EE2211 Pre-Tutorial 7

Dr Feng LIN

feng_lin@nus.edu.sg

Agenda

- Recap
- Self-learning
- Tutorial 7

Today’s Attendance

Recap

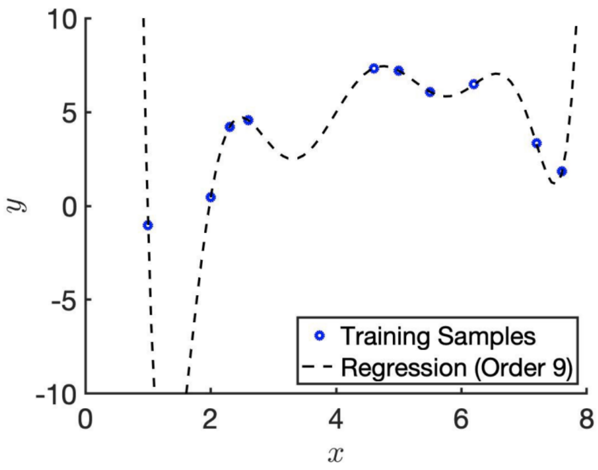
- Overfitting, underfitting & model complexity
  - Overfitting: low error in training set, high error in test set
  - Underfitting: high error in both training & test sets
  - Overly complex models can overfit; Overly simple models can underfit
- Feature selection
  - Extract useful features from training set
- Regularization (e.g., L2 regularization)
  - Solve "ill-posed" problem (e.g., more unknowns than data points)
  - Reduce overfitting
- Bias-Variance Decomposition Theorem
  - Test error = Bias Squared + Variance + Irreducible Noise
  - Can be interpreted as trading off bias & variance:
    - Overly complex models can have high variance, low bias
    - Overly simple models can have low variance, high bias

Overfitting

Training

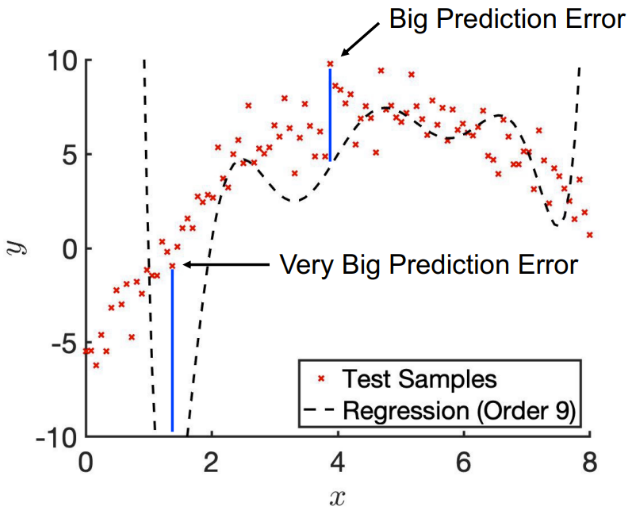
Testing

Overfitting Example



	Training Set Fit	
Order 9	Good	

Overfitting Example



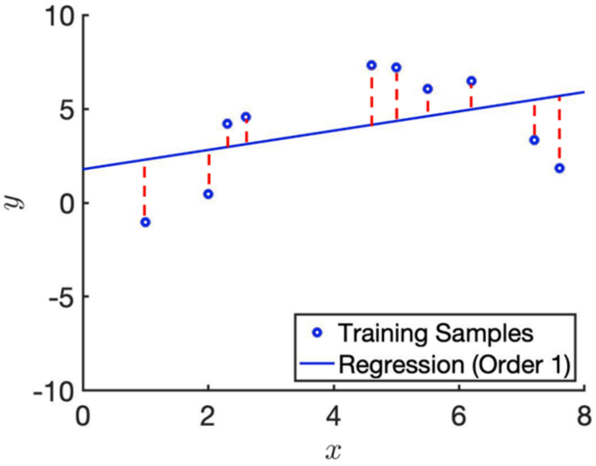
	Training Set Fit	Test Set Fit
Order 9	Good	Bad

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i\, x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij}\, x_i x_j + \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d} w_{ijk}\, x_i x_j x_k + \cdots$$

Underfitting

Training

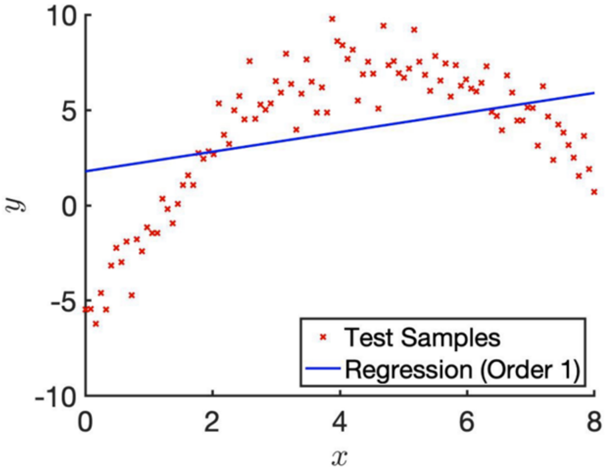
Underfitting Example



	Training Set Fit	Test Set Fit
Order 9	Good	Bad
Order 1	Bad	

Testing

Underfitting Example

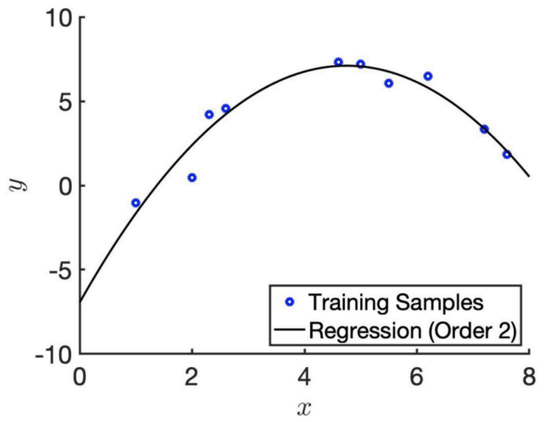


	Training Set Fit	Test Set Fit
Order 9	Good	Bad
Order 1	Bad	Bad

Perfect Fitting

Training

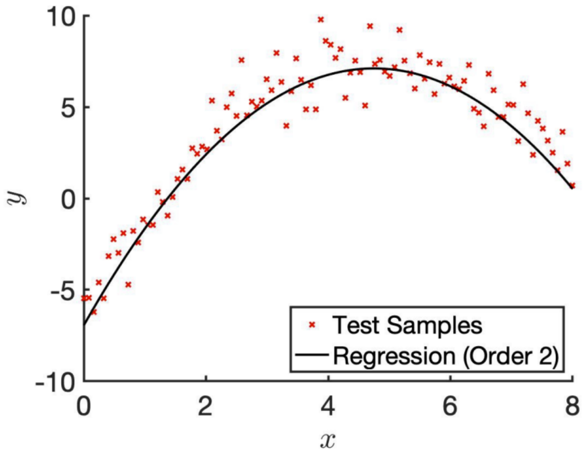
“Just Nice”



	Training Set Fit	Test Set Fit
Order 9	Good	Bad
Order 1	Bad	Bad
Order 2	Good	

Testing

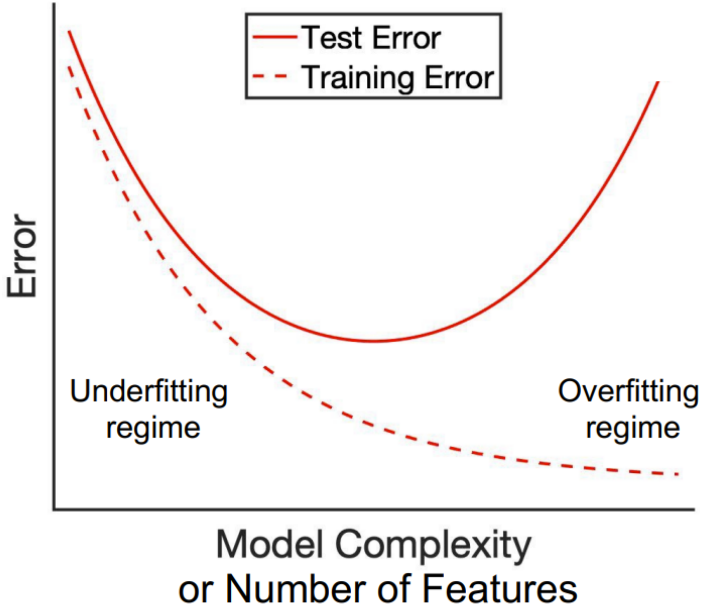
“Just Nice”



	Training Set Fit	Test Set Fit
Order 9	Good	Bad
Order 1	Bad	Bad
Order 2	Good	Good

Fitting VS Model Complexity

Overfitting / Underfitting Schematic



Linear model

Complex model

Overfitting

- **Overfitting** occurs when model predicts the training data well, but predicts new data (e.g., from test set) poorly
- **Reason 1**
  - Model is too complex for the data
  - Previous example: Fit order 9 polynomial to 10 data points
- **Reason 2**
  - Too many features but number of training samples too small
  - Even linear model can overfit, e.g., linear model with 9 input features (i.e., w is 10-D) and 10 data points in training set => data might not be enough to estimate 10 unknowns well
- **Solutions**
  - Use **simpler models** (e.g., lower order polynomial)
  - Use **regularization** (see next part of lecture)

Underfitting

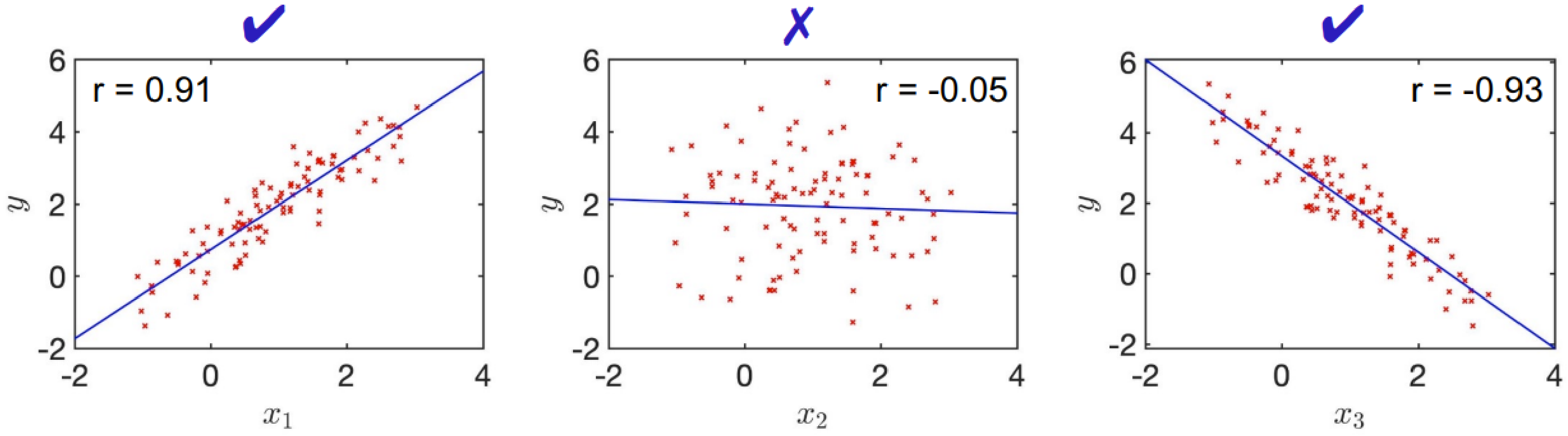
- **Underfitting** is the inability of trained model to predict the targets in the training set
- **Reason 1**
  - Model is too simple for the data
  - Previous example: Fit order 1 polynomial to 10 data points that came from an order 2 polynomial
  - **Solution**: Try more complex model
- **Reason 2**
  - Features are not informative enough
  - **Solution**: Try to develop more informative features

Feature Selection

- Less features might reduce overfitting
  - Want to discard useless features & keep good features, so perform feature selection
- Feature selection procedure
  - Step 1: feature selection in training set
  - Step 2: fit model using selected features in training set
  - Step 3: evaluate trained model using test set
- Very common mistake
  - Feature selection with test set (or full dataset) leads to inflated performance
  - Do not perform feature selection with test data

Pearson's R

– Pearson's correlation r measures linear relationship between two variables



Pearson's R

- Given features x, we want to predict target y
- Assume x & y both continuous
- Compute Pearson's correlation coefficient between each feature & target y in the training set
  - Pearson's correlation r measures linear relationship between two variables
- Two options
  - Option 1: Pick K features with largest absolute correlations
  - Option 2: Pick all features with absolute correlations > C
  - K & C are "magic" numbers set by the ML practitioner
- Other metrics besides Pearson's correlation are possible

Regularization

- Regularization is an umbrella term that includes methods forcing learning algorithm to build less complex models.
- Motivation 1: Solve an ill-posed problem
  - For example, estimate 10th order polynomial with just 5 datapoints
- Motivation 2: Reduce overfitting
- For example, in previous lecture, we added $\lambda \mathbf{w}^T \mathbf{w}$:

$$\underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{P}\mathbf{w} - \mathbf{y})^T (\mathbf{P}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w}$$

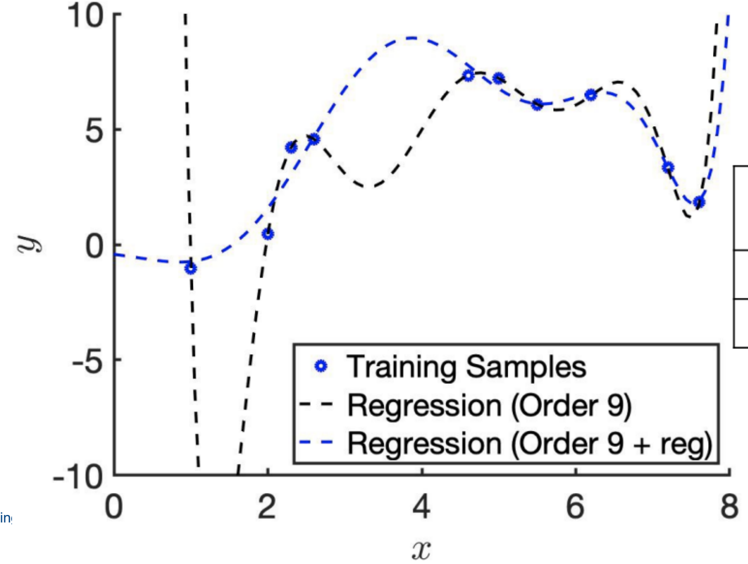
- Minimizing with respect to $\mathbf{w}$, primal solution is

$$\hat{\mathbf{w}} = (\mathbf{P}^T \mathbf{P} + \lambda \mathbf{I})^{-1} \mathbf{P}^T \mathbf{y}$$

- For $\lambda > 0$, matrix becomes invertible (Motivation 1)
- $\hat{\mathbf{w}}$ might also perform better in test set, i.e., reduces overfitting (Motivation 2) – will show example later

Regularization

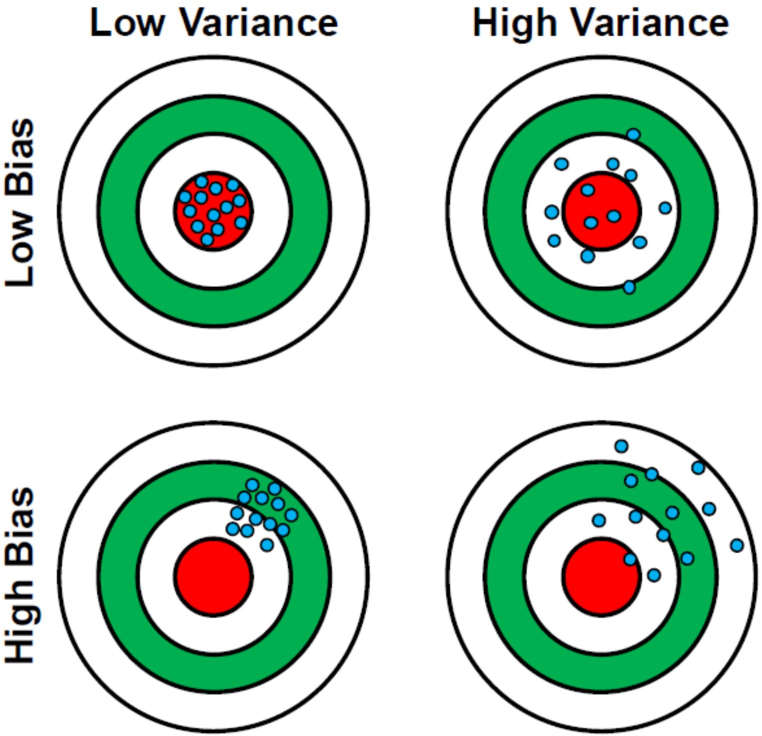
$$\underset{\mathbf{w}}{\operatorname{argmin}} \underbrace{(\mathbf{P}\mathbf{w}-\mathbf{y})^T(\mathbf{P}\mathbf{w}-\mathbf{y})}_{\text{Cost function quantifying data fitting error in training set}} + \underbrace{\lambda \mathbf{w}^T\mathbf{w}}_{\text{Regularization}}$$



	Training Set Fit	Test Set Fit
Order 9	Good	Bad
Order 9, λ = 1	Good	

Bias vs Variance

- Suppose we are trying to predict red target below:



Low Bias: blue predictions on average close to red target
Low Variance: low variability among blue predictions

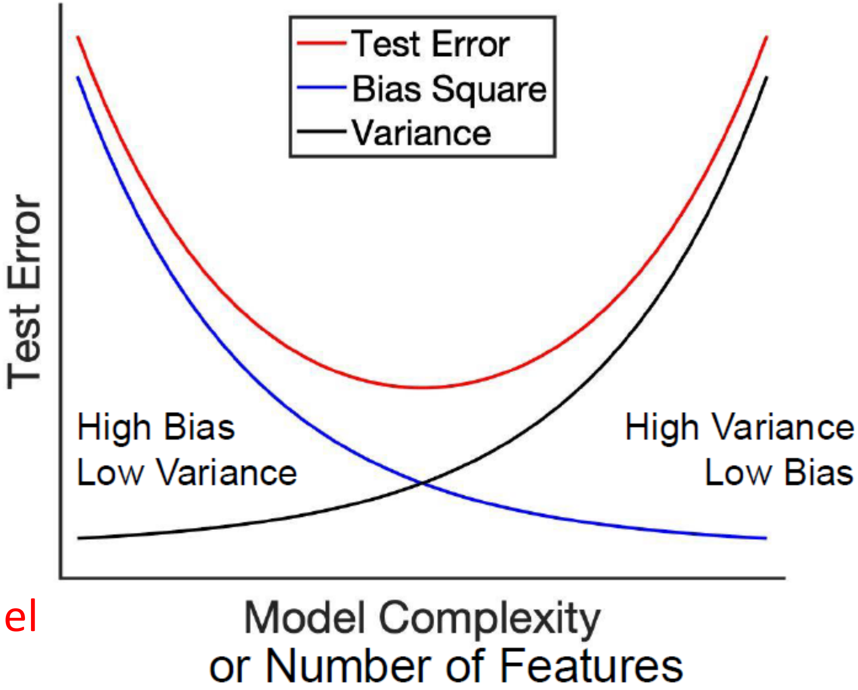
Low Bias: blue predictions on average close to red target
High Variance: large variability among blue predictions

High Bias: blue predictions on average not close to red target
Low Variance: Low variability among blue predictions

High Bias: blue predictions on average not close to red target
High Variance: high variability among blue predictions

Bias + Variance Trade Off

- Test error = Bias Squared + Variance + Irreducible Noise



Linear model

Complex model

Bias + Variance Example

Each time we randomly pick 10 different training samples.

- Simulate data from order 2 polynomial (+ noise)
- Randomly sample 10 training samples each time
- Fit with order 2 polynomial: low variance, low bias
- Fit with order 4 polynomial: high variance, low bias

Order 2 Achieves Lower Test Error

Given a new (or test) sample $x \in R^d$, we can obtain its prediction $\hat{f}(x)$ as follows:

$$\hat{f}_D(x) = x^T \hat{w}$$

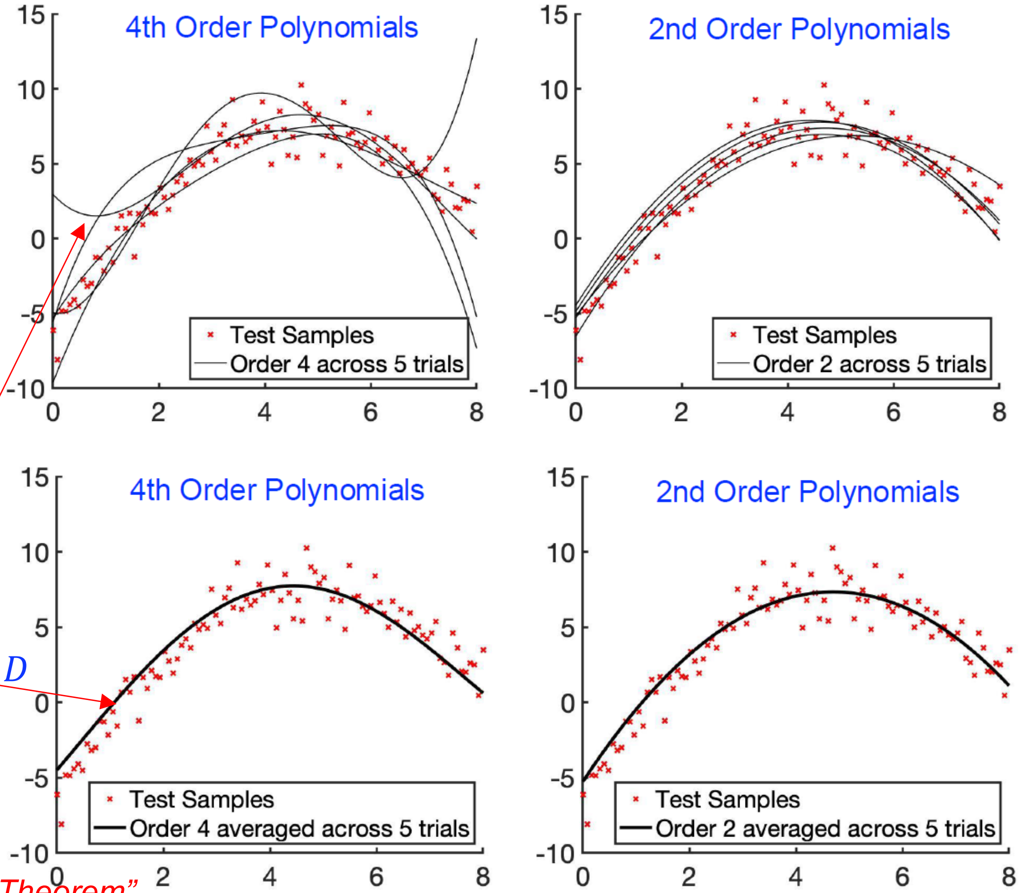
Evaluate the bias and variance of $\hat{f}(x)$ below:

$$\boxed{Variance\left(\hat{f}(s)\right) = E\left[\left(\hat{f}_D(s) - E[\hat{f}_D(s)]\right)^2\right]}$$

$$Bias\left(\hat{f}(x)\right) = \boxed{E[\hat{f}_D(x)]} - f(x)$$

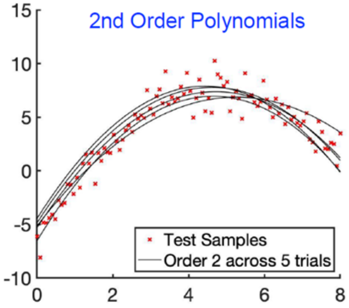
'average'

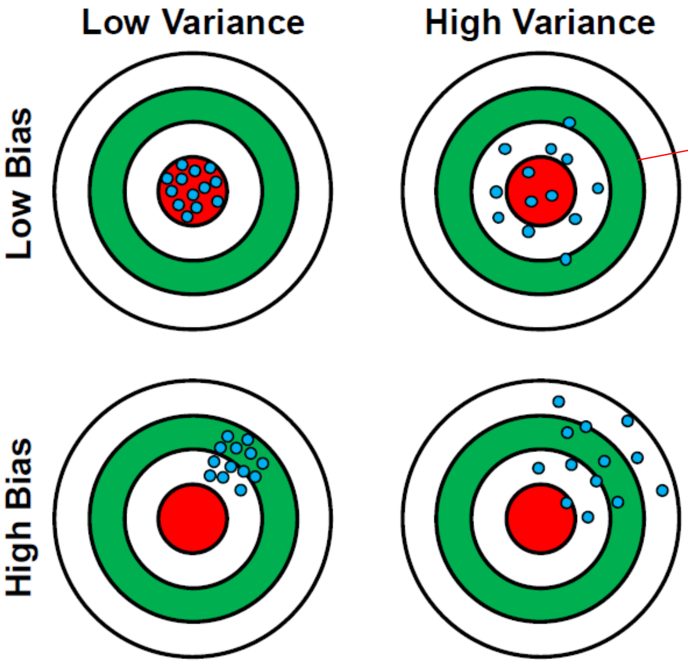
- $\hat{f}_D(x)$ is the model's prediction for a specific dataset D
- $E[\hat{f}_D(s)]$ is the expected prediction of the model (averaged over all possible datasets)
- $f(s)$ is the true underlying functions

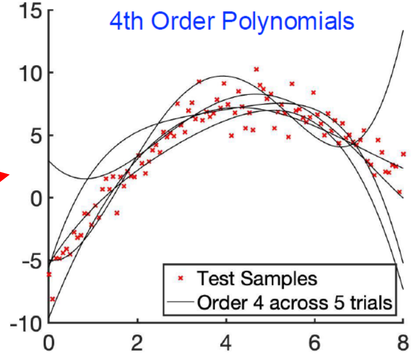


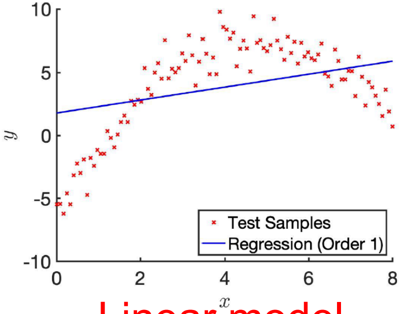
Refer to "Optional Material on Bias-Variance Decomposition Theorem"

Bias + Variance Example









Linear model

$$E_D\left[\left(y(x)-\hat{f}_D(x)\right)^2\right] = \mathrm{Bias}(\hat{f})^2 + \mathrm{Var}(\hat{f}) + \sigma^2,$$

Bias-Variance Decomposition Theorem

- **Test error** = Bias Squared + Variance + Irreducible Noise
  - Mathematical details in optional uploaded material (won't be tested)
- "**Variance**" refers to variability of prediction models across different training sets
  - In previous example, every time the training set of 10 samples changes, the trained model changes
  - "Variance" quantifies variability across trained models
- "**Bias**" refers to how well an average prediction model will perform
  - In previous example, every time the training set of 10 samples changes, the trained model changes
  - If we average the trained models, how well will this average trained model perform?
- "**Irreducible Noise**" reflects the fact that even if we are perfect modelers, it might not be possible to predict target y with 100% accuracy from feature(s) x

Refer to "Optional Material on Bias-Variance Decomposition Theorem"

THANK YOU