Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

QUESTION 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

In the question we are given a class wise loss function as a function of $\mathbf{w}_c$ and $\mathbf{M}_c$ parameters for each class, and we need to solve the following optimization problem for each class:

$$(\hat{\mathbf{w}}_c, \hat{\mathbf{M}}_c) = arg \min_{(\mathbf{w}_c, \mathbf{M}_c)} \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \mathbf{w}_c)^T \mathbf{M}_c (\mathbf{x}_c - \mathbf{w}_c) - \log |\mathbf{M}_c|$$

We can call the above loss function as $L(\mathbf{w}_c, \mathbf{M}_c)$ for some class c. In order to minimize this loss, we can write following equations:

$$\frac{\partial}{\partial \mathbf{w}_c}(L(\mathbf{w}_c, \mathbf{M}_c)) = 0 \tag{i}$$

$$\frac{\partial}{\partial \mathbf{M}_c}(L(\mathbf{w}_c, \mathbf{M}_c)) = 0 \tag{ii}$$

Solving the i further,

$$\implies (-1). \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{M}_c + \mathbf{M}_c^T)(\mathbf{x}_c - \mathbf{w}_c) = 0$$

$$\implies (\mathbf{M}_c + \mathbf{M}_c^T)(\mathbf{w}_c - \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c}(\mathbf{x}_c)) = 0$$

$$\implies (\mathbf{M}_c + \mathbf{M}_c^T)(\mathbf{w}_c - \boldsymbol{\mu}_c) = 0 \tag{iii}$$

{where $\boldsymbol{\mu}_c$ is the mean of all the training inputs}

Now, performing a left matrix multiplication with $(\mathbf{w}_c - \boldsymbol{\mu}_c)^T$ in iii,

$$\implies (\mathbf{w}_c - \boldsymbol{\mu}_c)^T (\mathbf{M}_c + \mathbf{M}_c^T)(\mathbf{w}_c - \boldsymbol{\mu}_c) = 0$$

Now, since $\mathbf{M}_c$ is a positive definite matrix(PD), we can obtain that $(\mathbf{M}_c + \mathbf{M}_c^T)$ is also a positive definite matrix. Also, we know that for any positive definitive matrix A, $x^T A x$ can not be 0 until x is equal to 0.

Following from the above analogy, we have

$$(\mathbf{w}_c - \boldsymbol{\mu}_c) = 0$$

$$\implies \boxed{\mathbf{w}_c = \boldsymbol{\mu}_c} \tag{iv}$$

Now, solving the second differential equation ii,

$$\sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \mathbf{w}_c)(\mathbf{x}_c - \mathbf{w}_c)^T - (\mathbf{M}_c^{-1})^T = 0$$

$$\sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \boldsymbol{\mu}_c)(\mathbf{x}_c - \boldsymbol{\mu}_c)^T - (\mathbf{M}_c^{-1})^T = 0 \qquad (\text{ } \mathbf{w}_c \text{ from iv})$$

$$\implies (\mathbf{M}_c^{-1})^T = \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \boldsymbol{\mu}_c)(\mathbf{x}_c - \boldsymbol{\mu}_c)^T$$

$$\boxed{\mathbf{M}_c = \frac{I_D}{\sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \boldsymbol{\mu}_c)(\mathbf{x}_c - \boldsymbol{\mu}_c)^T}} \tag{v}$$

{ Where $\boldsymbol{\mu}_c$ is the mean of training inputs of class c }

Hence, from *iv* and *v* we have the optimal value of $\mathbf{w}_c$ and $\mathbf{M}_c$.
Special case of $\mathbf{M}_c$ being an identity matrix: In this case the value of $\mathbf{w}_c$ comes to

$$(\hat{\mathbf{w}}_c) = arg \min_{\mathbf{w}_c} \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} (\mathbf{x}_c - \mathbf{w}_c)^T (\mathbf{x}_c - \mathbf{w}_c)$$

$$(\hat{\mathbf{w}}_c) = arg \min_{\mathbf{w}_c} \sum_{\mathbf{x}_c : y_n = c} \frac{1}{N_c} \|\mathbf{x}_c - \mathbf{w}_c\|^2$$

Here, $\mathbf{w}_c$ locates to minimum distance of all the training inputs of class c, which means that it is acting as the prototype of the class c. Hence, this model reduces to prototype base model in this special case.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

QUESTION 2

Given, infinite amount data with Bayes optimal error rate is zero we can say that there will be one such classifier which will be consistent. Now, for the case of one-nearest-neighbour algorithm, once it is trained on infinite amount of data with Bayes opitmal error. As k-nearest-neighbour create boundary once trained, it will also have a boundary which will classify the training inputs correctly. Now, we can say that the test point will be also in the training data since it is infinite and hence it will be consistent in that test point as well.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

QUESTION 3

A good criterion to choose a feature to split on for regression is the **reduction in variance**. This will show how much the variance of the real-valued labels decreases after splitting on a feature. Since, we want to reduce the variance of the labels to get more and more homogenity. Therefore, a feature that reduces the variance of the labels at each node creates more homogeneous subsets of data.

The reduction in variance is calculated as follows:

1. Let V be the variance of the labels at the parent node before splitting.
2. Let V_L and V_R be the variances of the labels at the left and right child nodes after splitting on a feature F.
3. Let N be the total number of examples at the parent node, and N_L and N_R be the number of examples at the left and right child nodes respectively.
4. Then, the reduction in variance (RIV) due to splitting on F is given by:

$$RIV = V - \frac{N_L}{N} V_L - \frac{N_R}{N} V_R$$

5. The feature that maximizes the (RIV) is chosen as the best splitting feature for regression.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

QUESTION 4

Let us asssume that $f(\mathbf{x}_*)$ is the prediction at a test input $\mathbf{x}_*$ and value of $\hat{w}$ is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. We can write $f(\mathbf{x}_*)$ as,

$$f(\mathbf{x}_*) = \mathbf{x}_*^T\hat{\mathbf{w}}$$

$$\implies f(\mathbf{x}_*) = \mathbf{x}_*^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \qquad \text{(value of } \hat{\mathbf{w}}\text{)}$$

$$\implies f(\mathbf{x}_*) = (\mathbf{X}((\mathbf{X}^T\mathbf{X})^{-1})^T\mathbf{x}_*)^T\mathbf{y}$$

$\{$ taking transpose out of the bracket $\}$

$$\implies f(\mathbf{x}_*) = \mathbf{W}^T\mathbf{y}$$

$\{$ where, $\mathbf{W}$ is an $N \times 1$ shaped matrix representing $\mathbf{X}((\mathbf{X}^T\mathbf{X})^{-1})^T\mathbf{x}_*$ $\}$

$$\implies f(\mathbf{x}_*) = \sum_{n=1}^{N} w_n y_n$$

Here, $\mathbf{W}$ is made up of w_n's and $\mathbf{y}$ is made of y_n's, Let's explore $\mathbf{W}$ to get a feel of weights w_n's.

$$\mathbf{W} = \mathbf{X}((\mathbf{X}^T\mathbf{X})^{-1})^T\mathbf{x}_*$$

$$\implies \mathbf{W} = \mathbf{X}((\mathbf{X}^T\mathbf{X})^T)^{-1}\mathbf{x}_*$$

$\{$ taking transpose inside the bracket $\}$

Exploring each part of matrix multiplication of right hand side,

$$\implies \frac{\mathbf{W}}{N \times 1} = \frac{\mathbf{X}}{N \times D}\frac{(\mathbf{X}^T\mathbf{X})^{-1}}{D \times D}\frac{\mathbf{x}_*}{D \times 1}$$

So, here the term $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}$ can be interpreted as matrix of processed training inputs of shape $N \times D$.

$$\mathbf{W} = \frac{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}}{N \times D}\frac{\mathbf{x}_*}{D \times 1}$$

$$\mathbf{W} = \frac{\tilde{\mathbf{X}}}{N \times D}\frac{\mathbf{x}_*}{D \times 1}$$

$\{$ where $\tilde{\mathbf{X}}$ is the contains a D dimensional process feature vector for all the N training inputs $\}$

Hence for each test input $\mathbf{x}_*$, it multiplied by $\tilde{\mathbf{X}}$ to generate a N dimensional vector which then used as weights to get the weighted sum of all the training responses.

$$\mathbf{W} = \frac{\tilde{\mathbf{X}}\mathbf{x}_*}{N \times 1}$$

$$\text{So, } \boxed{w_n = \tilde{\mathbf{x}_n}\mathbf{x}_*}$$

Differences with KNN:

- In the case of K nearest neighbours each w_n is basically the inverse distance of $\mathbf{x}_*$ from a training input $\mathbf{x}_n$, while here the weights is not the inverse of distance but some kind of weight to relation of x_* with training input x_n.
- The inverse distance in case K nearest neighbour is limited to each training input $\mathbf{x}_n$ while here, it weight corresponding to any training input is dependent on all the training inputs as we saw for the expression of $\tilde{\mathbf{X}}$.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

QUESTION 5

We have loss function $L = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\tilde{\mathbf{x}_n})^2$ where we have $\tilde{\mathbf{x}_n} = \mathbf{x}_n \circ \mathbf{m}_n$ where $\mathbf{m}_n$ is the $D \times 1$ shapes binary mask vector.

Let's calculate the expectation of loss function, $\mathbb{E}[L]$

$$\mathbb{E}[L] = \mathbb{E}[\sum_{n=1}^{N}(y_n - \mathbf{w}^T\tilde{\mathbf{x}_n})^2]$$

$$\implies \mathbb{E}[L] = \mathbb{E}[\sum_{n=1}^{N}(y_n^2 + (\mathbf{w}^T\tilde{\mathbf{x}_n})^2 - 2y_n\mathbf{w}^T\tilde{\mathbf{x}_n})]$$

{ expanding the above equation }

$$\implies \mathbb{E}[L] = \sum_{n=1}^{N}(\mathbb{E}[y_n^2] + \mathbb{E}[(\mathbf{w}^T\tilde{\mathbf{x}_n})^2] - \mathbb{E}[2y_n\mathbf{w}^T\tilde{\mathbf{x}_n}])$$

{ taking expectation inside the bracket }

Now, since y_n is not random, we have $\mathbb{E}[y_n^2] = y_n^2$

$$\mathbb{E}[L] = \sum_{n=1}^{N}(y_n^2 + \mathbb{E}[(\mathbf{w}^T\tilde{\mathbf{x}_n})^2] - 2y_n\mathbb{E}[\mathbf{w}^T\tilde{\mathbf{x}_n}]) \tag{i}$$

We know,

$$\mathbb{E}[(\mathbf{w}^T\mathbf{x}_n - (\mathbb{E}[\mathbf{w}^T\mathbf{x}_n]))^2] = \mathbf{w}^T\mathbf{M}_2\mathbf{w} \qquad \text{(from matrix cookbook)}$$

Where, $\mathbf{M}_2$ is covariance matrix of $\mathbf{x}_n$

$$\implies \mathbb{E}[(\mathbf{w}^T\mathbf{x}_n)^2] - (\mathbb{E}[\mathbf{w}^T\mathbf{x}])^2 = \mathbf{w}^T\mathbf{M}_2\mathbf{w}$$

$$\implies \mathbb{E}[(\mathbf{w}^T\mathbf{x}_n)^2] = \mathbf{w}^T\mathbf{M}_2\mathbf{w} + (\mathbb{E}[\mathbf{w}^T\mathbf{x}])^2 \tag{ii}$$

Using (ii) in (i) we have,

$$\mathbb{E}[L] = \sum_{n=1}^{N}(y_n^2 + \mathbf{w}^T\mathbf{M}_2\mathbf{w} + (\mathbb{E}[\mathbf{w}^T\tilde{\mathbf{x}_n}])^2 - 2y_n\mathbb{E}[\mathbf{w}^T\tilde{\mathbf{x}_n}])$$

Now, since d^{th} feature will be kept with probability p and we are doing that independently, we can say that $\mathbb{E}[\tilde{\mathbf{x}_n}] = p\mathbf{x}_n$

$$\text{Hence, } \mathbb{E}[\mathbf{w}^T\tilde{\mathbf{x}_n}] = p\mathbf{w}^T\mathbf{x}_n \qquad (iii)$$

Using (iii),

$$\mathbb{E}[L] = \sum_{n=1}^{N}(y_n^2 + \mathbf{w}^T\mathbf{M}_2\mathbf{w} + p^2(\mathbf{w}^T\mathbf{x}_n)^2 - 2y_n p\mathbf{w}^T\mathbf{x}_n)$$

$$\mathbb{E}[L] = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x})^2 + \mathbf{w}^T\mathbf{M_2}\mathbf{w}$$

Now, this is clearly a form of regularized loss function of form $(L(\mathbf{w}) + f(\mathbf{w}, \mathbf{x}_n))$.

Introduction to ML (CS771), Autumn 2023
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Divyansh
Roll Number: 210355
Date: September 15, 2023

QUESTION 6

- The test accuracy in the convex part is `46.89 %`
- The test accuracies in the regression part corresponding to λ's is

Value of λ	Test Accuracy(%)
0.01	58.09
0.1	59.54
1	67.39
10	73.28
20	71.68
50	65.08
100	56.47