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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION

1

Given optimization problem:

$$(\hat{w}_c, \hat{M}_c) = \underset{w_c, M_c}{\operatorname{argmin}} \sum_{x_n : y_n = c} \frac{1}{N_c} \left((x_n - w_c)^T M_c (x_n - w_c) - \log |M_c| \right)$$

Our aim is to find the optimal values of w_c and M_c.

1. Finding optimal w_c:

   To find the optimal w_c, we need to find derivative of given objective function with respect to w_c and equate it to zero. We are using the method of First Order Optimality here for calculating optimal w_c.

   Taking partial derivative of given objective function with respect to w_c we get:

$$\frac{\partial(\hat{w}_c, \hat{M}_c)}{\partial w_c} = \sum_{x_n : y_n = c} \frac{1}{N_c} \left(-2M_c(x_n - w_c)\right) - 0$$

$$0 = \sum_{x_n : y_n = c} \frac{1}{N_c} \left(-2M_c x_n + 2M_c w_c\right)$$

$$0 = \sum_{x_n : y_n = c} -2M_c x_n + \sum_{x_n : y_n = c} 2M_c w_c$$

$$\sum_{x_n : y_n = c} M_c w_c = \sum_{x_n : y_n = c} M_c x_n$$

   As M_c is independent of w_c, we consider it constant. Therefore,

$$M_c \sum_{x_n : y_n = c} w_c = M_c \sum_{x_n : y_n = c} x_n$$

$$N_c w_c = \sum_{x_n : y_n = c} x_n$$

$$\boxed{w_c = \frac{1}{N_c} \sum_{x_n : y_n = c} x_n}$$

2. Finding the optimal M_c:

   To find the optimal M_c, we need to find derivative of given objective function with respect to M_c and equate it to zero. We are using the method of First Order Optimality here for calculating optimal M_c.

Taking partial derivative of given objective function with respect to M_c we get:

$$\frac{\partial(\hat{w}_c, \hat{M}_c)}{\partial M_c} = \sum_{x_n:y_n=c} \frac{1}{N_c}((x_n - w_c)(x_n - w_c)^T) - ({M_c}^{-1})^T$$

$$({M_c}^{-1})^T = \sum_{x_n:y_n=c} \frac{1}{N_c}((x_n - w_c)(x_n - w_c)^T)$$

$$({M_c}^T)^{-1} = \sum_{x_n:y_n=c} \frac{1}{N_c}((x_n - w_c)(x_n - w_c)^T)$$

As M_c is symmetric, ${M_c}^T = M_c$. Substituting this value in equation we get,

$$(M_c)^{-1} = \sum_{x_n:y_n=c} \frac{1}{N_c}((x_n - w_c)(x_n - w_c)^T)$$

$$\boxed{M_c = \left[\sum_{x_n:y_n=c} \frac{1}{N_c}((x_n - w_c)(x_n - w_c)^T)\right]^{-1}}$$

3. Special Case when M_c is an Identity Matrix:

   Now, if M_c is an identity matrix ($M_c = I$), the model simplifies significantly:

$$(\hat{w}_c, \hat{I}_c) = \underset{w_c, I_c}{\operatorname{argmin}} \sum_{x_n:y_n=c} \frac{1}{N_c} \big((x_n - w_c)^T I_c (x_n - w_c) - \log|I_c|\big)$$

   We know that $|I_c| = 1$ and we can neglect I_c since it does not modify anything. So, the modified objective function becomes,

$$(\hat{w}_c) = \underset{w_c}{\operatorname{argmin}} \sum_{x_n:y_n=c} \frac{1}{N_c} \big((x_n - w_c)^T (x_n - w_c) - \log 1\big)$$

   We know, $\log(1) = 0$. Substituting this value, the objective function becomes,

$$(\hat{w}_c) = \underset{w_c}{\operatorname{argmin}} \sum_{x_n:y_n=c} \frac{1}{N_c} \big((x_n - w_c)^T (x_n - w_c)\big)$$

$$(\hat{w}_c) = \frac{1}{N_c} \sum_{x_n:y_n=c} ||x_n - w_c||^2$$

   This modified objective function is equivalent to mean of squared euclidean distance between x_n and w_c. w_c can be seen as mean of class c.

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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION 2

The one-nearest-neighbor algorithm will be consistent in the above mentioned noise free setting. The reason is that , 1-NN takes into account the closest neighbor with respect to distances(mostly euclidean distances are used). In noise free setting, entire training data has correct labels i.e, the optimal error rate (Bayes optimal) will approach to zero. As the Bayes optimal value is close to zero, the labels predicted using 1NN classifier for test data will be accurate since the closest neighbor to a test point will have correct label. So, the test point will also be correctly labeled.

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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION 3

To construct a Decision Tree for Regression, we prefer to split on features that divide a node such that the training points having similar output are accumulated together in a leaf node. The splitting criteria using in Decision Tree for Regression is Variance Reduction. It is somewhat similar to Information Gain but we cannot use Information Gain for regression as outputs are real valued. Using Variance Reduction, we get the training inputs which have similar outputs at a particular node.

To calculate variance reduction, we need to find a split point of a feature based on which, we perform splitting. This split point can be any value the feature can take. At each node, we perform thresholding with respect to this split point in order to split the training data set. Also, we need to find the variance reduction for each value of the feature. The value of feature for which the Variance Reduction is maximum gets selected at that particular node and with respect to that split point, the training data set gets splitted.

To calculate Variance Reduction, we calculate variance for all the training inputs at a particular node.

$$\boxed{Variance(Z) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where,

- Z can be a parent node, left node, right node.
- n is the siae of training data set.
- $\bar{y}$ is mean of all the outputs of training data set.
- y_i is the output value for a particular value of input feature value.

Variance is calculated using the formula of Mean Squared Error (MSE). Here $\bar{y}$ is the mean of outputs of all the training inputs. Similarly, we calculate variance at child nodes. After calculation of variance of child nodes and the root node, we calculate variance reduction by using the formula:

$$\boxed{\text{Variance Reduction} = \text{Var(Z)} - \Big(\frac{n_1}{n}\text{Var(left child)} + \frac{n_2}{n}\text{Var(right child)}\Big)}$$

Where,

- Var(Z) is Variance of parent node.
- Var(left child) is Variance of the left training subset.
- Var(right child) is Variance of the right training subset.
- n_1 and n_2 are training subset sizes of left training subset and right training subset.

Using this formula of variance reduction, we can obtain more "pure" splits so that we get the training points for which the true values are same.

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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION 4

My solution to problem 4

Given: $\hat{w} = (X^T X)^{-1} X^T y$. Therefore, a prediction for test input x^* can be written as:

$$y^* = \hat{w}^T x^* = (x^*)^T \hat{w}$$

Substituting the value of $\hat{w}$ we get, $y^* = (x^*)^T (X^T X)^{-1} X^T y$

Let's define a new matrix $\mathbf{M} = (X^T X)^{-1} X^T$. Dimension of M will be $D \times N$. Substituting this matrix in equation,

$$y^* = (x^*)^T \mathbf{M} y$$

If we consider $w = (x^*)^T \mathbf{M}$ then, w is a row vector of size $1 \times N$. Then the equation can be written as,

$$y^* = \mathbf{wy} = \sum_{n=1}^{N} w_n y_n$$

Where, w_n are the n_{th} index of the $1 \times N$ vector w. Knowing that X is a matrix whose rows are the N training vectors x_n, w_n can be written as:

$$w_n = x^* T (X^T X)^{-1} x_n$$

The value of w_n depends on both the input x^* and the entire training data set ranging from x_1 to x_N. This differs from the weighted k-Nearest Neighbors (kNN) method, where the individual weights solely depend on x^* and x_n. Another difference lies in the representation of w_n; in this method, they are expressed as products involving x^*, whereas in kNN, they are represented as sums in the denominator.

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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION 5

Consider the new loss function with masked inputs:

$$L_{\text{masked}}(w) = \sum_{n=1}^{N} (y_n - w^T \tilde{x}_n)^2$$

where $\tilde{x}_n = x_n \odot m_n$, and m_n is a D x 1 binary mask vector with elements $m_{nd} \sim$ Bernoulli(p). $m_{nd} = 0$ means that feature got masked with probability of (1-p) and $m_{nd} = 1$ means that feature got retained.

Substituting the value of $\tilde{x}_n$ in the function we get,

$$L_{\text{masked}}(w) = \sum_{n=1}^{N} (y_n - (w^T (x_n \odot m_n))^2$$

We now need to calculate the expected value of $L_{\text{masked}}(w)$ with respect to the random mask vectors m_n and then minimize it.

$$\begin{aligned}
\mathbb{E}[L_{\text{masked}}(w)] &= \sum_{n=1}^{N} \mathbb{E}\left[(y_n - (w^T (x_n \odot m_n))^2\right] \\
&= \sum_{n=1}^{N} \mathbb{E}\left[y_n^2 - 2y_n w^T (x_n \odot m_n) + (w^T (x_n \odot m_n))^2\right] \\
&= \sum_{n=1}^{N} \left(\mathbb{E}[y_n^2] - 2\mathbb{E}[y_n w^T (x_n \odot m_n)] + \mathbb{E}[(w^T (x_n \odot m_n))^2]\right)
\end{aligned}$$

- Since $\mathbb{E}[y_n^2]$ is independent of w, $\mathbb{E}[y_n^2]$ is considered constant. Our loss function becomes,

$$\begin{aligned}
\mathbb{E}[L_{\text{masked}}(w)] &= \sum_{n=1}^{N} \left(-2\mathbb{E}[y_n w^T (x_n \odot m_n)] + \mathbb{E}[(w^T (x_n \odot m_n))^2]\right) \\
&= \sum_{n=1}^{N} \left(-2\mathbb{E}[y_n w^T (x_n \odot m_n)]\right) + \sum_{n=1}^{N} \left(\mathbb{E}[(w^T (x_n \odot m_n))^2]\right)
\end{aligned}$$

$$\boxed{\mathbb{E}[L_{\text{masked}}(w)] = \sum_{n=1}^{N}\sum_{d=1}^{D} (-2\mathbb{E}[y_n w_n (x_{nd} m_{nd})]) + \sum_{n=1}^{N}\sum_{d=1}^{D} \left(\mathbb{E}[(w_n (x_{nd} m_{nd}))^2]\right)}$$

In the above equation, the regularization term is the expected value of the square of the masked linear prediction. As we observe from the above equation, there are two terms, one is the loss function and other is the regularization term. The regularization term signifies the importance given to a particular feature. As some of the features get masked out, these features are given less importance as they are not frequently considered for prediction. On the other hand, the features which aren't frequently masked off are given more importance.

w_{nd} corresponds to d^{th} feature that signifies the weight assigned to d^{th} feature of the input. If the feature gets frequently masked off, w_{nd} is kept low. If the feature is frequently retained, w_{nd} get a larger value. In this way, regularization is done so that the values of the weight vector are kept optimal in order to avoid overfitting on training data.

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

Student Name: Pranjal Maroti Nandeshwar
Roll Number: 231110035
Date: September 15, 2023

QUESTION
6

1. Accuracy obtained by method 1 is,

$$\boxed{\textit{Accuracy} : 46.893203883495 15\%}$$

2. Accuracies obtained by method 2 are,

λ	Accuracy
0.01	58.090614886731395 %
0.1	59.54692556634305 %
1.0	67.39482200647248 %
10	73.28478964401295 %
20	71.68284789644012 %
50	65.08090614886731 %
100	56.47249190938511 %

Table 1: Accuracies obtained by method 2

From the table, we can observe that we get the maximum accuracy for $\lambda = 10$.