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

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION 1

For this question, $x_n \in \mathbb{R}^D$, $w_c \in \mathbb{R}^D$, $\mathbf{M}_c$ is a $D \times D$ positive semi-definite (PSD) matrix and N_c is a scalar that denotes the number of training examples for class c.

We have to estimate the w_c and M_c by solving the below optimization problem

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

$\hat{w}_c, \hat{M}_c$ are estimate of w_c, M_c

Let

$$L(w_c, \mathbf{M}_c) = \sum_{x_n : y_n = c} \frac{1}{N_c}(x_n - w_c)^T \mathbf{M}_c (x_n - \mu_c) - \log|\mathbf{M}_c| \tag{2}$$

for optimal w_c, $\frac{\partial L}{\partial w_c}$ should be 0

$$\frac{\partial L}{\partial w_c} = \frac{1}{N_c} \sum_{x_n : y_n = c} \frac{\partial (x_n - w_c)^\top \mathbf{M}_c (x_n - w_c)}{\partial (x_n - w_c)} \cdot \frac{\partial (x_n - w_c)}{\partial w_c} = \mathbf{0} \tag{3}$$

$$\sum_{x_n : y_n = c} (\mathbf{M}_c + \mathbf{M}_c^\top)(w_c - x_n) = \mathbf{0} \tag{4}$$

M_c is PSD and Symmetric, so,

$$\sum_{x_n : y_n = c} 2\mathbf{M}_c (w_c - x_n) = 0 \tag{5}$$

As $\mathbf{M}_c$ is PSD, and non-singular,$\mathbf{M}_c \mathbf{x} = \mathbf{0}$ then $\mathbf{x} = \mathbf{0}$. So,

$$\sum_{x_n : y_n = c} (w_c - x_n) = \mathbf{0} \tag{6}$$

$$w_c = \frac{1}{N_c} \sum_{x_n : y_n = c} x_n = \mu_c \tag{7}$$

Now, for the optimal M_c, $\frac{\partial L}{\partial M_c}$ should be 0

$$\frac{\partial L}{\partial \mathbf{M}_c} = \frac{1}{N_c} \sum_{x_n : y_n = c} (x_n - w_c)(x_n - w_c)^\top - \frac{1}{|\mathbf{M}_c|} \cdot \mathrm{Adj}(\mathbf{M}_c)^\top = \mathbf{0} \tag{8}$$

And,

$$\frac{\mathrm{Adj}(\mathbf{M}_c)^\top}{|\mathbf{M}_c|} = \frac{1}{N_c} \sum_{x_n : y_n = c} (x_n - \mu_c)(x_n - \mu_c)^\top = \frac{1}{N_c} \sum_{x_n : y_n = c} (x_n x_n^\top + \mu_c \mu_c^\top - x_n \mu_c^\top - \mu_c x_n^\top) \tag{9}$$

After simplifying:

$$(\mathbf{M}_c^{-1})^\top = \left(\frac{1}{N_c} \sum_{x_n : y_n = c} x_n x_n^\top \right) - \mu_c \mu_c^\top \tag{10}$$

$$\mathbf{M}_c^{-1} = \left(\left(\frac{1}{N_c} \sum_{x_n : y_n = c} x_n x_n^\top \right) - \mu_c \mu_c^\top \right)^\top = \left(\frac{1}{N_c} \sum_{x_n : y_n = c} x_n x_n^\top \right) - \mu_c \mu_c^\top \tag{11}$$

So, the optimal values of w_c and $\mathbf{M}_c$ are:

$$\hat{w}_c = \mu_c = \frac{1}{N_c} \sum_{x_n : y_n = c} x_n \tag{12}$$

$$\hat{\mathbf{M}}_c = \left(\frac{1}{N_c} \sum_{x_n : y_n = c} x_n x_n^\top \right) - \mu_c \mu_c^\top \tag{13}$$

When M_c is an identity matrix, then the objective function will be

$$L(w_c, I) = \sum_{x_n : y_n = c} \frac{1}{N_c} (x_n - w_c)^T (x_n - \mu_c) - \log |I| \tag{14}$$

$\log |I| = 0$ so the final objective function in the case of M_c is an identity matrix will be

$$L(w_c) = \sum_{x_n : y_n = c} \frac{1}{N_c} (x_n - w_c)^T (x_n - \mu_c) \tag{15}$$

As $w_c = \mu_c$ so

$$L(w_c) = \sum_{x_n : y_n = c} \frac{1}{N_c} ||x_n - \mu_c||^2 \tag{16}$$

By looking at the above function, it looks like we are trying to find the optimal w_c that minimizes the squared distance between data points and their corresponding class centers, weighted by the class prior probabilities, i.e., $\frac{1}{N_c}$. So the problem now becomes K-means clustering

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION 2

In the noise-free setting, where every training input is labeled correctly, the Bayes optimal error rate is zero because there are no errors in the training data. However, the one-nearest-neighbor might not be consistent. Because in one-nearest-neighbor, any output relies on the nearest neighbor. In the case of finite data, this might be the case when the nearest neighbor may not be the correct distribution. When data becomes infinite, it's not guaranteed that one-nearest-neighbor will be consistent as it is more on the closest-neighbor distribution than the whole data distribution here. So, one-nearest-neighbor is not always consistent, even in a noise-free setting.

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

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION 3

In the case of the regression problem for a similar input type, the output values should also be similar but depend on the algorithm. If the values are related, the variance is less, and if the values are far apart, the gap between the values is more, which means the variance is high. We will use the concept of variance at each node for the decision tree.

In the decision tree, at each node within the data set, if variance is low, then that data set will be homogeneous and similar types of labels. So, we will divide the data at each node so that the variance of labels in the child node should be minimal.

For any node P, the variance can be calculated as:

$$\text{Variance}(P) = \frac{1}{|N(P)|} \sum_{y_i \in N(P)} (y_i - \mu(P))^2 \tag{17}$$

here, $N(P)$ is the number of data points at node P
y_i is the label corresponding to data at node P
$\mu(P)$ is the mean of labels at node P

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION 4

Let predict for x_* as y_*

$$f(x_*) = y_* = x_*\hat{w} \tag{18}$$

$$\hat{w} = (\mathbf{X^T X})^{\mathbf{-1}}\mathbf{X^T}y \tag{19}$$

$$f(x_*) = x_*(\mathbf{X^T X})^{\mathbf{-1}}\mathbf{X^T}y = \sum_{n=1}^{N} w_n y_n \tag{20}$$

After simplifying (20) we get

$$\sum_{n=1}^{N}(x_*(\mathbf{X^T X})^{-1}x_n)y_n = \sum_{n=1}^{N} w_n y_n \tag{21}$$

And, from (21) we can see that $w_n = x_*^T(\mathbf{X^T X})^{-1}x_n$ here w_n is the weighted relation of x_n and x_*, where as in case of K-nearest-neighbor $w_n = \frac{1}{||x_* - x_n||}$. In the case of K-nearest-neighbor, very little weight will be assigned to the points that are far away from x_*, whereas in the case of Linear regression, viewed as nearest-neighbor higher weight assigned to the points that are away from x_*. Also, in the case of Linear Regression viewed as nearest-neighbor while weighting the points, it utilizes the whole training data points, whereas in the case of K-nearest-neighbor only takes the absolute difference between the coordinates of x_n and x_*

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

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION

5

Feature masking as Regularization

Note:- here in this question $\odot$ representing the dot product and w^T denoted the transpose of w

Here, in the case when we are using masking, the loss function is:

$$L(w) = \sum_{n=1}^{N} (y_n - w^T(x_n \odot m_n))^2 \tag{22}$$

where, $L(w)$ is the new loss function, w is the weight vector, y_n is the true target value for nth example, x_n is the input feature for nth example and m_n is the binary mask vector for the nth example, where $m_{nd} \sim \text{Bernoulli}(p)$.

Now we have to minimize the expectation of $\mathbb{E}[L(w)]$

$$\mathbb{E}[L(w)] = \sum_{n=1}^{N} \mathbb{E}[(y_n - w^T(x_n \odot m_n))^2] \tag{23}$$

After squaring the inside term and expanding, we get this

$$\mathbb{E}[L(w)] = \sum_{n=1}^{N} \mathbb{E}\left([y_n^2] - 2y_n w^T(x_n \odot m_n) + (w^T(x_n \odot m_n))^2\right) \tag{24}$$

$$\mathbb{E}[L(w)] = \sum_{n=1}^{N} \mathbb{E}[y_n^2] - 2w^T \mathbb{E}[y_n x_n \odot m_n] + \mathbb{E}[(w^T(x_n \odot m_n))^2] \tag{25}$$

Now if we see $\mathbb{E}[y_n^2]$ is a constant values w.r.t to w, $\mathbb{E}[y_n x_n \odot m_n] = (\mathbb{E}[y_n x_n]) \odot \mathbb{E}[m_n]$ and since $\sim$ Bernoulli(p) so, $(\mathbb{E}[y_n x_n]) \odot \mathbb{E}[m_n] = p \odot \mathbb{E}[y_n x_n]$. Also $\mathbb{E}[\left(w^T(x_n \odot m_n)\right)^2]$ can be written as $\mathbb{E}[w^2 \odot (x_n \odot m_n)^2]$

So, the final equation can be written as

$$\mathbb{E}[L(w)] = \sum_{n=1}^{N} \left(-2w^T p \odot \mathbb{E}[y_n x_n^2] + w^2 \odot (p \odot x_n^2)\right) \tag{26}$$

And the loss in (26) is similar to minimizing the regularized loss

$$\text{Regularized Loss}(w) = \sum_{n=1}^{N} \left(y_n^2 - 2w \odot p \odot \mathbb{E}[y_n x_n^2] + w^2 \odot (p \odot x_n^2)\right) \tag{27}$$

Here, the regularization term is $p \odot x_n^2$, similar to $L2$ Regularization, and smaller p will make more robust Regularization.

Student Name: Nitish Kumar
Roll Number: 19807573
Date: September 15, 2023

QUESTION 6

For method-1(Convex)
The test-set classification accuracy using this method is 46.89%
For method-2(Regress)
The test-set classification accuracy using this method is 58.09% for $\lambda = 0.01$
The test-set classification accuracy using this method is 59.55% for $\lambda = 0.1$
The test-set classification accuracy using this method is 67.39% for $\lambda = 1$
The test-set classification accuracy using this method is 73.28% for $\lambda = 10$
The test-set classification accuracy using this method is 71.68% for $\lambda = 20$
The test-set classification accuracy using this method is 65.08% for $\lambda = 50$
The test-set classification accuracy using this method is 56.47% for $\lambda = 100$
The $\lambda = 10$ gives the best test set accuracy of **73.28%**