Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

QUESTION 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

Before moving on to the solution, I will introduce the notations and state the definitions I will be needing for the problem.

- T_A: Tree A, T_B: Tree B
- $IG(T)$: Information gain of tree T
- $MR(T)$: Misclassification rate of tree T
- D: original training dataset, D_t: training subset at node t
- N: total number of training samples, N_t: number of traning samples at node t
- $H(S)$: Entropy of $S \subseteq D$
- Let p denote the parent node, l denote the left node and r the right node
- $p_c(t) := \frac{N_t^c}{N_t}$: probability of a sample belonging to class c at node t

The formal definitions of *misclassification rate* and *information gain* are given by:

$$MR(T) := \sum_{t \in leaves(T)} \frac{N_t}{N} \left(1 - \max_{c \in \{0,1\}} \{p_c(t)\}\right) \tag{1}$$

$$IG(T) := H(D_p) - \left(\frac{N_l}{N} H(D_l) + \frac{N_r}{N} H(D_r)\right) \text{ where } H(D_t) = -\sum_{c \in classes} p_c(t) \log_2(p_c(t)) \tag{2}$$

1. **Misclassification rate**: For T_A, substituting $N = 800, N_l = N_r = 400$, and observing

$$p_0(l) = 300/400 = 3/4, p_1(l) = 100/400 = 1/4$$

$$p_0(r) = 100/400 = 1/4, p_1(r) = 300/400 = 3/4$$

Now, we substitute these values in equation 1 and get

$$MR(T_A) = (1/2)[1/4 + 1/4] = 1/4 = 0.25$$

Similarly, working out for T_B yields

$$MR(T_B) = (3/4)[1/3 + 0] = 1/4 = 0.25$$

$$\boxed{MR(T_A) = 0.25, \quad MR(T_B) = 0.25}$$

Thus, we observe that the misclassification rate for both the trees is the same. Thus, the two trees are incomparable on the metric of misclassification rate.

2. **Information gain**: First, we compute $H(D_p), H(D_l)$ and $H(D_r)$ using equation 2 for tree T_A.

$$H(D_p) = -\frac{1}{2} log_2(\frac{1}{2}) - \frac{1}{2} log_2(\frac{1}{2}) = 1$$

$$H(D_l) = -\frac{3}{4} log_2(\frac{3}{4}) - \frac{1}{4} log_2(\frac{1}{4}) = 0.81125$$

$$H(D_r) = -\frac{1}{4}log_2(\frac{1}{4}) - \frac{3}{4}log_2(\frac{3}{4}) = 0.81125$$

Further, using the equation for information gain in equation 2, we compute the information gain for tree T_A

$$IG(T_A) = 1 - \left(\frac{400}{800}\right) 0.81125 - \left(\frac{400}{800}\right) 0.81125$$

$$\boxed{\therefore IG(T_A) = 0.18875}$$

Similarly, working out for tree T_B, we get

$$\boxed{\therefore IG(T_B) = 0.3115}$$

Thus, we note that $IG(T_B) > IG(T_A)$ indicating that data split in the latter results in a better decision tree in terms of the metric (information gain) based on entropy.

3. **Comparison**: We get different answers in part 1 and 2. This is indicating that entropy-based information gain is a better metric than misclassification rate since it is evident that the split is more *pure* in the tree T_B relative to that of tree T_A and the same is reflected by the information gain. This, however, is not the case with misclassification rate which assigns the same values for both the trees.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

QUESTION 2

Claim: 1-nearest neighbor (1-NN) classification is a *consistent* classification method.
Justification:Suppose we consider 1-nearest neighbor (1-NN) classification trained on infinite number of training examples. Note that k-NN has error rate not worse than twice that of Bayes optimal classifier.

$$E_{k-NN} \leq 2E_{BO} \rightarrow 0 \Rightarrow E_{k-NN} \rightarrow 0$$

Note that the Bayes optimal error rate goes to 0 in a noise free setting. Thus, the error rate for 1-NN also go to zero. Another way to look at it is that since 1-NN considers distance from just one nearest sample point at test time, it is prone to overfitting and gives almost zero training error. Now for a given test example, since we assume that the training set is infinite, it essentially implies that the test example will have some training example in very close proximity and thus the error rate would be very low, close to 0. Thus, 1-NN is a consistent classification method.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

QUESTION 3

Solution to the linear regression problem is given by $\hat{w} = (X^T X)^{-1} X^T y$. For a test example x_*, we can predict the output as follows:

$$f(x_*) = \hat{w}^T x_*$$

$$\therefore f(x_*) = ((X^T X)^{-1} X^T y)^T x_*$$
$$\therefore f(x_*) = y^T (X^T)^T [(X^T X)^{-1}]^T x_*$$
$$\therefore f(x_*) = y^T X [(X^T X)^T]^{-1} x_* = y^T X [X^T X]^{-1} x_*$$
$$\therefore f(x_*) = \langle X(X^T X)^{-1} x_*, y \rangle = \sum_{n=1}^{N} w_n y_n$$

where w_n is the n^{th} component of the term $X(X^T X)^{-1} x_*$. Let $A := X(X^T X)^{-1}$. Let us denote this weight vector coressponding to linear regression as w^{LR} and that of weighted kNN as w^{kNN}. Our goal is to find/interpret some sort of a relation between these two. As given, we know:

$$w_n^{kNN} = \frac{1}{d(x_n, x_*)}$$

Now, we will try to interpret what w_n^{LR} means. Note that $w^k NN$ is a sort of a similarity between the input data matrix X and x_*. Similarly, observe that $w^{LR} = X(X^T X)^{-1} x_*$ and thus we have

$$w_n^{LR} = x_n^T (X^T X)^{-1} x_*$$

Thus, w_n^{LR} is also a sort of a similiarity between x_n and x_* which is weighted by the matrix $(X^T X)^{-1}$ much like the Mahalanobois matrix.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

QUESTION 4

The usual l_2 regularized least squares regression objective can be written as

$$\mathcal{L}(w) = \sum_{n=1}^{N}(y_n - w^T x_n)^2 + \frac{\lambda}{2}w^T w$$

Now, the first term can be re-written in terms of input data matrix and output label vector. Also, considering different hyperparameters for each feature, say λ_i for feature i, the objective function can be re-stated as

$$\mathcal{L}(w) = (y - Xw)^T(y - Xw) + \sum_{i=1}^{N}\frac{\lambda_i}{2}w_i^2$$

We need to compute $\frac{\partial \mathcal{L}(w)}{\partial w}$. We already know the derivative of the first term from analysis of least squares regression. For the second term, w.r.t the feature i, we have

$$\frac{\partial(\sum_{i=1}^{N}\frac{\lambda_i}{2}w_i^2)}{\partial w_i} = 2 * (\lambda_i/2)w_i = \lambda_i w_i$$

Thus, the gradient w.r.t w can be computed:

$$\frac{\partial(\sum_{i=1}^{N}\frac{\lambda_i}{2}w_i^2)}{\partial w} = \begin{bmatrix}\lambda_1 w_1 & \lambda_2 w_2 & \lambda_3 w_3 & \dots & \lambda_N w_N\end{bmatrix}^T$$

$$\therefore \frac{\partial(\sum_{i=1}^{N}\frac{\lambda_i}{2}w_i^2)}{\partial w} = \begin{bmatrix}\lambda_1 & 0 & 0 & \dots & 0\\ 0 & \lambda_2 & 0 & \dots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \dots & \lambda_N\end{bmatrix}\begin{bmatrix}w_1 & w_2 & w_3 & \dots & w_N\end{bmatrix}^T = diag(\bar{\lambda})w$$

where $diag(v)$ is a diagonal matrix for vector v and $\bar{\lambda} = \begin{bmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \dots & \lambda_N\end{bmatrix}$. Now, equating $\frac{\partial \mathcal{L}(w)}{\partial w} = 0$, we get

$$0 = X^T y - (X^T X + diag(\bar{\lambda}))w$$

$$\therefore (X^T X + diag(\bar{\lambda}))w = X^T y$$

$$\boxed{\therefore w = (X^T X + diag(\bar{\lambda}))^{-1} X^T y}$$

Thus, we get the closed form solution as $\hat{w} = (X^T X + diag(\bar{\lambda}))^{-1} X^T y$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

QUESTION 5

Consider the multi-output regression in which each output $y_n \in \mathbb{R}^m$ is a real-valued vector, rather than a scalar. Assuming a linear model, we can model the outputs as $Y = XW$, where X is the usual input data matrix, Y is the response matrix of dimensions $N \times M$, W is the weight matrix to be learned of dimensions $D \times M$. The least squares objective in this setting can be stated as

$$\mathcal{L}(W) = \sum_{n=1}^{N}\sum_{m=1}^{M}(y_n m - w_m^T x_n)^2$$

$$\therefore \mathcal{L}(W) = \sum_{m=1}^{M}\left(\sum_{n=1}^{N}(y_n m - w_m^T x_n)^2\right)$$

$$\therefore \mathcal{L}(W) = \sum_{m=1}^{M}\left(\sum_{n=1}^{N}(y_m - Xw_m)^T(y_m - Xw_m)\right)$$

$$\therefore \mathcal{L}(W) = TRACE\left((Y - XW)^T(Y - XW)\right)$$

Here, we will assume that the weight matrix W can be written as a product of two matrices, i.e., $W = BS$ where B is $D \times K$ and S is $K \times M$ (assume $K < \min\{D, M\}$). We further assume that B is known and we would like to estimate S. Thus, estimating S can be stated as an optimix=zation problem as follows:

$$\hat{S} = \arg\min_S \quad TRACE[(Y - XW)^T(Y - XW)]$$

The loss function can be stated a function of S as follows:

$$\mathcal{L}(S) = tr[(Y-XBS)^T(Y-XBS)] = tr[Y^TY]-tr[(XBS)^TY]-tr[Y^TXBS]+tr[(XBS)^TXBS]$$

Now, we need to compute $\frac{\partial\mathcal{L}(S)}{\partial S}$ and equate it to 0 to find the optimal solution for S.

$$\frac{\partial\mathcal{L}(S)}{\partial S} = \frac{\partial tr(Y^TY)}{\partial S} - \frac{\partial tr[(XBS)^TY]}{\partial S} - \frac{\partial tr[(Y^TXB)S]}{\partial S} + \frac{\partial tr[(XBS)^TXBS]}{\partial S}$$

$$\therefore \frac{\partial\mathcal{L}(S)}{\partial S} = 0 - \frac{\partial tr[S^T(B^TX^TY)]}{\partial S} - \frac{\partial tr[(Y^TXB)S]}{\partial S} + \frac{\partial tr[S^T(B^TX^TXB)S]}{\partial S}$$

$$\therefore \frac{\partial\mathcal{L}(S)}{\partial S} = -B^TX^TY - (Y^TXB)^T + (B^TX^TXB)S + (B^TX^TXB)^TS$$

$$\therefore 0 = 2B^TX^TY + 2(B^TX^TXB)S$$

$$\therefore (B^TX^TXB)S = B^TX^TY$$

$$\therefore S = (B^TX^TXB)^{-1}B^TX^TY$$

$$\boxed{\therefore S = ((XB)^TXB)^{-1}(XB)^TY = (\bar{X}^T\bar{X})^{-1}\bar{X}Y}$$

where $\bar{X} = XB$ i.e., the input matrix transformed by the product with B. Thus, the form of the solution is identifical to the solution of standard multi-output regression, but uses a transformed version of the inputs X.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Piyush Bagad
Roll Number: 150487
Date: August 31, 2018

QUESTION 6

1. **Method 1: Modeling unseen means as convex combinations of seen means**
   The test accuracy obtained for prototype based classification is 0.4700647

   $\boxed{\text{Test accuracy : 0.4700647}}$

2. **Method 2: Modeling unseen means through multi-output regression**
   The test accuracies for prototype based classification for varying lambdas are noted in table 2 We observe that the highest test accuracy is obtained for $\lambda = 10$.

λ	**Test accuracy**
0.01	0.5654
0.1	0.5848
1.0	0.6710
10	**0.7338**
20	0.7158
50	0.6474
100	0.5731

Table 1: Test accuracies vs lambdas