Midterm Review

3 Linear Regression

① Replacing least squares error with sum of absolute values as error:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \Longrightarrow \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where $\hat{y}_i = w^T x_i + b$

Advantage: In general, the sum of absolute error doesn't penalize outliers as much as MSE does (due to the square). Thus, this error function is more robust to outliers.

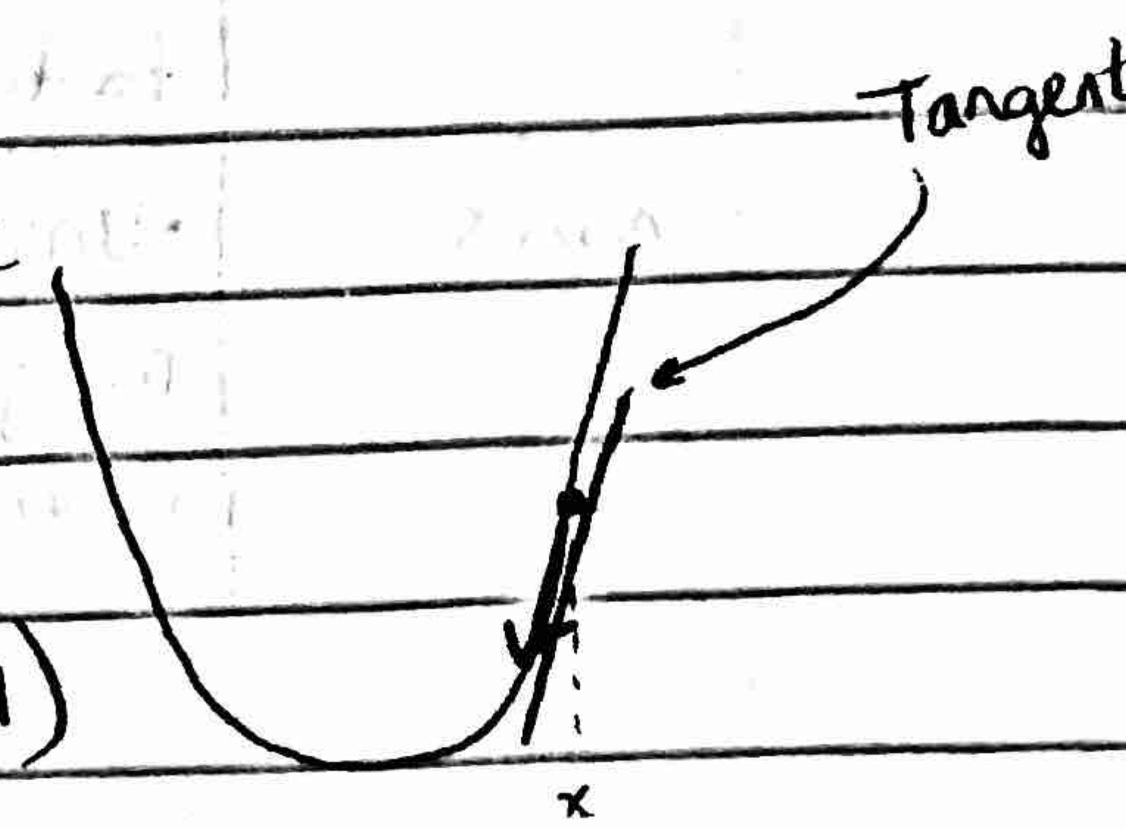
Disadvantage: If data has samples that appear like outliers (but are not), this error function won't penalize a poor fit as much as desired.

② SGD on least squares objective (batch size = 1)

$\hookrightarrow \mathcal{L}(y, \hat{y}) = (y - \hat{y})^2 = (y - w^T x + b)^2$

- Batch size ⇒ how many samples do we train on & collect loss before updating weights?
- Update weights via gradient descent (SGD in this case, where samples are randomized & batch size = 1)

Gradient of $\mathcal{L}(y, \hat{y})$ - partial derivative of loss. Subtract from current weight parameters (we want to go opposite gradient - ball rolling down hill)



Gradients: $\mathcal{L}(y,\hat{y}) = (y-\hat{y})^2 = (y-w^T x+b)^2$

$$\left.\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= 2(y-\hat{y})x \\ \frac{\partial \mathcal{L}}{\partial b} &= 2(y-\hat{y}) \end{aligned}\right\} \text{Chain rule!}$$

Code:

```
for i=1 to n-epochs
    for j = 1 to n              # Shuffled!
        ŷj = w^T xj + b         # make prediction
        diff = yj - ŷj

        w = w - 2ε diff x  }  #update
        b = b - 2ε diff    }
```

where n-epochs = # of times we iterate over training set
n = # of training examples (shuffled for SGD)
ε = learning rate

4 Classification

①	Decision Tree	LDA
Pros	• Interpretable, fast to test points	• Simple computations, closed form
cons	• Unstable to new data, may require restructuring (complex)	• Linear decision boundary (not suited for all problems)

*I replaced the a in the question with x

(2) Derivative of $\sigma(x) = \frac{1}{1+e^{-x}} = (1+e^{-x})^{-1}$

$$\sigma'(x) = -1 \cdot (1+e^{-x})^{-2} \cdot -e^{-x}$$ Chain rule

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$ Consolidate

$$= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}}$$ Separate terms

$$= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}}$$ Add 1/subtract 1 to numerator of second term

$$= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right)$$ Factor out

$$= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right)$$ Simplify

$$= \sigma(x) \cdot (1-\sigma(x))$$

(3) Fisher Linear Discriminant weights ($\hat{w}_{Fisher}$) for a 2 class problem

We know from class that $\hat{w}_{Fisher}$ can be computed by the following –

$$\hat{w}_{Fisher} = \underbrace{S_W^{-1}}_{\text{within class variance}} \underbrace{(\mu_2 - \mu_1)}_{\text{difference of means}}$$

Separate covariance (within-class variance) matrix for each class, then add together

$$\text{where} \quad S_W = \sum_{c} \overbrace{\sum_{i \in C} (x_i - \mu_c)(x_i - \mu_c)^T}$$

To compute: μ_1, μ_2, Σ_1, Σ_2, $\Sigma_W (S_W)$

$\boxed{\mu_1}$ 1x2

$$\mu_1 = \begin{bmatrix} \frac{0 + (-1) + 1}{3} \\ \frac{0 + 0 + 1}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}$$

$\boxed{\mu_2}$ 1x2

$$\mu_2 = \begin{bmatrix} \frac{10 + 11}{2} \\ \frac{10 + 10}{2} \end{bmatrix} = \begin{bmatrix} 21/2 \\ 10 \end{bmatrix}$$

$\boxed{\Sigma_1}$ 2x2

$$\Sigma_1 = \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)\left(\begin{bmatrix} -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)^T$$

$$+ \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1/3 \end{bmatrix}\right)^T$$

$$= \begin{bmatrix} 0 \\ -1/3 \end{bmatrix}\begin{bmatrix} 0 & -1/3 \end{bmatrix} + \begin{bmatrix} -1 \\ -1/3 \end{bmatrix}\begin{bmatrix} -1 & -1/3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2/3 \end{bmatrix}\begin{bmatrix} 1 & 2/3 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & 1/9 \end{bmatrix} + \begin{bmatrix} 1 & 1/3 \\ 1/3 & 1/9 \end{bmatrix} + \begin{bmatrix} 1 & 2/3 \\ 2/3 & 4/9 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 5/9 \end{bmatrix} = \Sigma_1$$

$\boxed{\Sigma_2}$ 2x2

$$\Sigma_2 = \left(\begin{bmatrix}10\\10\end{bmatrix} - \begin{bmatrix}21/2\\10\end{bmatrix}\right)\left(\begin{bmatrix}10\\10\end{bmatrix} - \begin{bmatrix}21/2\\10\end{bmatrix}\right)^T + \left(\begin{bmatrix}11\\10\end{bmatrix} - \begin{bmatrix}21/2\\10\end{bmatrix}\right)\left(\begin{bmatrix}11\\10\end{bmatrix} - \begin{bmatrix}21/2\\10\end{bmatrix}\right)^T$$

$$= \begin{bmatrix}-1/2\\0\end{bmatrix}\begin{bmatrix}-1/2 & 0\end{bmatrix} + \begin{bmatrix}1/2\\0\end{bmatrix}\begin{bmatrix}1/2 & 0\end{bmatrix}$$

$$= \begin{bmatrix}1/4 & 0\\0 & 0\end{bmatrix} + \begin{bmatrix}1/4 & 0\\0 & 0\end{bmatrix} = \begin{bmatrix}1/2 & 0\\0 & 0\end{bmatrix} = \Sigma_2$$

$\boxed{\Sigma_{SW}}$ 2x2 $= \Sigma_1 + \Sigma_2$

$$= \begin{bmatrix}2 & 1\\1 & 5/9\end{bmatrix} + \begin{bmatrix}1/2 & 0\\0 & 0\end{bmatrix} = \begin{bmatrix}2\,1/2 & 1\\1 & 5/9\end{bmatrix}$$

$$\hat{w}_{Fisher} = S_W^{-1}(\mu_2 - \mu_1)$$

Inverse of 2x2 $\Rightarrow \begin{bmatrix}a & b\\c & d\end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix}d & -b\\-c & a\end{bmatrix}$

$$\begin{bmatrix}2\,1/2 & 1\\1 & 5/9\end{bmatrix}^{-1} = \frac{1}{\frac{5}{2}\cdot\frac{5}{9} - 1}\begin{bmatrix}5/9 & -1\\-1 & 5/2\end{bmatrix} = \frac{18}{7}\begin{bmatrix}5/9 & -1\\-1 & 5/2\end{bmatrix}$$

(Not simplified)

$$\hat{w}_{Fisher} = \frac{18}{7}\begin{bmatrix}5/9 & -1\\-1 & 5/2\end{bmatrix}\left(\begin{bmatrix}\frac{21}{2}\\10\end{bmatrix} - \begin{bmatrix}0\\1/3\end{bmatrix}\right)$$

5 General

When would you use a validation set in addition to training & testing sets?

A validation set is a portion of your data not used for training or testing sets, used to tune hyperparameters

In Fisher's LDA, the decision rule is

$$h_{Fisher}(\vec{x}) = I\{\hat{w}_{Fisher}{}^T \vec{x} > \tau\}$$

where τ acts as the bias (more literally, τ = "negative bias", since $\hat{w}_{Fisher}{}^T x + b = 0$

$\hat{w}_{Fisher}{}^T x = -b$, so $-b = \tau$).

The threshold τ, the point where we determine whether a test point will be classified as class 1 or 2, is determined by cross-validation using a validation set.