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

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 1

The Given optimization problem is-

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

To find the optimal value of $(\hat{w}_c)$ and $(\hat{M}_c)$ (i.e. value on which loss will be minimum) I have to partially differentiate the loss function with respect to (w_c) and (M_c), and equating the results with zero will give me the optimal values for $(\hat{w}_c)$ and $(\hat{M}_c)$ correspondingly.

First partially differentiating given equation w.r.t w_c and equating it to zero-

$$\frac{\partial}{\partial w_c} \left(\text{argmin}_{w_c, M_c} \sum_{x_n : y_n = c} \frac{1}{N_c} (x_n - w_c)^T M_c (x_n - w_c) - \log |M_c| \right) = 0 \tag{2}$$

partially differentiating $\log |M_c|$ w.r.t w_c will results in zero as $\log |M_c|$ is a constant w.r.t w_c,

$$\frac{-2 * M_c}{N_c} \sum_{x_n : y_n = c} (x_n - w_c) = 0 \tag{3}$$

Simplifying the expression,

$$\sum_{x_n : y_n = c} x_n - \sum_{x_n : y_n = c} w_c = 0 \tag{4}$$

Since there are N_c numbers of input from class C so this expression becomes-

$$\sum_{x_n : y_n = c} x_n - N_c * w_c = 0 \tag{5}$$

Further Simplifying,

$$w_c = \frac{1}{N_c} \sum_{x_n : y_n = c} x_n \tag{6}$$

Again partially differentiating given equation w.r.t M_c and equating it to zero-

$$\frac{\partial}{\partial M_c} \left(\sum_{x_n : y_n = c} \frac{1}{N_c} (x_n - w_c)^T M_c (x_n - w_c) - \log |M_c| \right) = 0 \tag{7}$$

Since $\frac{\partial}{\partial M_c} \log |M_c| = (M_c^{-1})^T$ but positive definite matrix is symmetric so $(M_c^{-1})^T = M_c^{-1}$,hence after partially differentiating -

$$\frac{1}{N_c} \sum_{x_n : y_n = c} (x_n - w_c)^T (x_n - w_c) - M_c^{-1} = 0 \tag{8}$$

$$\frac{1}{N_c} \sum_{x_n : y_n = c} (x_n - w_c)^T (x_n - w_c) = M_c^{-1} \tag{9}$$

Pre-multiply by M_c on both sides and simplify to calculate for M_c,

$$M_c * \sum_{x_n : y_n = c} (x_n - w_c)^T (x_n - w_c) = N_c * I_{d*d} \tag{10}$$

$$M_c * \sum_{x_n : y_n = c} \|x_n - w_c\|_2^2 = N_c * I_{d*d} \tag{11}$$

$$M_c = \frac{N_c * I_{d*d}}{\sum_{x_n : y_n = c} \|x_n - w_c\|_2^2} \tag{12}$$

the second part of the question when M_c is reduced to identity matrix then the given optimization expression becomes-

$$(\hat{w_c}, \hat{M_c}) = \text{argmin}_{w_c, M_c} \sum_{x_n : y_n = c} \frac{1}{N_c} (x_n - w_c)^T I (x_n - w_c) - \log |I| \tag{13}$$

simplifying,

$$(\hat{w_c}) = \text{argmin}_{w_c} \sum_{x_n : y_n = c} \frac{1}{N_c} \left((x_n - w_c)^T (x_n - w_c) \right) \tag{14}$$

$$(\hat{w_c}) = \text{argmin}_{w_c} \sum_{x_n : y_n = c} \frac{1}{N_c} \|x_n - w_c\|_2^2 \tag{15}$$

Hence, when M_c is reduced to the identity matrix then the problem becomes the minimization of the sum of l_2 norm of each input(for a class C) w.r.t w_c

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

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 2

One-nearest-neighbor algorithm predicts a class label based on the class label of the nearest training input from the test data point. Since it has access to infinite amounts of training data, in this case, the algorithm will always find the nearest training point having the same label as the test data so the algorithm will approach zero error which is the condition for consistency. Hence, **the one-nearest-neighbor algorithm will be consistent in this setting**.

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

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 3

In Decision tree construction for classification, we split on a feature that gives the highest information gain(high entropy reduction) on splitting. (i.e. a notion for deciding the purity of children nodes, moreover purity refers to the homogeneity of labels present in that node) similarly, for regression using a decision tree, we can **use standard deviation or variance reduction** for ensuring the purity of children nodes having real-valued labels.

how standard deviation or variance quantify the homogeneity/diversity of the set of real-valued labels-
standard deviation (i.e.the square root of the variance) is the measure of the dispersion of values, standard deviation will be zero for a set of completely homogeneous real-valued labels and will be high for completely diverse labels.

standard deviation$(\sigma) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$
variance$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$

Idea for building the decision tree using standard deviation/variance reduction

On each step to decide the feature for splitting:
(i) First I calculate the standard deviation for output labels on that node then split data using every feature that is present(remaining at the corresponding step) and again calculate the standard deviation for output labels.
(ii) subtracting each resulting standard deviation from the standard deviation before the split will give me a standard deviation reduction. (kind of similar to the highest entropy reduction in classification)
(iii) Now I will Split on the feature that is giving the highest standard deviation reduction.

Since the decision tree is a recursive algorithm, to implement a regression model using a decision tree and also to ensure that the model is not over-fitting I will stop splitting further when the standard deviation of output labels is below a predefined threshold (also stop when there is no more feature or very few data points in a node)

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

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 4

For the unregularized linear regression model the solution for $\hat{w}$ is-

$$\hat{w} = (\mathbf{X}^\top \mathbf{X})^{-\mathbf{1}} \mathbf{X}^\top \mathbf{y} \tag{16}$$

and Prediction for an input x_* at test time is written as-

$$y_* = \hat{w}^T x_* = x_*^T \hat{w} \tag{17}$$

Substituting the value of $\hat{w}$ in $y_* = x_n^T \hat{w}$,

$$y_* = x_*^T (\mathbf{X}^\top \mathbf{X})^{-\mathbf{1}} \mathbf{X}^\top \mathbf{y} \tag{18}$$

This expression can also be written as-

$$y_* = \sum_{n=1}^{N} (x_*^T (x_n^\top x_n)^{-1} x_n^\top) y_n \tag{19}$$

OR

$$y_* = \sum_{n=1}^{N} \mathbf{W} y_n \tag{20}$$

The overall equation can be written as-

$$y_* = \sum_{n=1}^{N} w_n y_n \tag{21}$$

Here each $w_n = x_*^T (x_n^\top x_n)^{-1} x_n^\top$ or $\mathbf{W} = x_*^T (\mathbf{X}^\top \mathbf{X})^{-\mathbf{1}} \mathbf{X}^\top$ and by the expression for the w_n , i can say that w_n is dependent on n^{th} training input and test input x_* (also note that to make a prediction y_* all the $(w_1, w_2 ... w_N)$ vectors and all the $(y_1 y_2, ... y_n)$ vectors are needed)

The difference in this w_n and the weights in the weighted K nearest neighbors is that w_n is calculated by n^{th} training input (and all the weights i.e. $(w_1, w_2 ... w_N)$ needed for prediction) but in weighted KNN weight depends on only K nearest input and weights are assigned by the inverse of their distance from the test point. Also in KNN weights are normalized by dividing the sum and that is not the case in calculating w_n.

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 5

Given the new loss function(after masking inputs) is-

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

When $\mathbf{x}$ is dropped s.t. any input dimension is retained with probability p, then the expected value of L(w)-

$$L(w) = E_{R \backsim Bernoulli(N,p)}[L(w)] \tag{23}$$

Replacing $\tilde{x_n}$ by $x_n \odot m_n$ in L(w) and minimizing the objective,

$$\mathbf{L(w)} = \text{argmin}\, w\{E_{R \backsim Bernoulli(n,p)}[\sum_{n=1}^{N} (y_n - \mathbf{w^T}(\mathbf{x_n} \odot \mathbf{m_n}))^2]\} \tag{24}$$

This can be written as-

$$L(w) = \|\mathbf{y} - (\mathbf{M} * \mathbf{X})\mathbf{w}\|^2 \tag{25}$$

We minimize this w.r.t. w , so new objective function will be-

$$\mathbf{L(w)} = \text{argmin}\, w\{ER \backsim Bernoulli(n,p)[\|\mathbf{y} - (\mathbf{M} * \mathbf{X})\mathbf{w}\|^2]\} \tag{26}$$

$$\mathbf{L(w)} = \text{argmin}\, w\{ER \backsim Bernoulli(n,p)[(\mathbf{y} - (\mathbf{M} * \mathbf{X})\mathbf{w})^T(\mathbf{y} - (\mathbf{M} * \mathbf{X})\mathbf{w})]\} \tag{27}$$

Let $\mathbf{b} = \mathbf{y} - (\mathbf{M} \cdot \mathbf{X})\mathbf{w}$ and $\mu = E[R \sim \text{Bernoulli}(n,p)]$. Then, we have $[\mathbf{b}] = \mathbf{y} - p\mathbf{X}\mathbf{w}$.

$$\mathbf{L(w)} = \arg\min_w \left(E[R \sim \text{Bernoulli}(n,p)] \left\|\mathbf{b^T b}\right\|\right) \tag{28}$$

$$\mathbf{L(w)} = \arg\min_w \left(\mu^T\mu + \text{TRACE}\left((\mathbf{b} - \mu)(\mathbf{b} - \mu)^T\right)\right) \tag{29}$$

$$\mathbf{L(w)} = \arg\min_w \left((\mathbf{y} - p\mathbf{X}\mathbf{w})^T(\mathbf{y} - p\mathbf{X}\mathbf{w}) + p(1-p)\text{TRACE}\left((\mathbf{X}\mathbf{w})(\mathbf{X}\mathbf{w})^{\mathbf{T}}\right)\right) \tag{30}$$

$$\mathbf{L(w)} = \arg\min_w \left(\|\mathbf{y} - p\mathbf{X}\mathbf{w}\|^2 + p(1-p)\text{TRACE}\left(\mathbf{X}\mathbf{w}\mathbf{w^T}\mathbf{X^T}\right)\right) \tag{31}$$

$$\mathbf{L(w)} = \arg\min_w \left(\|\mathbf{y} - p\mathbf{X}\mathbf{w}\|^2 + p(1-p)\|(\sqrt{diag\mathbf{X^T}\mathbf{X}}\mathbf{w})\|\right) \tag{32}$$

Now when we compare this with ridge regression objective function

$$\mathbf{L}_{\text{ridge}}(\mathbf{w}) = \arg\min_w \left((y - Xw)^T(y - Xw) + \lambda w^T w\right) \tag{33}$$

$$= \arg\min_w \left(\|y - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2\right) \tag{34}$$

By comparing this with the ridge regression objective we can say that $\lambda\|\mathbf{w}\|^2$ is acting as a regularizer term.

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

Student Name: Prashant Kumar
Roll Number: 231110036
Date: September 15, 2023

QUESTION 6

A **'231110036.zip'** file is submitted with code and readme file.