5CS037-Concepts and Technologies of AI.

Tutorial-09: Bias-Variance Tradeoff in Machine Learning.

Things to remember from Lecture.

Recap: Gradient Descent for Logistic Regression.

- **Outline**:
  - Have cost function $\boldsymbol{J(\theta)}$, where $\boldsymbol{\theta} = [\boldsymbol{\theta_0}, \dots, \boldsymbol{\theta_m}]$
  - Start off with some guesses for $\theta_0, \dots, \theta_m$
    - It does not really matter what values you start off with, but a common choice is to set them all initially to zero
  - Repeat until convergence{

$$\theta_j = \theta_j - \alpha \sum_{i=1}^{n} \left(\frac{1}{1 + e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)} \right) x_j{}^{(i)}$$

}

Logistic Regression: A Concrete Example.

- The Training Phase:
- Dataset: **X=[x_0,x_1,x_2,x_3,x_4,x_5]➔Feature Vector.**

/	X_0=1	X_1	X_2	X_3	X_4	X_5	Y
a	1	1	1	0	1	1	1
b	1	0	0	1	1	0	0
c	1	0	1	1	0	0	1
d	1	1	0	0	1	0	0
e	1	1	0	1	0	1	1
f	1	1	0	1	1	0	0

To account for the intercept

The Training Phase:

- Initialize: $\alpha = 0.5$ and $\theta = [0,0,0,0,0,0]$

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_0$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

The Training Phase:

- Initialize: $\alpha = 0.5$ and $\theta = [0,0,0,0,0,0]$

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_0$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	$(\frac{1}{1+e^{-0}} - 1) \times 1 = -0.5$
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	$(\frac{1}{1+1} - 1) \times 1 = -0.5$
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	$(\frac{1}{1+1} - 1) \times 1 = -0.5$
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	$(\frac{1}{1+1} - 0) \times 1 = 0.5$

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}} - y^{(i)}\right)x_0{}^{(i)} = \mathbf{0}$$

Then,

$\theta_0 = \theta_0 - \alpha \times \mathbf{0}$

$= 0 - 0.5 \times \mathbf{0} = \mathbf{0}$

New θ_0

New Paramter Vector:

$\boldsymbol{\theta = [0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5]}$

1.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_1$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

1.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_1$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0.5
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) {x_1}^{(i)} = \mathbf{0}$$

Then,

$$\theta_1 = \theta_1 - \alpha \times \mathbf{0}$$
$$= 0 - 0.5 \times \mathbf{0} = \mathbf{0}$$

New Paramter Vector:
$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \boldsymbol{\theta_2}, \boldsymbol{\theta_3}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$

2.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_2$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

2.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_2$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	-0.5
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	0
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\boldsymbol{\theta}^T x^{(i)}}} - y^{(i)}\right) {x_2}^{(i)} = -\mathbf{1}$$

Then,

$$\theta_2 = \theta_2 - \alpha \times (-\mathbf{1})$$
$$= 0 - 0.5 \times (-\mathbf{1}) = \mathbf{0.5}$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \boldsymbol{\theta_3}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

3.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_3$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

3.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_3$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	0
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0.5
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	-0.5
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}} - y^{(i)}\right)x_3{}^{(i)} = \mathbf{0}$$

Then,

$$\theta_3 = \theta_3 - \alpha \times \mathbf{0}$$
$$= 0 - 0.5 \times 0 = \mathbf{0}$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \boldsymbol{\theta_4}, \boldsymbol{\theta_5}]$$

4.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_4$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

4.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_4$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0.5
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0.5
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	0
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0.5

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}} - y^{(i)}\right)x_4{}^{(i)} = \mathbf{1}$$

Then,

$$\theta_4 = \theta_4 - \alpha \times \mathbf{1}$$
$$= 0 - 0.5 \times 1 = \mathbf{-0.5}$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \boldsymbol{\theta_5}]$$

5.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_5$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	

5.

x	y	$\theta^T x$	$(\frac{1}{1+e^{-\theta^T x}} - y)x_5$
[1,1,1,0,1,1]	**1**	[0,0,0,0,0,0]×[1,1,1,0,1,1]=0	-0.5
[1,0,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,0,0,1,1,0]=0	0
[1,0,1,1,0,0]	**1**	[0,0,0,0,0,0]×[1,0,1,1,0,0]=0	0
[1,1,0,0,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,0,1,0]=0	0
[1,1,0,1,0,1]	**1**	[0,0,0,0,0,0]×[1,1,0,1,0,1]=0	-0.5
[1,1,0,1,1,0]	**0**	[0,0,0,0,0,0]×[1,1,0,1,1,0]=0	0

$$\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x^{(i)}}} - y^{(i)}\right)x_5{}^{(i)} = -\mathbf{1}$$

$$\boldsymbol{Then},$$

$$\theta_5 = \theta_5 - \alpha \times (-\mathbf{1})$$
$$= 0 - 0.5 \times (-1) = \mathbf{0.5}$$

New Paramter Vector:

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$$

The Testing Phase:

- Testing is typically done over a portion of the dataset that is not used during training, but rather kept only for testing the accuracy of the algorithm's predictions thus far
- $\theta = [0,0,0.5,0,-0.5,0.5]$→ Learned Parameters

$(if\ h_\theta(x) \geq 0.5, y' = 1; else\ y' = 0)$

x	y	$\theta^T x$	$h_\theta(x) = (\frac{1}{1+e^{-\theta^T x}})$	Predicted Class
[1,1,1,0,1,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,1,0,1,1]=0.5	0.622459331	
[1,0,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,0,0,1,1,0]=-0.5	0.377540669	
[1,0,1,1,0,0]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,0,1,1,0,0]=0.5	0.622459331	
[1,1,0,0,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,0,1,0]=-0.5	0.377540669	
[1,1,0,1,0,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,0,1]=0.5	0.622459331	
[1,1,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,1,0]=-0.5	0.377540669	

The Testing Phase:

$(if\ h_\theta(x) \geq 0.5, y' = 1; else\ y' = 0)$

x	y	$\theta^T x$	$h_\theta(x) = (\frac{1}{1+e^{-\theta^T x}})$	Predicted Class (or y')
[1,1,1,0,1,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,1,0,1,1]=0.5	0.622459331	**1**
[1,0,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,0,0,1,1,0]=-0.5	0.377540669	**0**
[1,0,1,1,0,0]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,0,1,1,0,0]=0.5	0.622459331	**1**
[1,1,0,0,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,0,1,0]=-0.5	0.377540669	**0**
[1,1,0,1,0,1]	**1**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,0,1]=0.5	0.622459331	**1**
[1,1,0,1,1,0]	**0**	[0,0,0.5,0,-0.5,0.5]×[1,1,0,1,1,0]=-0.5	0.377540669	**0**

The Inference:

- Let us infer whether a given new data, say, $\boldsymbol{g} = [1, 0, 1, 0, 0, 1] \in \{0 \; or \; 1\}$, using logistic regression with the just learnt parameter vector

$$\boldsymbol{\theta} = [\mathbf{0}, \mathbf{0}, \mathbf{0.5}, \mathbf{0}, \mathbf{-0.5}, \mathbf{0.5}]$$

/	X_1=1	X_2	X_3	X_4	X_5	X_6	Y
a	1	1	1	0	1	1	1
b	1	0	0	1	1	0	0
c	1	0	1	1	0	0	1
d	1	1	0	0	1	0	0
e	1	1	0	1	0	1	1
f	1	1	0	1	1	0	0
g	1	0	1	0	0	1	?

The Inference:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \\ -0.5 \\ 0.5 \end{bmatrix} [1, 0, 1, 0, 0, 1] = (0.5 \times 1) + (0.5 \times 1) = 1$$

$$= \frac{1}{1 + e^{-1}}$$

$$= 0.731$$

$$\geq 0.5 \quad \rightarrow \text{Class 1 .}$$

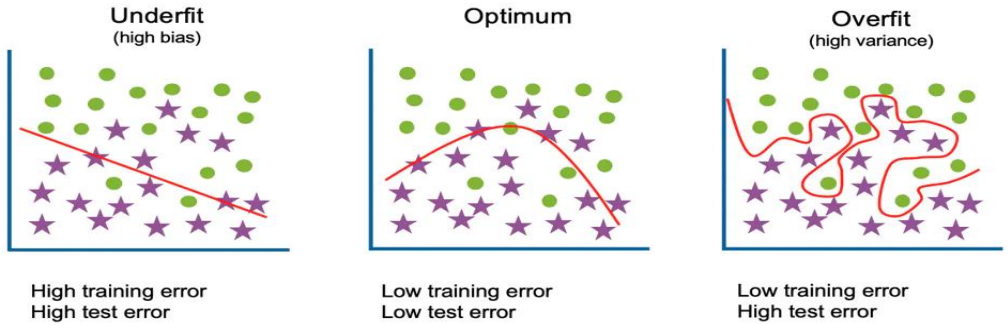
Overfitting and Underfitting.

Overfitting and Underfitting

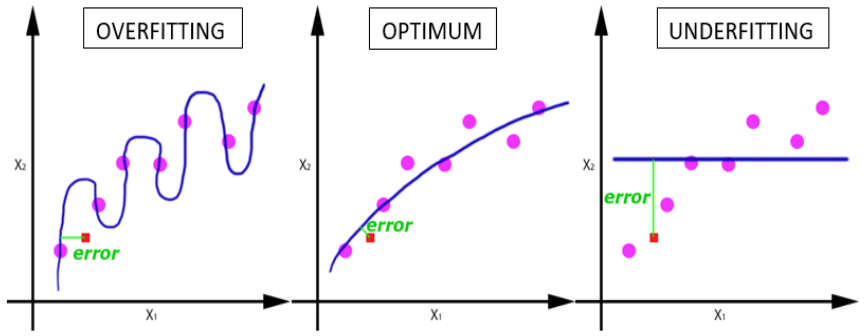
- **Overfitting:**
  - We train our machine learning model with finite set of sample dataset;
  - In the process of doing that, if we trained our model for too long i.e. many number of iteration or the model is too complex then it will start to learn the noise and other irrelevant information with in the dataset, then the model memorizes the noise and fits too closely to the training dataset, models becomes *"overfitted"* and it is unable to *generalize well to unseen data*. Then it will not be able to perform classification or regression task.
  - ***Very low training error but very high test error***
  - The overfitted model experience Low Bias and High variance.
- **Underfitting:**
  - It occurs when models are not train for enough time or the independent or feature variables are not significant enough to determine the meaningful relationship between the input and output variables. It also generalizes poorly.
  - ***Both Training and Test Error are very high***
  - The *"underfitted"* model experiences High Bias and Low Variance

Overfitting and Underfitting: A graphical Analogy

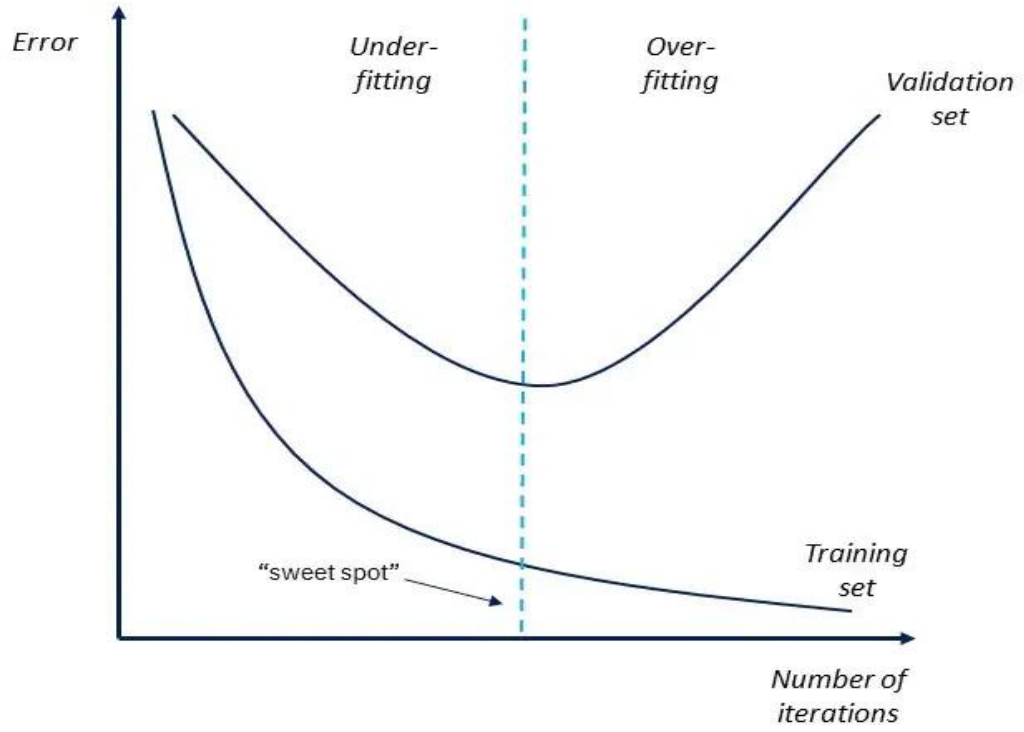
- For Classifier:



- For Regressor:



Ideally



Generalization Error

- Generalization Error or some times also known as *out of sample error*: measures the ability of the machine learning models to predict outcome values for ***previously unseen data.***
- We can estimate the Generalization error using Test Set
- Components of Error:
  - Noise
  - Bias
    - Occurs due to “expressive handicap”
  - Variance
    - Occurs due to finite sample of training set
- Noise can not be reduced, what about Bias and Variance?

Bias and Variance.

Bias

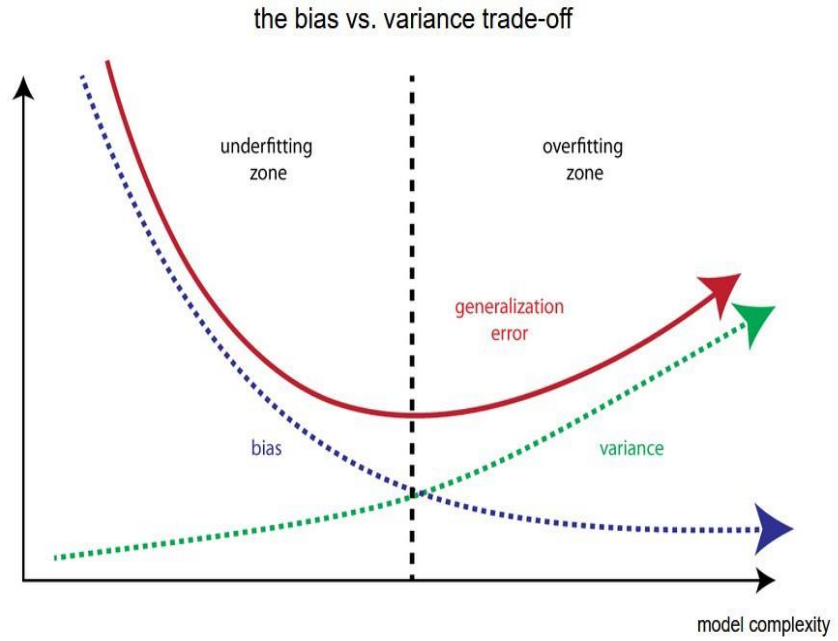
- It is the inability of the machine learning models to capture the true relationship of input and output variables.
- It occurs in the machine learning on its own due to incorrect assumptions.
- It describes how well the model matches the training data:
  - Higher Bias would not match the dataset
  - Low bias would closely match the dataset
- Characteristics of **High Bias Model**:
  - Does not capture true trend in dataset
  - Underfitts
  - Overly simplified model
  - High error rate

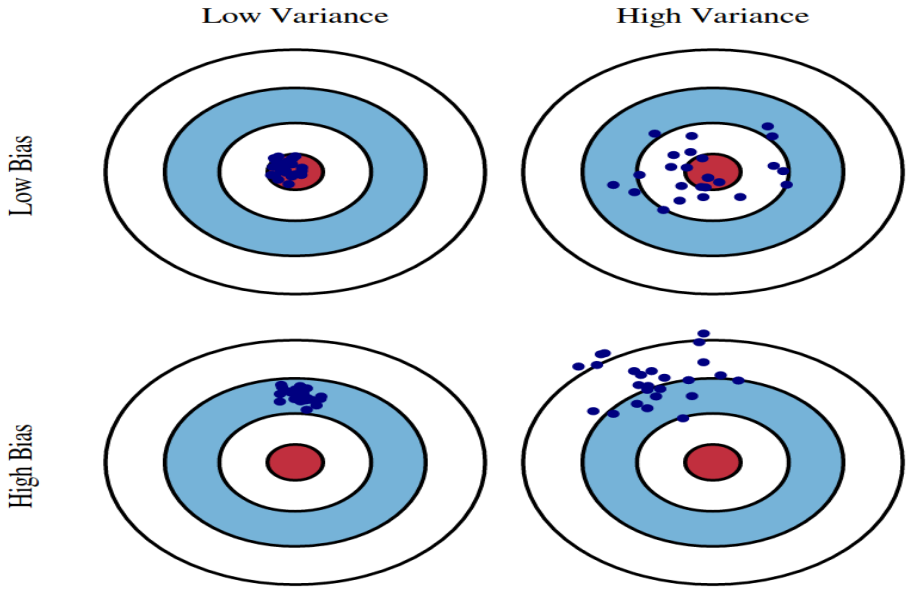
Variance

- Difference in the fits of dataset during training and test is variance
- It describes the variability in the model i.e. how it can adjust to unseen datasets
- Variance increases with the complexity of the models
- Characteristics of a **High Variance Model**:
  - Noise in the dataset
  - Overfitting
  - Complex models
  - Trying to fit all the datapoints including irrelevant information

Bias-Variance Trade off

- Models with high bias will have low variance.
- Models with high variance will have a low bias.





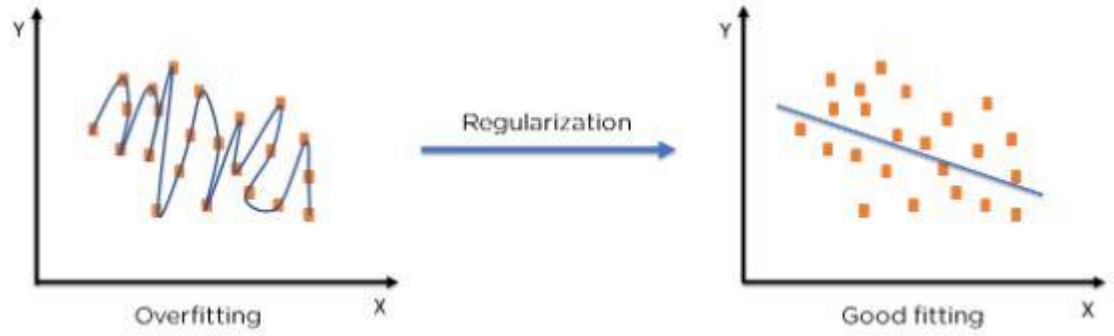
How to handle overfitting?

- Some Techniques to avoid overfitting:
  - Early stopping
  - Train with more data
  - Feature Selection or Data augmentation
  - Cross-Validation
  - Ensemble Methods
  - Regularization

(Note: We will talk about Ensemble methods and Cross-validation in week 3)

Regularization

- Regularization refers to **techniques that are used to calibrate machine learning models** in order to minimize the adjusted loss function and prevent overfitting or Underfitting.
- Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

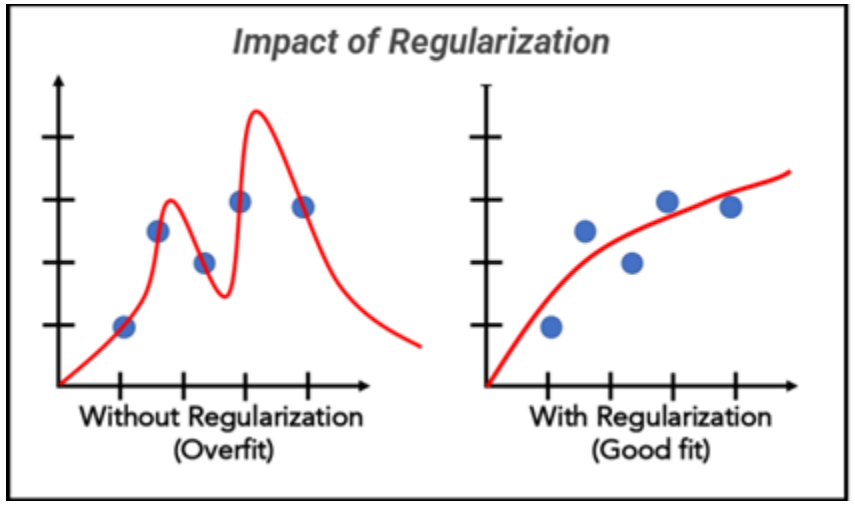


Ridge Regression (L2 Regularization)

- Regularized form of linear regression
- Works for overfitted model by introducing some degree of bias into the data
- Which is introduced in the form of a regularization parameter; which penalizes the size of the error. **(Loss function + Regularized Term)**
  - $J(\theta) = \sum(y_i - \hat{y}_i)^2 + \lambda * \theta^2$
- Characteristics of λ:
  - $\lambda = 0$;
    - No impact, model would still overfitt.
  - $\lambda \Rightarrow Minimal$;
    - Generalized model and may have acceptable accuracy.
  - $\lambda \Rightarrow High$;
    - Very high impact may lead to Underfitting.
- How to pick appropriate λ then?
  - Cross-Validation and Hyper-parameter tuning with Validation set.

Lasso Regression (L1 Regularization)

- Similar to Ridge Regression i.e. **(Loss function + Regularized Term)**
- Regularized Term becomes:
  - $J(\theta) = \sum(y_i - \hat{y}_i)^2 + \lambda * |\theta|$



Discussion

- Overfitting relates to have *High Variance*, to fight overfitting we need to focus on reducing the variance by *increasing the regularization*, obtain larger dataset, decreasing the number of features.
- Underfitting relates to have *High Bias*, to fight we focus on reducing the bias by *decreasing the regularization*, using larger model and larger features.
- To improve the *Generalization error*, understand which components have highest impact on the dataset, then go after that.
- Always analyze the model by looking into ***training and validation error*** simultaneously.

Summary.