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

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

QUESTION 1

To find the optimal values of w_c and M_c for the given objective function, we will break down the optimization problem into two parts: optimizing w_c and optimizing M_c.

1. Optimizing with respect to w_c:

The objective function is:

$$L(w_c, M_c) = \frac{1}{N_c} \sum_{n:y_n=c} \left((x_n - w_c)^T M_c (x_n - w_c)\right) - \log|M_c|$$

To find the optimal w_c, we take the derivative of this objective function with respect to w_c and set it to zero:

$$\frac{\partial L}{\partial w_c} = \frac{2}{N_c} \sum_{n:y_n=c} M_c(x_n - w_c) = 0$$

Solving for w_c:

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

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

So, w_c is the mean of all x_n that are labeled as class c.

2. Optimizing with respect to M_c:

To find the optimal M_c, we take the derivative of the objective function with respect to M_c and set it to zero:

$$\frac{\partial L}{\partial M_c} = \frac{1}{N_c} \sum_{n:y_n=c} \left((x_n - w_c)(x_n - w_c)^T\right) - (M_c^{-1})^T = 0$$

Solving for M_c:

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

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

This implies that M_c has to be invertible, which is also justified because $\log|M_c|$ is defined only when $|M_c| > 0$ which means that only when M_c is invertible.

3. Special case when M_c is an identity matrix (I):

If $M_c = I$, then the term $|M_c|$ simplifies to $|I| = 1$, and the objective function becomes:

$$L(w_c, I) = \frac{1}{N_c} \sum_{n:y_n=c} \left((x_n - w_c)^T I (x_n - w_c)\right) - \log|I|$$

$$L(w_c, I) = \frac{1}{N_c} \sum_{n:y_n=c} \|x_n - w_c\|^2 - \log 1 = \frac{1}{N_c} \sum_{n:y_n=c} \|x_n - w_c\|^2$$

$$L(w_c, I) = \frac{1}{N_c} \sum_{n:y_n=c} \|x_n - w_c\|^2$$

This is equivalent to the **mean squared error (MSE) loss**. So, in this special case, the model reduces to a classification model with the MSE loss function.

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

QUESTION 2

Yes, the one-nearest-neighbor (1-NN) algorithm is consistent in the noise-free setting where every training input is labeled correctly. In this scenario, 1-NN always selects the nearest neighbor from the training data, which is guaranteed to be the correct label since every input is labeled correctly. As the amount of training data approaches infinity, 1-NN will consistently choose the nearest neighbor with zero error, leading to an error rate that approaches the optimal error rate of zero. Therefore, in the noise-free setting, the 1-NN algorithm is consistent.

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

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

QUESTION 3

Variance

A suitable criterion to choose a feature to split on while constructing Decision Trees can be the variance. When constructing a regression tree, the objective is to partition the data in a way that the labels within each child node are as similar as possible. Minimizing the variance of labels within a node means that the labels are tightly clustered around the node's mean. Thus it will be an optimal criteria.

We will use the following steps to find the required split point:

1. Calculate the variance of the labels in the current node $\text{Var}_{\text{total}}$:

$$\text{Var}_{\text{total}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2$$

where:
- n is the number of data points in the current node.
- y_i represents the label of the ith data point.
- μ is the mean of the labels in the current node.

2. For each potential feature and split point: a. Split the data into two subsets based on the feature and split point. b. Calculate the variance of the labels in each of the resulting child nodes (Var_{left} *and* $\text{Var}_{\text{right}}$) :

$$\text{Var}_{\text{left}} = \frac{1}{n_{\text{left}}}\sum_{i=1}^{n_{\text{left}}}(y_i - \mu_{\text{left}})^2$$

$$\text{Var}_{\text{right}} = \frac{1}{n_{\text{right}}}\sum_{i=1}^{n_{\text{right}}}(y_i - \mu_{\text{right}})^2$$

where:
- n_{left} and n_{right} are the number of data points in the left and right child nodes, respectively.
- y_i represents the label of the ith data point.
- μ_{left} and μ_{right} are the means of the labels in the left and right child nodes, respectively.

3. Calculate the reduction in variance:

$$\text{Reduction in Variance} = \text{Var}_{\text{total}} - \left(\frac{n_{\text{left}}}{n}\cdot\text{Var}_{\text{left}} + \frac{n_{\text{right}}}{n}\cdot\text{Var}_{\text{right}}\right)$$

We will choose the feature and split point that maximize this reduction in variance, as it quantifies the improvement in node homogeneity after the split.

.

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

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

QUESTION 4

For the unregularized linear regression model, where the solution is $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, the prediction at a test input $\mathbf{x}^*$ can be expressed as:

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

Therefore,

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

This equation can be rewritten as:

$$y^* = \mathbf{W}\mathbf{y}$$

Where $\mathbf{W}$ is defined as:

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

Consequently, $\mathbf{W}$ is a 1xN matrix, and $\mathbf{y}$ can be represented as a column vector:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Hence, the prediction y^* can be expressed as:

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

Here, w_n corresponds to the n-th element of the 1xN matrix $\mathbf{W}$.
$\mathbf{X}$ is the matrix whose rows are the N training vectors $\mathbf{x}_n$, therefore w_n can be expressed as:

$$w_n = \mathbf{x}^{*T}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_n$$

Therefore, w_n depends on x^* and all the training data from x_1 to x_N

In k-Nearest Neighbors, w_n depends on the inverse distance between x^* and x_n as:

$$w_n = \frac{1}{\|x^* - x_n\|}$$

The differences between linear regression and KNN are:
• In linear regression, the weights depend on the inner product of x^* and x_n weighted by the matrix X^TX, while in KNN, the weights depend on the inverse Euclidean distance between x^* and x_n.
• In the linear regression case, $\mathbf{x}^*$ is in the numerator of the weight expression, while in KNN, it appears in the denominator.

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

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

QUESTION 5

The new loss function is defined as:

$$\mathbf{L}(w) = \sum_i \left(y_i - \mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2$$

Where: - i is the index of the data points - y_i is the target for data point i - $\mathbf{w}$ is the weight vector - $\mathbf{x}_{\tilde{i}}$ is the masked input for data point i

The expectation over the masked inputs is given by:

$$\mathbf{E}[\mathbf{L}(\mathbf{w})] = \mathbf{E}\left[\sum_i \left(y_i - \mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2\right]$$

Since the mask vectors $\mathbf{m}_i$ are random and follow a Bernoulli distribution with parameter p, we can consider the expectation over these masks:

$$\mathbf{E}[\mathbf{L}(\mathbf{w})] = \sum_i \mathbf{E}\left[\left(y_i - \mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2\right]$$

Expanding the squared term inside the expectation:

$$\mathbf{E}\left[\left(y_i - \mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2\right] = \mathbf{E}\left[y_i^2 - 2y_i \mathbf{w}^T \mathbf{x}_{\tilde{i}} + \left(\mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2\right]$$

Now, we can compute the expectation term by term:

$$\mathbf{E}[\mathbf{y_i^2}]$$

is just a constant with respect to $\mathbf{w}$, so it doesn't affect the minimization.

$$\mathbf{E}[\mathbf{-2y_i w^T x_{\tilde{i}}}] = -2\mathbf{w}^T \mathbf{E}[\mathbf{y_i x_{\tilde{i}}}]$$

$$\mathbf{E}\left[\left(\mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)^2\right] = \mathbf{E}\left[\left(\mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)\left(\mathbf{w}^T \mathbf{x}_{\tilde{i}}\right)\right] = \mathbf{E}[\mathbf{w}^T \mathbf{x}_{\tilde{i}} \mathbf{x}_{\tilde{i}}^T \mathbf{w}]$$

Now, let's compute the expectation of $\mathbf{x}_{\tilde{i}} \mathbf{x}_{\tilde{i}}^T$, where $\mathbf{x}_{\tilde{i}}$ is the elementwise product of $\mathbf{x}_i$ and the mask $\mathbf{m}_i$:

$$\begin{aligned}
\mathbf{E}[\mathbf{x}_{\tilde{i}} \mathbf{x}_{\tilde{i}}^T] &= \mathbf{E}\left[\left(\mathbf{x}_i \odot \mathbf{m}_i\right)\left(\mathbf{x}_i \odot \mathbf{m}_i\right)^T\right] \\
&= \mathbf{E}[\mathbf{x}_i \mathbf{x}_i^T \odot (\mathbf{m}_i \mathbf{m}_i^T)] \\
&= \mathbf{E}[\mathbf{x}_i \mathbf{x}_i^T] \odot (p\mathbf{I})
\end{aligned}$$

Where $\mathbf{I}$ is the identity matrix and $\odot$ denotes the elementwise product.

Now, we can rewrite the expected loss as:

$$\mathbf{E}[\mathbf{L}(\mathbf{w})] = \sum_i \left(\mathbf{E}[\mathbf{y_i x_{\tilde{i}}}]\right) - 2\mathbf{w}^T \left(\mathbf{E}[\mathbf{x_i x_{\tilde{i}}}]\right) \odot (p\mathbf{I})\mathbf{w}$$

Now, let's define a regularized loss function by adding a regularization term:

$$\mathbf{L}_{\text{reg}}(\mathbf{w}) = \sum_i \left(\mathbf{E}[\mathbf{y_i x_{\tilde{i}}}]\right) - 2\mathbf{w}^T \left(\mathbf{E}[\mathbf{x_i x_{\tilde{i}}}]\right) \odot (p\mathbf{I})\mathbf{w} + \lambda\|\mathbf{w}\|^2$$

Where λ is the regularization parameter, and $\|\mathbf{w}\|^2$ represents the L2 norm of the weight vector $\mathbf{w}$.

Comparing the two expressions, we can see that minimizing the expected value of the new loss function is equivalent to minimizing the regularized loss function with the regularization term:

$$\lambda\|\mathbf{w}\|^2 = 2\mathbf{w}^T \left(\mathbf{E}[\mathbf{x_i x_{\tilde{i}}}]\right) \odot (p\mathbf{I})\mathbf{w}$$

Therefore, minimizing the expected value of the new loss function is equivalent to minimizing a regularized loss function with an L2 regularization term, where the regularization strength is determined by λ, and the regularization operates on the weights $\mathbf{w}$.

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

QUESTION 6

Student Name: Sharvani Jadhav
Roll Number: 210960
Date: December 1, 2024

Method 1: Using the Euclidean distance, the initial test accuracy is 46.89
The test accuracy after optimizing the model for various iterations is as follows:
Iteration 5: 47.18
Iteration 10: 48.24
Iteration 15: 48.90
Iteration 20: 49.72
Iteration 25: 50.13
Iteration 30: 50.21

Method 2: For different values of the regularization parameter (λ):
$\lambda = 0.01$: Test accuracy is 58.09
$\lambda = 0.1$: Test accuracy is 59.55
$\lambda = 1$: Test accuracy is 67.39
$\lambda = 10$: Test accuracy is 73.28
$\lambda = 20$: Test accuracy is 71.68
$\lambda = 50$: Test accuracy is 65.08
$\lambda = 100$: Test accuracy is 56.47
In conclusion, $\lambda = 10$ produced the best test accuracy of 73.28