G (\phi w - t)$$

$$= \frac{1}{2}(w^T\phi^T G\phi w - w^T\phi^T G t - t^T G\phi w + t^T G t)$$

$$= \frac{1}{2}(w^T\phi^T G\phi w - 2t^T G\phi w + t^T G t)$$

Taking gradient of $E_D(w)$

$$\Rightarrow \nabla E_D(w) = \phi^T G\phi w - t^T G\phi.$$

$$w = (\phi^T G\phi)^{-1} * t^T G\phi.$$

$$\Rightarrow \quad w = (\phi^T G\phi)^{-1}\phi^T G t.$$

$\therefore w^* = \dfrac{\phi^T G t}{\phi^T G\phi}$ minimizes the error func $E_D(w)$.

<b> <i> As data dependent noise variance.

Put $\frac{\partial}{\partial w} E_D(w) = 0$.

$$\Rightarrow -\sum_{n=1}^{N} g_n \left[E_n - w^T \phi(g_n)\right] \phi x_n = 0.$$

$$\Rightarrow \sum_{n=1}^{N} g_n t_n \phi(x_n) = \sum_{n=1}^{N} g_n \phi(x_n) \phi(x_n)^T w$$

$$\Rightarrow w = \left(\sum_{n=2}^{N} r_n \phi(x_n) \phi(x_n)^T\right)^{-1} \left(\sum_{n=1}^{N} g_n t_n \phi(x_n)\right)$$

<ii> As replicated data

the above form of.

$w = (\phi^T G \phi)^{-2} \phi^T G t$. is already in

replicated data form.

2.

$\max(P(h_i|D)) = 0.4$

$\operatorname{argmax}(P(h_i|D)) = h_2$

$P(F|h_2)) = 1$

$P(R|h_2) = 0.$

$P(L|h_2) = 0.$

Using MAP estimate, the robot ~~will~~ ~~move~~ should go forward.

Bayes optimal classifier:-

$$h_{BO} \equiv \arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D)$$

$$\sum_{h_i \in H} P(F \mid h_i) P(h_i \mid D) = 1 \times 0.4 = 0.4$$

$$\sum_{h_i \in H} P(R \mid h_i) P(h_i \mid D) = 0.2 + 0.2 + 0.2 = 0.5$$

$$\sum_{h_i \in H} P(L \mid h_i) P(h_i \mid D) = 0.2$$

Using Bayes optimal classifier, the robot should go left.

3.

The VC dimension for $\mathbb{R}^d$ data is given by $d+1$

$\therefore$ For one dimensional data $\in \mathbb{R}^1$, the VC dimension of H is 2.

4.

5.

a. Code attached

b.

<i> The logistic function is.

$$P(\hat{y}=1 \mid x_1 x_2) = \frac{1}{1+e^{-(-1+1.5x_1+0.5x_2)}}$$

Its cross entropy error function is

$$-(y \times \log(\hat{y})) - ((1-y) \times \log(1-\hat{y}))$$

<ii> The updated model is

$$P(\hat{y}=1 \mid x_1 x_2) = \frac{1}{1+e^{-(-1.003+1.540x_1+0.523x_2)}}$$

<iii> Accuracy = 0.5

Precision = 0.5

Recall = 1.0.

6. The top two score I got was 3.92887 and 3.89396. I got this using boosting algorithms, LightGBM and Cat boost respectively. I also tried other methods available in Sklearn package but none of them was able to get such good results with results averaging around ~5.4
LightGBM and CatBoost are boosting algorithms for tree based models. These models tend to provide even better results than XGBoost which is itself an improvement over Random Forest method. I have not done any hyperparameter tuning for both methods or used any ensemble method. Doing so might give even better results.
I also tried linear regression and ridge regression. These methods were were going to fail as the dataset is not simple and linearly separable. These methods form a large number of trees and provide results using weighted internal trees. Hence these methods tend to give better results.