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

**QUESTION 1**

*Student Name:* Subham Kumar
*Roll Number:* 160707
*Date:* February 23, 2019

---

**1.Misclassification rate**
Misclassification rate in case of tree model $\mathcal{A}$ is(Assuming a leaf node to belong to a class having majority of data points of that class):
number of misclassified data points at left leaf node = 100
number of misclassified data points at right leaf node = 100
Total number of data points = 800
Hence Misclassification Rate = 200/800 = **0.25**
Similarily,Misclassification Rate in case of tree model $\mathcal{B}$ is:
number of misclassified data points at left leaf node = 200
number of misclassified data points at right leaf node = 0
Total number of data points = 800
Hence Misclassification Rate = 200/800 = **0.25**
**2.Calculation of Information Gain**

$$IG = H(S) - \frac{|S_1|}{|S|}H(S_1) - \frac{|S_2|}{|S|}H(S_2)$$

where $H(S) = -\sum_{c \epsilon C} p_c \log_2 p_c$,C is the number of classes,$S_1$ and $S_2$ are the splits of S.
Information Gain in case of tree model $\mathcal{A}$ considering the splits $S_1$(300,100) and $S_2$(100,300) of set S(400,400) is given as: In this case H(S)=1 as $p_{c=0} = p_{c=1} = \frac{1}{2}$.
H($S_1$)=$-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.8112$; H($S_2$)=$-\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.8112$
Hence, $IG_A = 1 - \frac{400}{800} * 0.8112 - \frac{400}{800} * 0.8112$=**0.1887**
Information Gain in case of tree model $\mathcal{B}$ considering the splits $K_1$(200,400) and $K_2$(200,0) of set S(400,400) is given as:
H($K_1$)=$-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.9182$; H($K_2$)=$-1\log_2 1 - 0\log_2 0 = 0$
Hence, $IG_B = 1 - \frac{600}{800} * 0.9182 - 0 =$ **0.3112**.
Split in case of tree model $\mathcal{A}$ has higher entropy than tree model $\mathcal{B}$ which means the former has more uncertainty on unseen data and hence later is better predictor.
**3.**Yes,answers to (1) and (2) are different.The first case simply gives the information regarding the training data and comes out to be same for both the tree models whereas in second case we have some uncertainty about the prediction of unseen data and are different for both the models.So second one seems to be a better metric for comparision of the two tree models.

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

*Student Name:* Subham Kumar
*Roll Number:* 160707
*Date:* February 23, 2019

# QUESTION 2

As the number of points in training dataset is infinite it would cover the entire input space.So when a test point arrives it would be one of them and since every training points are labelled correctly so the error on the test input will be zero and hence the error rate.So its error rate approaches bayes optimal which is zero in this case.Hence one-nearsest-neighbor is consistent in this setting.

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

*Student Name:* Subham Kumar
*Roll Number:* 160707
*Date:* February 23, 2019

# QUESTION 3

Given the solution $\hat{\mathbf{w}}= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ for unregularized linear regression we can predict the output for a test point $\mathbf{x}_*$ as $f(\mathbf{x}_*) = \hat{\mathbf{w}}^T\mathbf{x}_*$.

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

Here $\mathbf{y}^T$ is a $1 \times N$ matrix whose each entry can be denoted as $y_n$ and $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_*$ is $N \times 1$ matrix denoted as $\mathbf{w}$ whose each entry can be represented as $w_n$ $\forall n \epsilon \{1,2,....,N\}$.Note that here each component of $\mathbf{w}$ gives some sort of similarity between $\mathbf{x}_*$ and $\mathbf{x_n}$.

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

Hence $f(\mathbf{x}_*)$ can be written as

$$f(\mathbf{x}_*) = \mathbf{y}^T\mathbf{w}$$
$$= \sum_{i=1}^{N} y_n w_n$$

The weights $w_n$ in this case are dependent on all the training points and gives some kind of similarity between $\mathbf{x_n}$ and $\mathbf{x}_*$ whereas in case of KNN it is simply dependent on $\mathbf{x_n}$ or more precisely on the distance between $\mathbf{x}_*$ and $\mathbf{x_n}$.

**Expression for weight in KNN:**

$$w_n = \frac{1}{\sqrt{(\mathbf{x_n} - \mathbf{x}_*)^T(\mathbf{x_n} - \mathbf{x}_*)}}$$

$$f(\mathbf{x}_*) = \sum_{i=1}^{N} y_n w_n$$

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

*Student Name:* Subham Kumar
*Roll Number:* 160707
*Date:* February 23, 2019

# QUESTION 4

The usual $l_2$ regularized least square regression objective can be written as:

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_n - \mathbf{w}^T \mathbf{x_n}) + \lambda \mathbf{w}^T \mathbf{w}$$

$$= \sum_{i=1}^{N} (y_n - \mathbf{w}^T \mathbf{x_n}) + \lambda \sum_{d=1}^{D} w_d^2$$

Clearly in this case the extent of regularization for each entry $w_d$ is same and controlled by $\lambda$.To make this extent of $l_2$ regularization different for each entry $w_d$ let us replace $\lambda$ with $\lambda_d$ $\forall$ d $\epsilon$ {1,2,....,D} and $\lambda_d > 0$.Now our new objective function looks like:

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_n - \mathbf{w}^T \mathbf{x_n}) + \sum_{d=1}^{D} \lambda_d w_d^2$$

$$= \sum_{i=1}^{N} (y_n - \mathbf{w}^T \mathbf{x_n}) + \mathbf{w}^T \hat{\lambda} \mathbf{w}$$

$$= (\mathbf{Y} - \mathbf{X}\mathbf{w})^T (\mathbf{Y} - \mathbf{X}\mathbf{w}) + \mathbf{w}^T \hat{\lambda} \mathbf{w}$$

where $\hat{\lambda}$ is a $D \times D$ diagonal matrix with $\hat{\lambda}_{ii} = \lambda_i$ $\forall$ i $\epsilon$ {1,2,....,D}, $\mathbf{Y}$ is a $N \times 1$ response matrix,$\mathbf{X}$ is a $N \times D$ feature matrix and $\mathbf{w}$ is a $D \times 1$ weight matrix.
For closed form solution taking gradient of $L(\mathbf{w})$ with respect to $\mathbf{w}$ and setting it to zero:

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} + 2\hat{\lambda}\mathbf{w}$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = 0$$
$$\implies -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} + 2\hat{\lambda}\mathbf{w} = 0$$
$$\implies \mathbf{X}^T\mathbf{Y} = (\mathbf{X}^T\mathbf{X} + \hat{\lambda})\mathbf{w}$$
$$\implies \mathbf{w} = (\mathbf{X}^T\mathbf{X} + \hat{\lambda})^{-1}\mathbf{X}^T\mathbf{Y}$$

This is the required closed form expression for the weight vector $\mathbf{w}$.

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

*Student Name:* Subham Kumar
*Roll Number:* 160707
*Date:* February 23, 2019

# QUESTION 5

Given squared loss function for multi-output regression is:

$$L(\mathbf{w}) = \sum_{n=1}^{N}\sum_{m=1}^{M}(y_{nm} - \mathbf{w}_m^T\mathbf{x}_n)^2$$

Now consider the matrix $(\mathbf{Y}-\mathbf{XW})$.Here $\mathbf{Y}$ is $N\times M$ response matrix,$\mathbf{X}$ is $N\times D$ feature vector and $\mathbf{W}$ is $D\times M$ weight vector.Its $(i,j)^{th}$ entry will be $(y_{ij} - \mathbf{w}_j^T\mathbf{x}_i)$ and hence the $(j,i)^{th}$ entry of $(\mathbf{Y}-\mathbf{XW})^T$ $\forall i\epsilon\{1,2,..,n\}$ and $j\epsilon\{1,2,...,m\}$.Finally $(j,j)^{th}$ entry of $(\mathbf{Y}-\mathbf{XW})^T(\mathbf{Y}-\mathbf{XW})$ will be $\sum_{i=1}^{n}(y_{ij} - \mathbf{w}_j^T\mathbf{x}_i)^2$.Hence Trace of this matrix will precisely give us the $L(\mathbf{w})$.
Replacing $\mathbf{W}$ with $\mathbf{BS}$, where $\mathbf{B}$ is $D\times K$ and $\mathbf{S}$ is $K\times M$ matrix, will give us the following objective function:

$$L(\mathbf{B},\mathbf{S}) = TRACE[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})]$$

Taking $\mathbf{B}$ fixed for the context, our optimization problem reduces to:

$$\hat{\mathbf{S}} = \arg\min_{\mathbf{S}} TRACE[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})]$$

This $\hat{\mathbf{S}}$ can be calculated by taking gradient of $L(\mathbf{B},\mathbf{S})$ w.r.t $\mathbf{S}$ and then setting it to zero.

$$\nabla_{\mathbf{S}}L(\mathbf{B},\mathbf{S}) = \frac{\partial(TRACE[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})])}{\partial\mathbf{S}}$$

$$= \frac{\partial(TRACE[(\mathbf{Y}^T-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T)(\mathbf{Y}-\mathbf{XBS})])}{\partial\mathbf{S}}$$

$$= \frac{\partial(TRACE[\mathbf{Y}^T\mathbf{Y}-\mathbf{Y}^T\mathbf{XBS}-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}+\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}])}{\partial\mathbf{S}}$$

$$= \frac{\partial(TRACE[-\mathbf{Y}^T\mathbf{XBS}])}{\partial\mathbf{S}} + \frac{\partial(TRACE[-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}])}{\partial\mathbf{S}} + \frac{\partial(TRACE[\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}])}{\partial\mathbf{S}}$$

$$= -(\mathbf{Y}^T\mathbf{XB})^T - (\mathbf{B}^T\mathbf{X}^T\mathbf{Y}) + (\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}) + (\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XB})^T$$

$$= -2\mathbf{B}^T\mathbf{X}^T\mathbf{Y} + 2\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}$$

$$\nabla_{\mathbf{S}}L(\mathbf{B},\mathbf{S}) = 0$$

$$\implies \mathbf{B}^T\mathbf{X}^T\mathbf{XBS} = \mathbf{B}^T\mathbf{X}^T\mathbf{Y}$$

$$\implies \hat{\mathbf{S}} = ((\mathbf{XB})^T(\mathbf{XB}))^{-1}(\mathbf{XB})^T\mathbf{Y}$$

The form of solution for $\hat{\mathbf{S}}$ is identical to the solution for $\hat{\mathbf{W}}$ in vanilla multioutput regression where the solution is:

$$\hat{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Clearly the later solution for $\mathbf{W}$ seems identical to the former one except that $\mathbf{X}$ is replaced by $\mathbf{XB}$ in former case.
**Bonus:**Taking $\mathbf{S}$ fixed for this context,our optimization problem reduces to:

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} TRACE[(\mathbf{Y}-\mathbf{XBS})^T(\mathbf{Y}-\mathbf{XBS})]$$

This $\hat{\mathbf{B}}$ can be calculated by taking gradient of $L(\mathbf{B}, \mathbf{S})$ w.r.t $\mathbf{B}$ and then setting it to zero.

$$\nabla_{\mathbf{B}} L(\mathbf{B}, \mathbf{S}) = \frac{\partial(TRACE[(\mathbf{Y} - \mathbf{XBS})^T(\mathbf{Y} - \mathbf{XBS})])}{\partial \mathbf{B}}$$

$$= \frac{\partial(TRACE[(\mathbf{Y}^T - \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T)(\mathbf{Y} - \mathbf{XBS})])}{\partial \mathbf{B}}$$

$$= \frac{\partial(TRACE[\mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{XBS} - \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y} + \mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}])}{\partial \mathbf{B}}$$

$$= \frac{\partial(TRACE[-\mathbf{Y}^T\mathbf{XBS}])}{\partial \mathbf{B}} + \frac{\partial(TRACE[-\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{Y}])}{\partial \mathbf{B}} + \frac{\partial(TRACE[\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{XBS}])}{\partial \mathbf{B}}$$

$$= -(\mathbf{Y}^T\mathbf{X})^T\mathbf{S}^T - \mathbf{X}^T\mathbf{Y}\mathbf{S}^T + \mathbf{X}^T\mathbf{XBSS}^T + (\mathbf{S}^T\mathbf{B}^T\mathbf{X}^T\mathbf{X})^T\mathbf{S}^T$$

$$= -2\mathbf{X}^T\mathbf{Y}\mathbf{S}^T + 2\mathbf{X}^T\mathbf{XBSS}^T$$

$$\nabla_{\mathbf{B}} L(\mathbf{B}, \mathbf{S}) = 0$$

$$\implies \mathbf{X}^T\mathbf{XBSS}^T = \mathbf{X}^T\mathbf{YS}^T$$

$$\implies \hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{YS}^T(\mathbf{SS}^T)^{-1}$$