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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 1

We begin with the optimization problem:

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

To minimize the optimization problem in two variables, we would be taking derivatives with respect to both the variables and equate to zero. Solving them simultaneously would give us the required solution.

Taking the partial derivative of the above objective with respect to $\mathbf{w}_c$, we obtain:

$$(\mathbf{M}_c + \mathbf{M}_c^T) \sum_{\mathbf{x}_n : y_n = c} (\mathbf{x}_n - \mathbf{w}_c) = 0 \qquad \text{((from matrixcookbook) (1))}$$

Since $\mathbf{M}_c$ is a positive definite matrix, it cannot be antisymmetric for all $\mathbf{x}$, i.e.,

$$\forall \mathbf{x}, (\mathbf{x}^T \mathbf{M}_c \mathbf{x} \geq 0 \implies \mathbf{x}^T \mathbf{M}_c^T \mathbf{x} \geq 0)$$

. This implies that we have:

$$\sum_{\mathbf{x}_n : y_n = c} (\mathbf{x}_n - \mathbf{w}_c) = 0 \tag{2}$$

Which further leads to:

$$\mathbf{w}_c = \frac{1}{N_c} \sum_{\mathbf{x}_n : y_n = c} \mathbf{x}_n \tag{3}$$

Next, taking the partial derivative of the objective with respect to $\mathbf{M}_c$, we get:

$$\sum_{\mathbf{x}_n : y_n = c} \frac{1}{N_c} (\mathbf{x}_n - \mathbf{w}_c)(\mathbf{x}_n - \mathbf{w}_c)^T - (\mathbf{M}_c^T)^{-1} = 0 \qquad \text{((from matrixcookbook) (4))}$$

Hence, from equation (5), we find where $\mathbf{w}_c$ can be substituted from above:

$$\mathbf{M}_c = \left(\left(\sum_{\mathbf{x}_n : y_n = c} \frac{1}{N_c} (\mathbf{x}_n - \mathbf{w}_c)(\mathbf{x}_n - \mathbf{w}_c)^T \right)^T \right)^{-1} \tag{4}$$

When $\mathbf{M}_c$ is replaced by $\mathbf{I}$, the objective function transforms as follows:

$$\sum_{\mathbf{x}_n : y_n = c} \frac{1}{N_c} (\mathbf{x}_n - \mathbf{w}_c)^T \mathbf{I} (\mathbf{x}_n - \mathbf{w}_c) - \log |\mathbf{I}| = \sum_{\mathbf{x}_n : y_n = c} \frac{1}{N_c} ||\mathbf{x}_n - \mathbf{w}_c||^2 - \log 1 = \sum_{\mathbf{x}_n : y_n = c} \frac{1}{N_c} ||\mathbf{x}_n - \mathbf{w}_c||^2 \tag{5}$$

Finally, it's worth noting that when the positive definite matrix $\mathbf{M}_c$ simplifies to an identity matrix, denoted as $\mathbf{I}$, the objective function mentioned earlier takes on a special significance and can be referred to as the Learn With Prototypes (LwP) model with an L_2-norm distance metric.

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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 2

The 1-Nearest Neighbor (1NN) classification algorithm demonstrates **consistency**. This consistency arises from the presence of an infinite number of precisely labeled training data points, all free from any noise or errors.

In essence, this abundance of correctly labeled training data ensures that when presented with a test data point, there will always be a training data point in close proximity to it. As the number of training data points approaches infinity, the probability of finding such a nearby point tends to 1. Therefore, the classification performed by 1NN in this scenario is error-free, as the test data point essentially already exists within the training set, ensuring accurate classification.

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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 3

In the context of using decision trees for regression analysis, we employ the Mean Squared Error (MSE) as a criterion for selecting the optimal feature to split on, in contrast to using Entropy or Information Gain, which are typically employed in classification tasks. The goal of this approach is to quantify the homogeneity or diversity of data within each node of the tree by measuring the variance of the labels.

Consider a feature denoted as F and a potential split point s along this feature. We divide the input data into two subsets:

- The set of data points with values of feature F greater than s, denoted as set $> s$.
- The set of data points with values of feature F less than s, denoted as set $< s$.

Let a be the number of elements in the $> s$ set, and b be the number of elements in the $< s$ set. Therefore, the total number of data points before the split is $a + b$. We calculate the variance of the labels in these two subsets. The variance of the labels in the $> s$ subset is denoted as V_1. The variance of the labels in the $< s$ subset is denoted as V_2.

Additionally, we calculate the variance of the original dataset (before the split) and denote it as V.

To determine the optimal feature F and split point s, we evaluate the following objective function:

$$f(F, s) = \arg\max_{F,s} \left(V - \frac{a}{a+b} V_1 - \frac{b}{a+b} V_2 \right) \tag{1}$$

We want to maximize the reduction in the overall variance V by considering the variances V_1 and V_2 weighted by the proportion of data points in each subset (a and b).

At each stage of constructing the decision tree, we choose the feature F and split point s that maximize the reduction in variance as defined by the $f(F, s)$ equation. This ensures that the tree branches in a way that minimizes the variance of the labels within the resulting nodes, ultimately leading to more homogeneous subsets and better regression performance.

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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 4

Given the linear regression coefficient vector as $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, we can express the prediction at a test input x^* as follows:

$$f(x^*) = \hat{\mathbf{w}}^T\mathbf{x}^* = \mathbf{x}^{*T}\hat{\mathbf{w}} \tag{1}$$

Therefore, we have:

$$f(x^*) = x^{*T}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

Hence, we can represent $f(x^*)$ in terms of a matrix multiplication:

$$f(x^*) = \mathbf{W}\mathbf{y} \tag{3}$$

Here, $\mathbf{W} = x^{*T}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, and it results in a $1 \times N$ matrix. Additionally, we can represent y as a column vector of order $N \times 1$ with its entries being from $y_1, y_2, y_3....y_n$

This allows us to express $f(x^*)$ as a specific product involving the training data and x^*:

$$f(x^*) = \sum_{n=1}^{n} x^{*T}(\mathbf{X}^T\mathbf{X})^{-1}x_n y_n \tag{4}$$

$$f(x^*) = \sum_{n=1}^{n} w_n y_n \tag{5}$$

where $n = 1$, and w_n corresponds to the n-th index of the $1 \times N$ matrix W.

Furthermore, we can express w_n in terms of x^* and all the training data x_n as follows:

$$w_n = x^{*T}(\mathbf{X}^T\mathbf{X})^{-1}x_n \tag{6}$$

This expression for w_n highlights that it depends on the test input x^* and all the training data from x_1 to x_n. This is different from weighted kNN, where individual weights depend solely on x^* and x_n (**inverse Euclidean distance**). The form of w_n is a general form of an **inner product similarity (similar to Mahalanobis distance)** between the test input x^* and training input x_n, modulated by matrix $(\mathbf{X}^T\mathbf{X})^{-1}$, which depends on the entire training data.

In addition, it's important to note that in this formulation, x^* appears in the numerator, whereas in kNN, it typically appears in the denominator of the weight calculation. Another distinction is that w_n is expressed as a product involving x^*, while in kNN, the weights are expressed as sums in the denominator and the similarities only depend on the test input and the nearest neighbour.

These differences illustrate how the linear regression approach and kNN differ in terms of weight computation and the role of the test input x^* in the weight calculation.

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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 5

Let the given Loss function using masked input using MSE (mean squared error) be:
$L(M) = \|\mathbf{y} - \mathbf{X}\tilde{w}\|^2$ which can be even expressed as:

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

where $\mathbf{R}$ is the matrix of random-variables that masks the input $\mathbf{X}$ with the bernoulli distribution.

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

$$E_{R\sim Bernoulli(n,p)}\left[L(M)\right]$$

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

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

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

Let $\mathbf{k} = \mathbf{y} - (\mathbf{R}^*\mathbf{X})\mathbf{w}$ and $\mu = E_{R\sim Bernoulli(n,p)}[\mathbf{k}] = \mathbf{y} - p\mathbf{X}\mathbf{w}$. Then, we get:

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

Now, writing $\mathbf{k}^T\mathbf{k} = (\mathbf{k}-\mu)^T(\mathbf{k}-\mu) + 2\mu^T\mathbf{k} - \mu^T\mu$, we get:

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

where $\text{TRACE}(AA^T) = \sum_{i=1}^{n} A_i^2$.

Now, replacing $\mathbf{k} - \mu = (R - pI)\mathbf{X}\mathbf{w}$ and evaluating the expectation, we get:

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

Therefore, we have after clubbing and opening both the terms

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

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

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

Clearly, the objective L($\mathbf{w}$) resembles the objective function of ridge regression, hence is equivalent to minimizing a regularized loss function where the term

$$p(1-p)\text{diag}(\mathbf{X}^T\mathbf{X})$$

acts as a regularizer and $\|\mathbf{y} - p\mathbf{X}\mathbf{w}\|^2$ is like squared loss.

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

Student Name: Shwetank Anand
Roll Number: 211025
Date: September 15, 2023

QUESTION 6

Using **L2 norm** for prototype based classifier,
For **Method 1**, the accuracy obtained is 46.89320388349515 %.
For **Method 2**,
Accuracy during test time for lambda= 0.01 is: 58.090614886731395 %.
Accuracy during test time for lambda= 0.1 is: 59.54692556634305 %.
Accuracy during test time for lambda= 1 is: 67.39482200647248 %.
Accuracy during test time for lambda= 10 is: 73.28478964401295 %.
Accuracy during test time for lambda= 20 is: 71.68284789644012 %.
Accuracy during test time for lambda= 50 is: 65.08090614886731 %.
Accuracy during test time for lambda= 100 is: 56.47249190938511 %.
The maximum accuracy is obtained for lambda= 10 using L2 norm.