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

QUESTION 1

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

Solution 1:

Misclassification rate for Tree 1 = $\frac{(100+100)}{800} = \frac{1}{4}$.
Misclassification rate for Tree 2 = $\frac{(200+0)}{800} = \frac{1}{4}$.
The misclassification rates are equal for the two trees.

For Tree 1:
$H(s) = \log(2) = 1$
$H(s_1) = \frac{3}{4} * \log(\frac{4}{3}) + \frac{1}{4} * \log(4) = 0.811$
$H(s_2) = \frac{3}{4} * \log(\frac{4}{3}) + \frac{1}{4} * \log(4) = 0.811$
$IG = H(s) - \frac{|s_1|}{s} * H(s_1) - \frac{|s_2|}{s} * H(s_2) = 1 - \frac{1}{2} * 0.811 - \frac{1}{2} * 0.811 \ = 0.188$

For Tree 2:
$H(s) = \log(2) = 1$
$H(s_1) = \frac{1}{3} * \log(3) + \frac{2}{3} * \log(\frac{3}{2}) = 0.918$
$H(s_2) = (-1) * 0 * \log(0) - 1 * \log(1) = 0$
$IG = 1 - \frac{2}{3} * 0.918 - 0 = 0.388$

So the information gain of Tree 2 is greater than Tree 1 , hence on the basis of information gain we can say that Tree 2 is better than Tree 1.
Although the missclassification rate for both the trees are same, we would prefer Tree 2 since it leads to pure node and same thing is indicated by the Information Gain values of the trees. So we could say that Information Gain through entropy is better way to grow our tree than missclassification rate.

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

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION 2

Solution:2

The nearest neighbour classifier partitions the input space into closed cells consisting of all points which are closer to a given training input x_i than to any other training input. The label of a test input which falls in a given cell is same as the label of the training input to which the cell belongs. If we increase the number of training inputs the size of cells will become smaller and smaller and in case of infinite training inputs the cells will be reduced to a point.
So any test input will concide with a training input and hence it will be labelled correctly. So one-nearest neighbour algorithm is consistent.

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

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION
3

Solution 3:

The prediciton at a test input $\mathbf{x}_*$ is given by:
$f(\mathbf{x}_*) = \hat{\mathbf{w}}^T\mathbf{x}_* = \mathbf{x}_*^T\hat{\mathbf{w}} = \mathbf{x}_*^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
Let us define a new matrix $\mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ then the equation can be written as:
$f(\mathbf{x}_*) = \mathbf{x}_*^T\mathbf{B}\mathbf{y}$
If we consider $\boldsymbol{w} = \mathbf{x}_*^T\mathbf{B}$ then $\boldsymbol{w}$ is a row vector of dimensions 1 X N and the equation can be written as:
$f(\mathbf{x}_*) = \boldsymbol{w}\mathbf{y} = \sum_{n=1}^{N} w_n y_n$
Here w_n is the dot product of the vector $\mathbf{x}_*$ and the n^{th} column vector of $\mathbf{B}$. This dot product measures how similar is the input vector $\mathbf{x}_*$ to the n^{th} training example. If the input vector is similar to the i^{th} training example then the value of dot product will be high and hence the value of w_i will be high.
The difference between these w_n and those defined in weighted version of the K-nearest neighbour problem is that there the similarity is measured using the euclidean distance between the input $\mathbf{x}_*$ and the training point $\mathbf{x}_n$ but in this case the similarity is a kind of cosine similarity between input $\mathbf{x}_*$ and training input $\mathbf{x}_n$.

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

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION 4

Solution 4:
In order to regularize differently with respect to each weight entry w_i we can construct a diagonal matrix $\boldsymbol{\lambda}$ which is defined as:

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix}$$

The loss function can be written as:
$L(\mathbf{w}) = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \mathbf{w}^T\boldsymbol{\lambda}\mathbf{w}$
This can also be written in the following form:
$L(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \mathbf{w}^T\boldsymbol{\lambda}\mathbf{w}$
$L(\mathbf{w}) = (\mathbf{y}^T - \mathbf{w}^T\mathbf{X}^T)(\mathbf{y} - \mathbf{X}\mathbf{w}) + \mathbf{w}^T\boldsymbol{\lambda}\mathbf{w}$
$L(\mathbf{w}) = \mathbf{y}^T\mathbf{y} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\boldsymbol{\lambda}\mathbf{w}$
$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 0 - \mathbf{X}^T\mathbf{y} - (\mathbf{y}^T\mathbf{X})^T + (\mathbf{X}^T\mathbf{X} + (\mathbf{X}^T\mathbf{X})^T)\mathbf{w} + (\boldsymbol{\lambda} + \boldsymbol{\lambda}^T)\mathbf{w}$
Since $\boldsymbol{\lambda}$ is a diagonal matrix $\boldsymbol{\lambda} = \boldsymbol{\lambda}^T$, so we get,
$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 2(\mathbf{X}^T\mathbf{y}) - 2(\mathbf{X}^T\mathbf{X} + \boldsymbol{\lambda})\mathbf{w}$
Setting the derivative to zero we get,
$\mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X} + \boldsymbol{\lambda})\mathbf{w}$
Which finally gives us
$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \boldsymbol{\lambda})^{-1}(\mathbf{X}^T\mathbf{y})$

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

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION 5

Solution 5:

$\hat{S} = \arg\min_S \text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})]$
The expression for loss function can be given as:
$L(\mathbf{S}) = \text{TRACE}[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})]$
$L(\mathbf{S}) = \text{TRACE}[(\mathbf{Y}^T - \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{XBS})]$
$L(\mathbf{S}) = \text{TRACE}[(\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{XBS} - \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y} + \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS})]$
$L(\mathbf{S}) = \text{TRACE}[\mathbf{Y}^T\mathbf{Y}] - \text{TRACE}[\mathbf{Y}^T\mathbf{XBS}] - \text{TRACE}[\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}] + \text{TRACE}[\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}]$
$\frac{\partial L(\mathbf{S})}{\partial \mathbf{S}} = 0 - \mathbf{B}^T\mathbf{X}^T\mathbf{Y} - (\mathbf{Y}^T\mathbf{XB})^T + (\mathbf{B}^T\mathbf{X}^T\mathbf{XBS} + (\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^T\mathbf{S})$
$\frac{\partial L(\mathbf{S})}{\partial \mathbf{S}} = 2(\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}) - 2(\mathbf{B}^T\mathbf{X}^T\mathbf{Y})$
Setting $\frac{\partial L(\mathbf{S})}{\partial \mathbf{S}} = 0$ we get

$(\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}) = (\mathbf{B}^T\mathbf{X}^T\mathbf{Y})$
Which finally gives us
$\mathbf{S} = (\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^{-1}(\mathbf{B}^T\mathbf{X}^T\mathbf{Y})$
If we define another matrix $\mathbf{X}_* = \mathbf{XB}$ then the solution to our problem reduces to:
$\mathbf{S} = (\mathbf{X}_*^T\mathbf{X}_*)^{-1}(\mathbf{X}_*^T\mathbf{Y})$
The solution in case of standard multi-output regression is given as:
$\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$
We see that both the solutions are very similar , except the fact that instead of using input vector $\mathbf{X}$ we are using a transformed input vector $\mathbf{X}_* = \mathbf{XB}$.

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

Student Name: Aman Tiwari
Roll Number: 160094
Date: April 18, 2019

QUESTION
6

Solution 6:

Classification Accuracy for method 1 = 0.468932
Classification Accuracy for different lambda values for method 2:
$\lambda = 0.01$ Accuracy = 0.580906
$\lambda = 0.1$ Accuracy = 0.595469
$\lambda = 1.0$ Accuracy = 0.673948
$\lambda = 10.0$ Accuracy = 0.732848
$\lambda = 20.0$ Accuracy = 0.716828
$\lambda = 50.0$ Accuracy = 0.650809
$\lambda = 100.0$ Accuracy = 0.564725

Maximum accuracy is 0.732848 for $\lambda = 10.0$