3.1 Problem 1

Assume we have 3 points, i.e., (1,2), (2,1), (3,2), in a 2-D Euclidean space. We want to fit a line which is $y = w_0 + w_1 x$ with respect to these 3 points. Derive the optimal solution for w_0 and w_1 with mean square error (MSE) loss. Show your steps for full score. (8pts)

$$\hat{y} = wx = (w_0\ w_1)\cdot\begin{pmatrix}1 & 1 & 1\\ 1 & 2 & 3\end{pmatrix}$$

$$y = \begin{pmatrix}2\\1\\2\end{pmatrix}$$

$$MSE = \frac{1}{3}\sum_{i=1}^{3}(\hat{y}_i - y_i)^2$$

$$= \frac{(w_0+w_1-2)^2 + (w_0+2w_1-1)^2 + (w_0+3w_1-2)^2}{3}$$

$$\begin{cases}\frac{\partial MSE}{\partial w_0} = 2w_0 + 4w_1 - \frac{10}{3}\\ \frac{\partial MSE}{\partial w_1} = 4w_0 + \frac{28}{3}w_1 - \frac{20}{3}\end{cases} \qquad \begin{cases}\frac{\partial MSE}{\partial w_0} = 0\\ \frac{\partial MSE}{\partial w_1} = 0\end{cases} \Rightarrow \begin{cases}w_1 = 0\\ w_0 = \frac{5}{3}\end{cases}$$

3.2 Problem 2

Assume we have 4 points, i.e., (-2,1), (-1,0), (1,0), (2,2), in a 2-D Euclidean space. We want to fit a 2nd order polynomial function $y = w_0 + w_1 x + w_2 x^2$ with respect to these 4 points. Remember the closed form solution would be $\hat{w} = argmin(\boldsymbol{Hw} - \boldsymbol{y})^T(\boldsymbol{Hw} - \boldsymbol{y})$, where $\boldsymbol{w} = [w_0, w_1, w_2]^T$. Write down what would be $\boldsymbol{H}$ and $\boldsymbol{y}$. (7pts)

$$H = \begin{pmatrix}1 & -2 & (-2)^2\\ 1 & -1 & (-1)^2\\ 1 & 1 & 1^2\\ 1 & 2 & 2^2\end{pmatrix} = \begin{pmatrix}1 & -2 & 4\\ 1 & -1 & 1\\ 1 & 1 & 1\\ 1 & 2 & 4\end{pmatrix} \qquad y = \begin{pmatrix}1\\0\\0\\2\end{pmatrix}$$

3.3 Problem 3

Assume we have 6 points showing 6 observations in a 2-D Euclidean space as below:

Observation	X_1	X_2	Y
1	0	2	class 1
2	0	3	class 1
3	1	3	class 1
4	1	1	class 2
5	2	1	class 2
6	2	2	class 2

Using a maximum margin classifier, determine the optimal separating hyperplane and give an equation for it, and determine which observations are the support vectors. (7pts)

$$G(x^{(i)}) = \frac{y^{(i)}(w \cdot x^{(i)} + b)}{\|w\|}$$

$$\max_{w,b} G \quad \text{s.t.}: \frac{y^{(i)}(w \cdot x^{(i)} + b)}{\|w\|} \geq G$$

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{s.t.}: y^{(i)}(w \cdot x^{(i)} + b) \geq 1 \text{ for } i \in [1, 6]$$

hyperplane: $w \cdot x + b = 0$ $\quad w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$

$$y(w \cdot x + b) = (1\ 1\ 1\ -1\ -1\ -1) \cdot \left(\begin{pmatrix} 0 & 2 \\ 0 & 3 \\ 1 & 3 \\ 1 & 1 \\ 2 & 1 \\ 2 & 2 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} b \\ b \\ b \\ b \\ b \\ b \end{pmatrix} \right)$$

$$\min_{w,b} \frac{1}{2}\|w\|^2, \text{ s.t.}: \begin{cases} 2w_2 + b \geq 1 \\ 3w_2 + b \geq 1 \\ w_1 + 3w_2 + b \geq 1 \\ -w_1 - w_2 - b \geq 1 \\ -2w_1 - w_2 - b \geq 1 \\ -2w_1 - 2w_2 - b \geq 1 \end{cases} \Rightarrow \begin{cases} w_1 = -1 \\ w_2 = 1 \\ b = -1 \end{cases}$$

observation $\{1, 3, 4, 6\}$ are support vector, since they satisfy $y^{(i)}(w \cdot x^{(i)} + b) = 1$

3.4 Problem 4

Describe at least 4 differences or similarities between bagging and boosting. (8pts)

① boosting firstly use one classifier to train and then use another classifier to train based on the previous one and so on, whereas the bagging can train all the classifiers simultaneously.

② The data set in bagging is obtained by random selecting, which means the data set may contain the same data and some data in original set may not be selected. Nonetheless, in boosting, all the data will be used.

③ The weight of data in boosting will be adjusted during training but bagging use the data with the same weight.

④ In classification, all the classifiers have the same weight, but in boosting, different classifiers may have different weights.

3.5 Problem 5

LOOCV is a statistical cross-validation method often used when our dataset is quite small. It stands for "Leave-One-Out Cross Validation". If we apply LOOCV method with some model on a dataset with N samples, we partition the dataset N times, in N iterations: in iteration i we partition the dataset into a train set by removing the i-th sample from original dataset and use the test set with only the i-th sample to validate our model.

The LOOCV estimate of the test Mean-Square-Error(MSE) is defined as:

$$CV_{(n)} = \frac{1}{N}\sum_{i=1}^{N} MSE_i$$

Where $MSE_i = \frac{(y_i-\hat{y}_i)^2}{1} = (y_i - \hat{y}_i)^2$. y_i is the true label value of the i-th sample, and $\hat{y}_i$ is the predicted label value of the i-th sample.

Even for simple models like ordinary least square linear regression, LOOCV could take a significant amount of computation. Luckily, a "shortcut" exists for LOOCV on ordinary least square linear regression. For example, for a univariate least square linear regression model we obtain

$$CV_{(n)} = \frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y}_i}{1-h_i})^2$$

Where h_i is the leverage statistic for the i-th sample, defined as

$$h_i = \frac{1}{N} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{N}(x_j - \bar{x})^2}$$

$\bar{x}$ is the mean of all the sample features x_i.

Prove that for univariate ordinary least square linear regression model we have the following:

$$CV_{(n)} = \frac{1}{N}\sum_{i=1}^{N} MSE_i = \frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y}_i}{1-h_i})^2$$

Hint: With the following notations:

X is an N by 2 matrix: $X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} = \begin{bmatrix} \vec{x_0} & \vec{x} \end{bmatrix}$

$\vec{x_i}$ is the i-th row of the matrix X: $\vec{x_i} = \begin{bmatrix} 1 & x_i \end{bmatrix}$

$\vec{x_0}$ is a vector of 1s with length N: $\vec{x_0} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

$\vec{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$ is the coefficient vector for the least square model,

You can try proving $\vec{x_i}(X^TX)^{-1}\vec{x_i}^T = h_i$. Try your best to connect your understanding and interpretations of the closed form solution for ordinary least square linear regression $\vec{\beta} = (X^TX)^{-1}X^T\vec{y}$ with the formula $\vec{x_i}(X^TX)^{-1}\vec{x_i}^T = h_i$, then you should be able to write your proof. (6pts)

① $X^TX = \begin{bmatrix} N & N\bar{x} \\ N\bar{x} & \sum_{j=1}^{N} x_j^2 \end{bmatrix}$

$$(X^TX)^{-1} = \frac{1}{N\sum_{j=1}^{N} x_j^2 - N^2\bar{x}^2} \begin{bmatrix} \sum_{j=1}^{N} x_j^2 & -N\bar{x} \\ -N\bar{x} & N \end{bmatrix}$$

$$\vec{x_i}(X^TX)^{-1}\vec{x_i}^T = \frac{1}{N\sum_{j=1}^{N} x_j^2 - N^2\bar{x}^2}\left(\sum_{j=1}^{N} x_j^2 - 2N\bar{x}x_i + Nx_i^2\right)$$

$$= \frac{1}{N} + \frac{-2\bar{x}x_i + \bar{x}^2 + x_i^2}{\left(\sum_{j=1}^{N} x_j^2 - N\bar{x}^2\right)}$$

$$= \frac{1}{N} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{N}(x_j - \bar{x})^2} \quad \text{since } \sum_{i=1}^{N} 2x_j\bar{x} = \bar{x}\,2N\bar{x} = 2N\bar{x}^2$$

$$= \frac{1}{N} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{N}(x_j - \bar{x})^2}$$

$$= h_i$$

② $\beta = (X^TX)^{-1}X^T\vec{y}$

$h_i = \vec{x_i}(X^TX)^{-1}\vec{x_i}^T$

⇓

$\vec{X}(X^TX)^{-1}\vec{X}^T = h$, where h is a symmetry matrix, whose diagon is $\{h_1, h_2, h_3 \ldots, h_n\}$. h looks like $\begin{bmatrix} h_1 & \times & \times & \times \\ \times & h_2 & \times & \times \\ \times & \times & h_3 & \times \\ \times & \times & \times & \ddots \\ \times & \times & \times & h_n \end{bmatrix}$

$\therefore \begin{cases} \vec{X}\beta = h\vec{y} \\ \vec{X}\beta = \vec{y} \end{cases} \Rightarrow h\vec{y} = \vec{y}$

$\vec{y}$ is the eigenvector of h, and the corresponding eigenvalue is 1

$\therefore (h - I) = \vec{y}$

$$\therefore \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

$$\frac{1}{N}\sum_{i=1}^{N} MSE_i = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$ proved