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

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

QUESTION 1

$$(\hat{\mathbf{w}}_c, \hat{M_c}) = \arg\min_{\mathbf{w}_c, M_c} \frac{1}{N_c} \sum_{n:y_n=c} \left((\mathbf{x}_n - \mathbf{w}_c)^T M_c (\mathbf{x}_n - \mathbf{w}_c) - \log |M_c| \right).$$

Let's simplify the gradient equation for $\hat{\mathbf{w}}_c$ by considering that M_c is PD:

$$\frac{1}{N_c} \sum_{n:y_n=c} M_c(\hat{\mathbf{w}}_c - \mathbf{x}_n) = 0.$$

Since M_c is PD, we can use the fact that M_c is symmetric and its inverse M_c^{-1} is also symmetric. Multiplying both sides by M_c^{-1} gives:

$$\frac{1}{N_c} \sum_{n:y_n=c} M_c^{-1} M_c(\hat{\mathbf{w}}_c - \mathbf{x}_n) = 0.$$

As $M_c^{-1} M_c$ results in the identity matrix I, we have:

$$\frac{1}{N_c} \sum_{n:y_n=c} I(\hat{\mathbf{w}}_c - \mathbf{x}_n) = 0.$$

Simplifying further:

$$\frac{1}{N_c} \sum_{n:y_n=c} (\hat{\mathbf{w}}_c - \mathbf{x}_n) = 0.$$

Solving for $\hat{\mathbf{w}}_c$:

$$\hat{\mathbf{w}}_c = \frac{1}{N_c} \sum_{n:y_n=c} \mathbf{x}_n.$$

Now, for $\hat{M_c}$, the equation is:

$$\frac{1}{N_c} \sum_{n:y_n=c} \left((\mathbf{x}_n - \mathbf{w}_c)(\mathbf{x}_n - \mathbf{w}_c)^T - \hat{M_c}^{-1} \right) = 0.$$

Solving for $\hat{M_c}^{-1}$:

$$\hat{M_c}^{-1} = \frac{1}{N_c} \sum_{n:y_n=c} (\mathbf{x}_n - \mathbf{w}_c)(\mathbf{x}_n - \mathbf{w}_c)^T.$$

To find $\hat{M_c}$, we take the inverse of $\hat{M_c}^{-1}$, which gives:

$$\hat{M_c} = \left(\frac{1}{N_c} \sum_{n:y_n=c} (\mathbf{x}_n - \mathbf{w}_c)(\mathbf{x}_n - \mathbf{w}_c)^T \right)^{-1}.$$

1. In the special case when M_c is the identity matrix, the optimal values are $\hat{\mathbf{w}}_c = \frac{1}{N_c} \sum_{n:y_n=c} \mathbf{x}_n$ for $\mathbf{w}_c$ and $\hat{M_c} = \left(\frac{1}{N_c} \sum_{n:y_n=c} (\mathbf{x}_n - \hat{\mathbf{w}}_c)(\mathbf{x}_n - \hat{\mathbf{w}}_c)^T \right)^{-1}$ for M_c.

2. These solutions simplify due to the identity matrix, making $\hat{\mathbf{w}}_c$ the average of class c data points and $\hat{M_c}$ the inverse of the sample covariance matrix of class c examples.

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

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

QUESTION 2

When we have a lot of training data that's perfectly labeled, and there are no mistakes in these labels, the one-nearest-neighbor (1NN) algorithm works really well. Imagine you have a huge collection of training examples, and they cover all possible cases. So, when you get a new test example, you can always find a super similar example in your training set. As you collect more and more training examples, you're almost guaranteed to find a very similar one for any test case. This means 1NN will make almost no mistakes in classifying test examples, making it consistent in this perfect scenario. so 1NN will be consistent in this case.

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

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

QUESTION 3

When constructing Decision Trees for regression, it is essential to select features that enable node splits leading to subsets of target values with minimal variability. In other words, the goal is to reduce the diversity of the real-valued labels within each child node. To achieve this, we can employ a criterion known as *Variance Reduction*, which quantifies the consistency of real-valued labels in the regression scenario.

Here are the steps for determining the optimal feature to split on using the Variance Reduction criterion:

1. For each feature, compute the variance of the target variable across all potential splits. 2. Identify the feature and split point that result in the most significant reduction in variance. 3. Continue with steps 1 and 2 iteratively until either completely homogeneous nodes are achieved or another predefined stopping condition is met.

The variance is calculated using this formula:

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

Where: - y_i represents the target value for the i-th example. - $\bar{y}$ is the mean target value. - n is the number of examples.

The formula for computing the reduction in variance due to a split is expressed as:

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

Where: - $\text{Variance}_{\text{parent}}$ is the variance of the target variable in the parent node. - $\text{Variance}_{\text{left}}$ and $\text{Variance}_{\text{right}}$ are the variances of the target variable in the left and right child nodes, respectively. - n_{left} and n_{right} denote the number of examples in the left and right child nodes, respectively. - n represents the total number of examples in the parent node.

This approach proves effective for regression tasks as it favors splits that lead to child nodes with more consistent sets of target values, thereby resulting in lower variances. This, in turn, ensures more precise and less erratic predictions within the context of regression analysis

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

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

QUESTION 4

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

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

Hence, $y^* = \mathbf{x}^{*T}\mathbf{W}$, where $\mathbf{W} = \mathbf{x}^{*T}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. Thus, $\mathbf{W}$ emerges as a 1xN matrix. We can also represent $\mathbf{y}$ as $\mathbf{y} = (y_1, y_2, \dots, y_N)^T$.

As a result, $y^* = \mathbf{W} \cdot \mathbf{y} = \sum_{n=1}^{N} w_n y_n$.

Now, the components w_n correspond to the n-th index of the 1xN matrix $\mathbf{W}$. It's worth noting that $\mathbf{W}$ depends on the input $\mathbf{x}^*$ as well as all the training data from $\mathbf{x}_1$ to $\mathbf{x}_N$. This is because the term $\mathbf{X}^T\mathbf{X}$ appears in the expression for w_n.

This stands in contrast to weighted k-NN, where individual weights depend only on $\mathbf{x}^*$ and $\mathbf{x}_n$. Additionally, in this case, $\mathbf{x}^*$ appears in the numerator, whereas in k-NN, it's in the denominator. Another distinction is that w_n is expressed as a product of $\mathbf{x}^*$, while in k-NN, they are expressed as a sum in the denominator.

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

QUESTION 5

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

Let the given Loss function using masked input be:

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

Therefore:

$$L(M) = \|\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}\|^2$$

$$L(M) = \|\mathbf{y} - (\mathbf{R} * \mathbf{X})\mathbf{w}\|^2$$

When the input $\mathbf{X}$ is dropped out such that any input dimension is retained with probability p, then the expected value of $L(M)$ is be:

$$\mathbb{E}_{R \sim \text{Bernoulli}(n,p)}[L(M)]$$

This needs to be minimized with respect to $\mathbf{w}$, so let the new objective function be:

$$L(\mathbf{w}) = \arg\min_{\mathbf{w}} \mathbb{E}_{R \sim \text{Bernoulli}(n,p)} \left[\|\mathbf{y} - (\mathbf{R} * \mathbf{X})\mathbf{w}\|_2^2 \right]$$

$$L(w) = \arg\min_{w} \mathbb{E}_{R \sim \text{Bernoulli}(n,p)} \left[(\mathbf{y} - (\mathbf{R} * \mathbf{X})\mathbf{w})^T \left(\mathbf{y} - (\mathbf{R} * \mathbf{X})\mathbf{w} \right) \right]$$

Let $k = \mathbf{y} - (\mathbf{R} * \mathbf{X})\mathbf{w}$ and $\mu = \mathbb{E}_{R \sim \text{Bernoulli}(n,p)}[k] = \mathbf{y} - p\mathbf{X}\mathbf{w}$ then -

$$L(w) = \arg\min_{w} \mathbb{E}_{R \sim \text{Bernoulli}(n,p)} \left[\mathbf{k}^T \mathbf{k} \right]$$

This leads to:

$$L(w) = \arg\min_{w} \left(\mu^T \mu + \text{TRACE}\left((\mathbf{k} - \mu)(\mathbf{k} - \mu)^T \right) \right)$$

Further simplifying:

$$L(w) = \arg\min_{w} \left((\mathbf{y} - p\mathbf{X}\mathbf{w})^T (\mathbf{y} - p\mathbf{X}\mathbf{w}) + p(1-p)\text{TRACE}\left((\mathbf{X}\mathbf{w})(\mathbf{X}\mathbf{w})^T \right) \right)$$

Which can be written as:

$$L(w) = \arg\min_{w} \left(\|y - pXw\|^2 + p(1-p)\text{TRACE}\left(Xww^T X^T \right) \right)$$

$$L(w) = \arg\min_{w} \left(\|y - pXw\|^2 + p(1-p) \left\| \sqrt{\text{diag}(\mathbf{X}^T\mathbf{X})}\mathbf{w} \right\|^2 \right)$$

Comparing $L(w)$ with the ridge regression objective function $L_{\text{ridge}}(w)$:

$$L_{\text{ridge}}(w) = \arg\min_{w} \left[(\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w} \right]$$

”Clearly, the goal $L(w)$ closely resembles the objective function seen in ridge regression. It can be equated to minimizing a loss function with regularization, where the component $p(1-p)\|\sqrt{\text{diag}(\mathbf{X}^T\mathbf{X})}\mathbf{w}\|^2$ acts as a regularization term, while $\|y - p\mathbf{X}\mathbf{w}\|^2$ simulates a squared loss.”

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

Student Name: Digambar singh
Roll Number: 200337
Date: September 15, 2023

QUESTION 6

Method 1:
Accuracy for test input: 46.9
Method 2:
accuracy for lambada (0.01) is: 58.1
accuracy for lambada (0.1) is: 59.5
accuracy for lambada (1) is: 67.4
accuracy for lambada (10) is: 73.3
accuracy for lambada (20) is: 71.7
accuracy for lambada (50) is: 65.1
accuracy for lambada (100) is: 56.5

lambda =10 gives the best accuracy if we increase lambda almost 11 accuracy increases after that accuracy decreases.