DDA2020 Machine Learning: L11 Over/Under-Fitting, Bias-Variance Trade-off

Haizhou Li
School of Data Science, CUHK-SZ

March 30, 2023

Outline

1. Overfitting, underfitting and model complexity

2. Bias-variance trade-off
   - Experimental observations
   - Statistical analysis

1. Overfitting, underfitting and model complexity

2. Bias-variance trade-off
   - Experimental observations
   - Statistical analysis

Overfitting and Underfitting

Motivation: Learning Goal is Prediction

$$
\begin{aligned}
\text{Learning:} \quad & \widehat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} && \text{(Linear)} \\
\text{or} \quad & \widehat{\mathbf{w}} = \left(\mathbf{P}^T\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{y} && \text{(Polynomial)} \\
\text{Prediction:} \quad & f_{\mathbf{w},b}\left(\mathbf{X}_{new}\right) = \mathbf{X}_{new}\widehat{\mathbf{w}} && \text{(Linear)} \\
\text{or} \quad & f_{\mathbf{w},b}\left(\mathbf{X}_{new}\right) = \mathbf{P}_{new}\widehat{\mathbf{w}} && \text{(Polynomial)}
\end{aligned}
$$

Overfitting and Underfitting

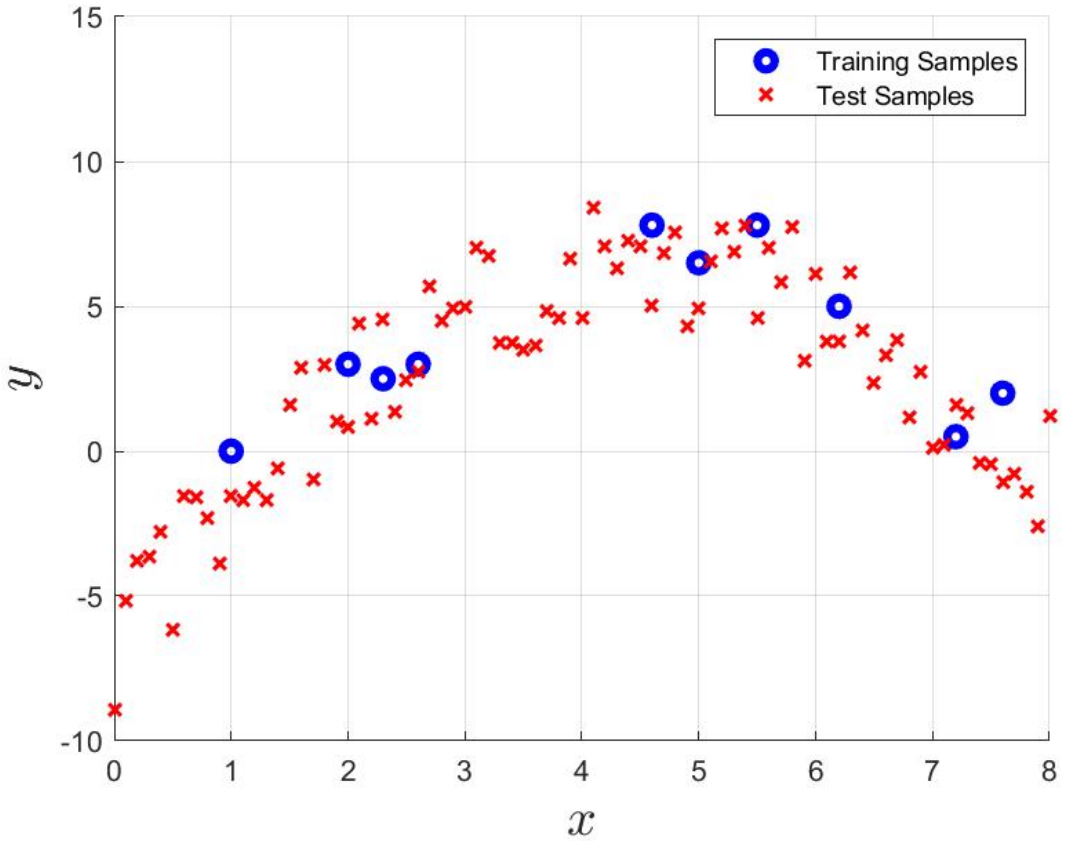
Underfitting

- Underfitting is the **inability** of the model to predict well the labels of the data it was trained on. There could be several reasons for underfitting, the most important of which are:
  - your model is too simple for the data (for example a linear model can often underfit)
  - the features you engineered are not informative enough
- The solution to the problem of underfitting is to try a **more complex model** or to **engineer features with higher predictive power**

Overfitting and Underfitting

Underfitting: Example

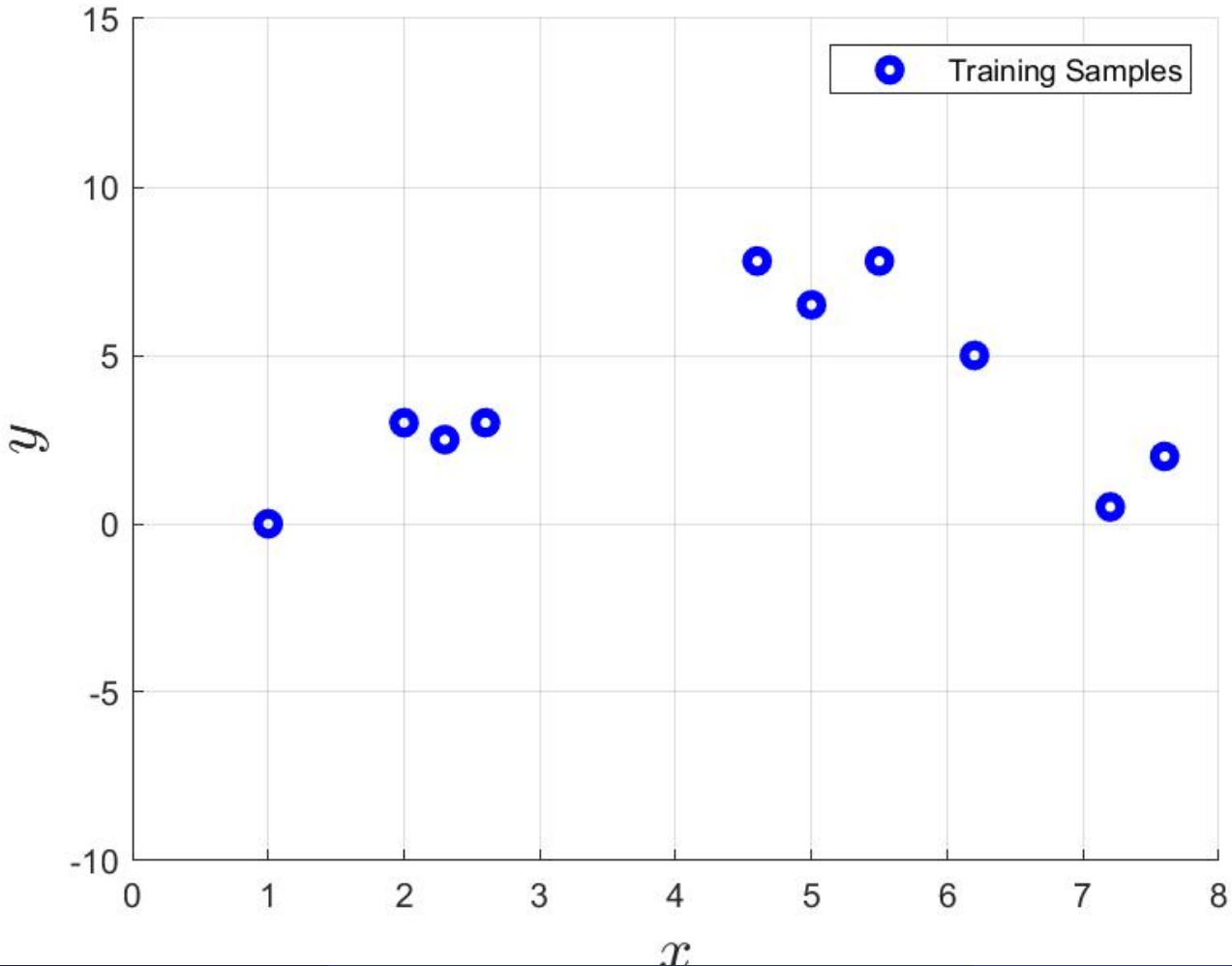
- Suppose these are the entire data samples for a particular experiment.
- It is common that only a part of the entire data is available for training. For example, out of all these points, only the blue circles are available for training.



Overfitting and Underfitting

Underfitting: Example

Based on these training samples, suppose we start with a linear model for learning.



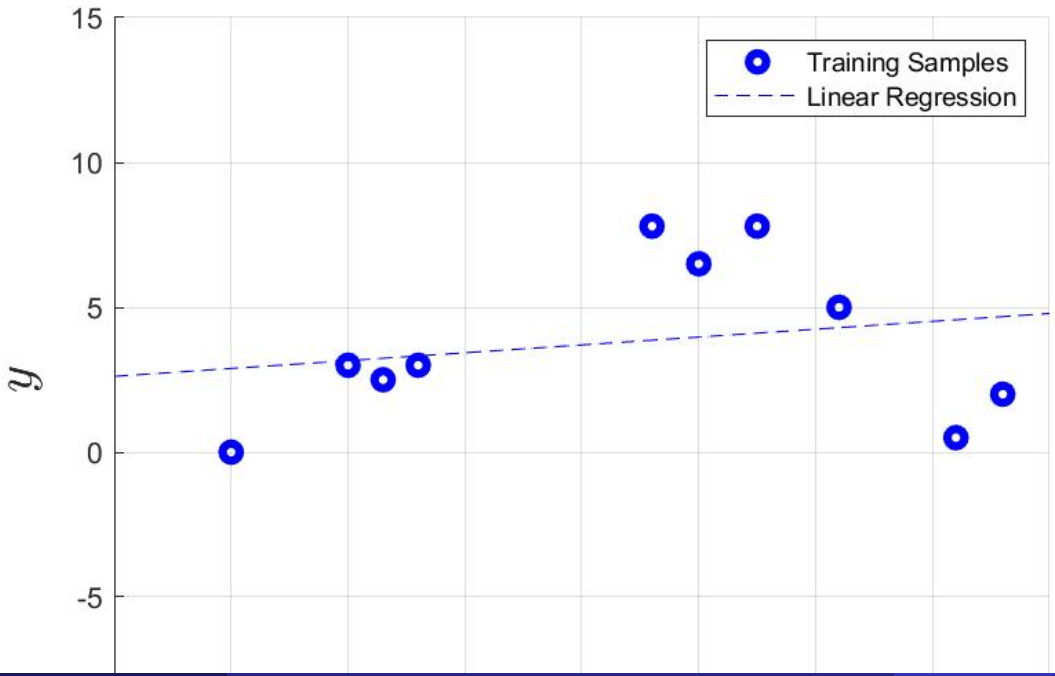
Overfitting and Underfitting

Underfitting: Example

Learning outcome: case 1 (linear regression)

This example illustrates underfitting using the linear model.

- The red lines are the error distance between the fitted line and the data point.
- The least squares error has the sum of these distances.
- Based on this linear regression fitting, it does not seem to describe the data well enough.



Overfitting and Underfitting

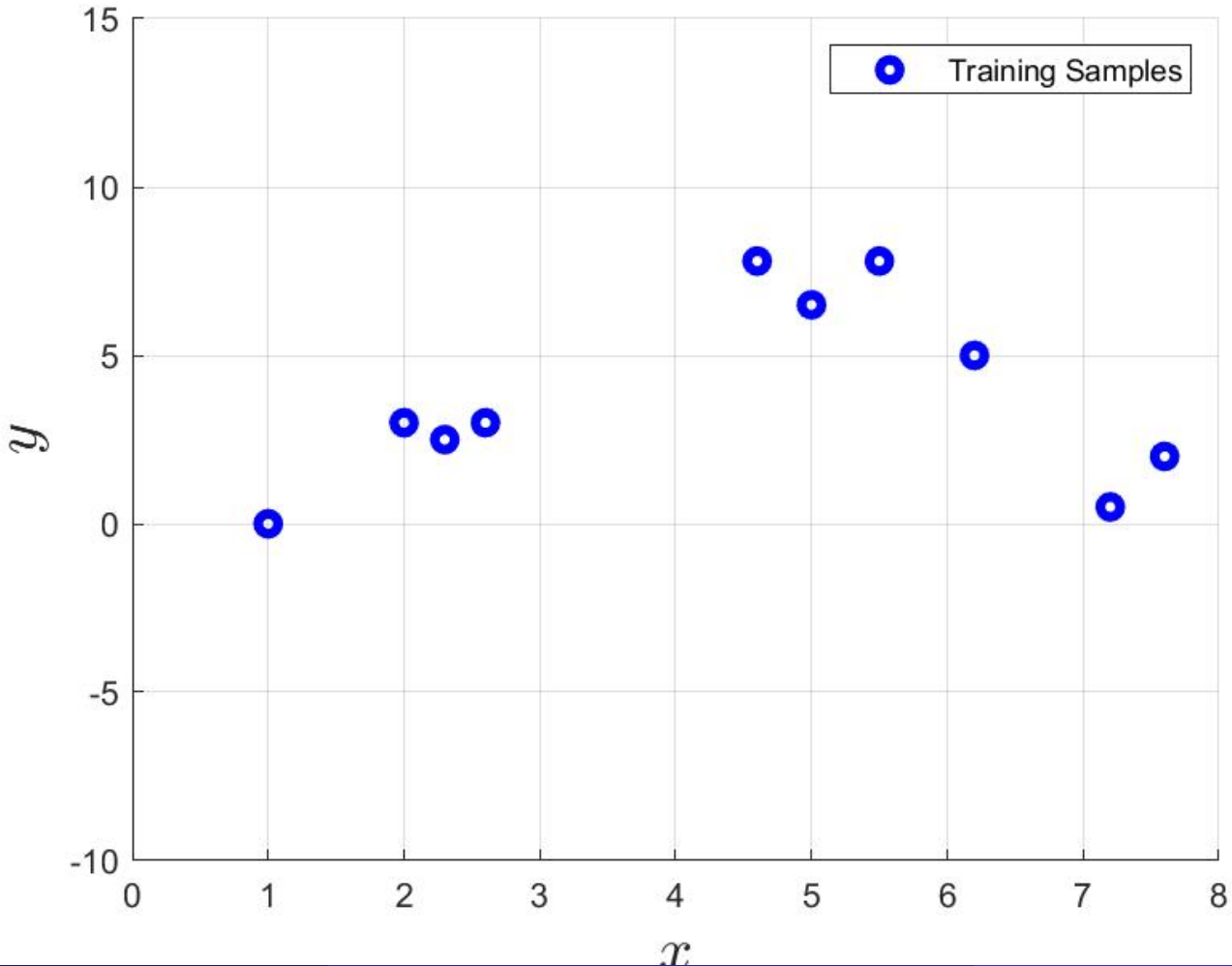
Overfitting

- Overfitting is another problem a model can exhibit
- The model that overfits **predicts very well on the training data but poorly on the testing data**. Several reasons can lead to overfitting, the most important of which are:
  - your model is too complex for the data (for example a very tall decision tree or a very deep or wide neural network often overfit)
  - you have too many features but a small number of training examples

Overfitting and Underfitting

Overfitting: Example

Using this same set of training samples, we can use a complex model such as a high order polynomial model for fitting.



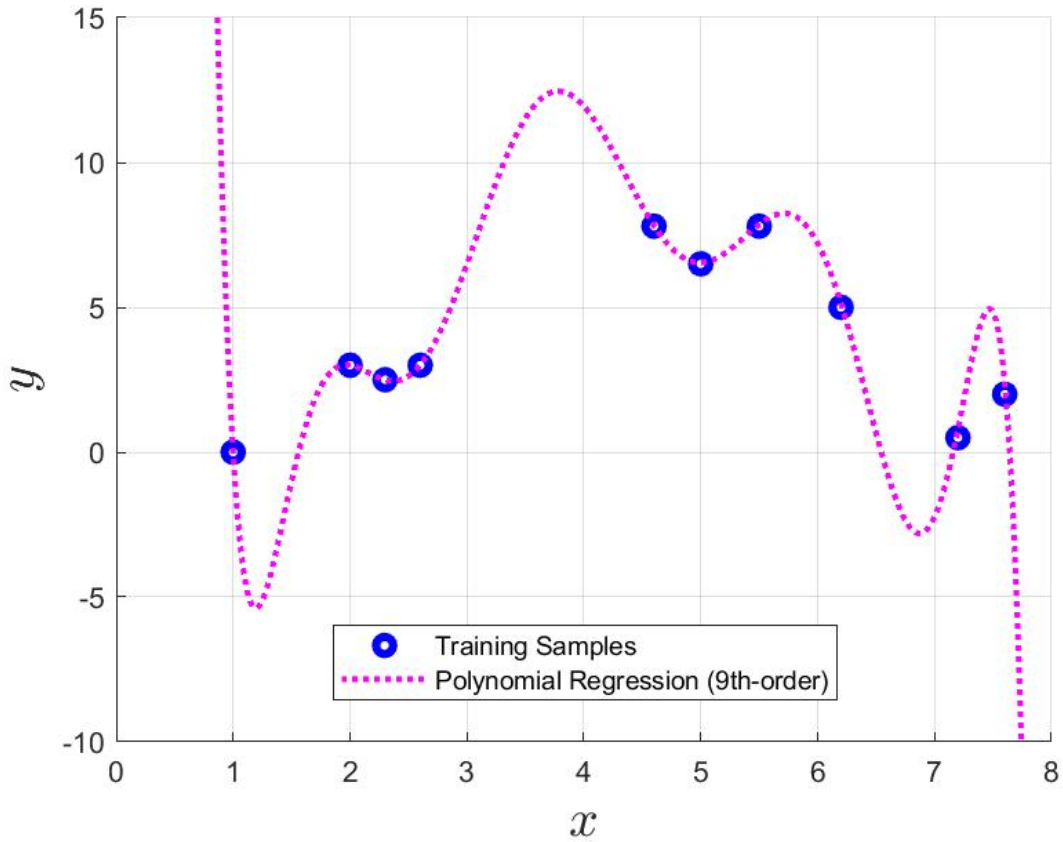
Overfitting and Underfitting

Overfitting: Example

Learning outcome: case 2 (polynomial regression)

Here is an example of overfitting:

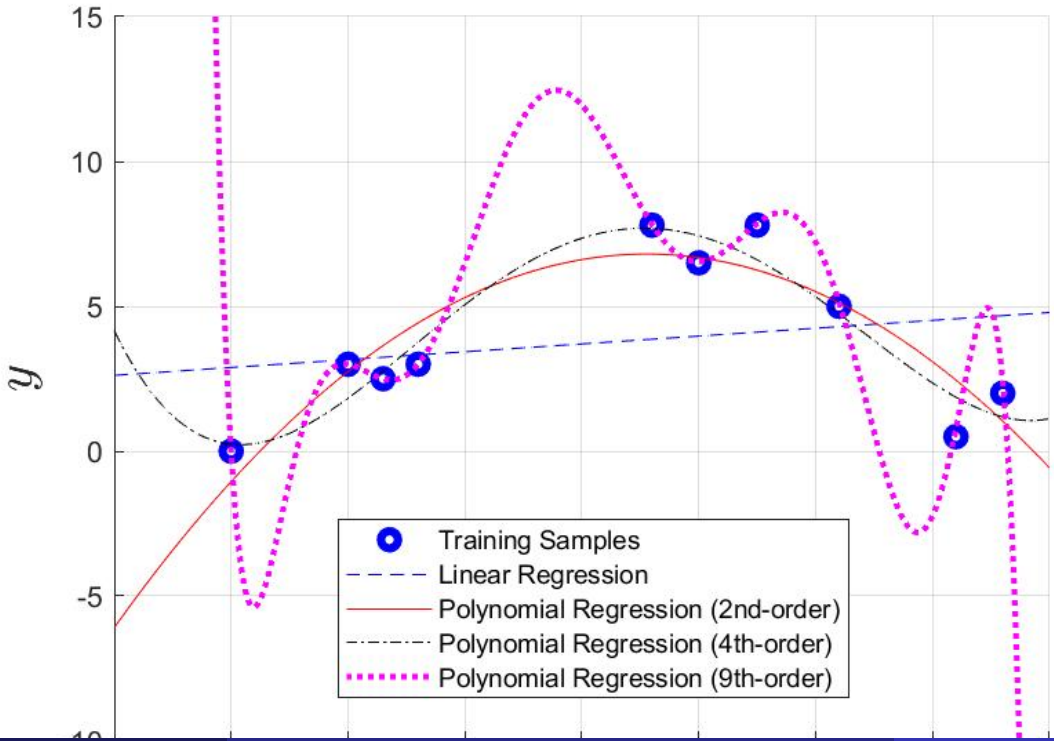
- A 9th order polynomial model (complex) is used to fit the same data.
- The fitted outcome appears more complex than the data distribution.



Overfitting and Model Complexity

More cases: polynomial regression of different orders
Comparing several models of different complexity:

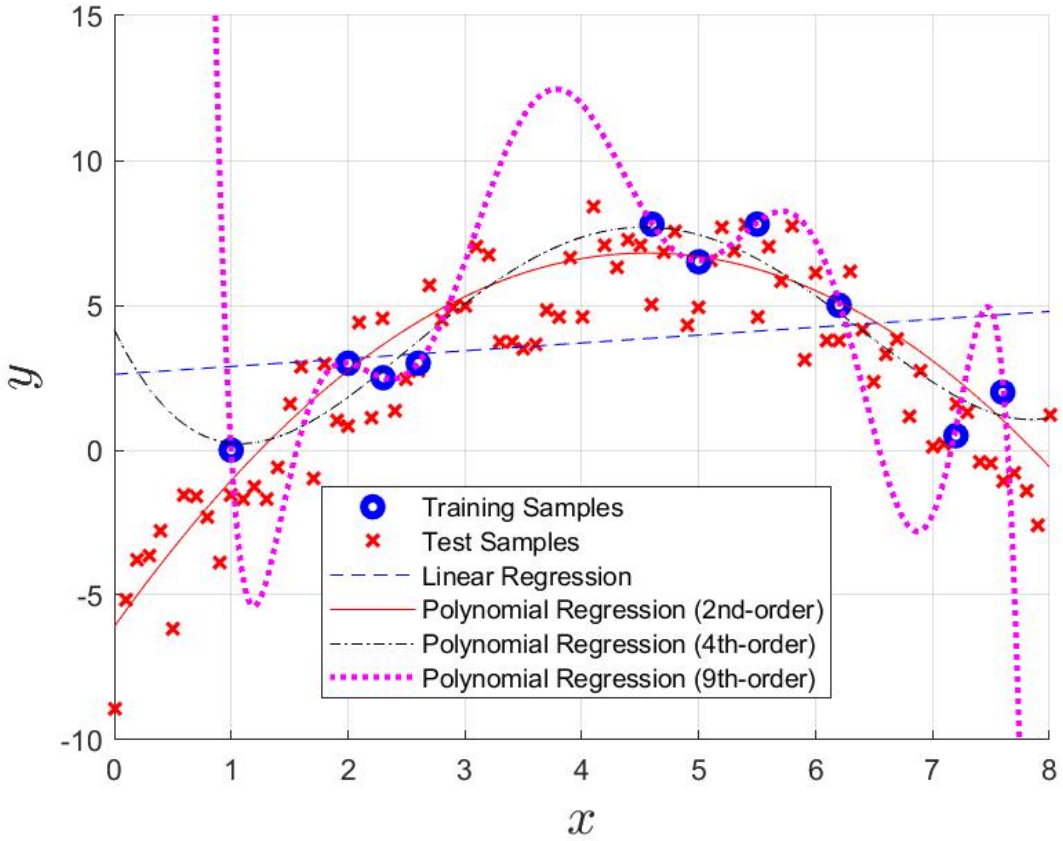
- The linear regression model can also be thought of as the 1st order polynomial model.
- Among these several models of different complexity (order of the polynomials), the 2nd and the 4th order polynomial models appear to fit better than the linear regression and the 9th order polynomial model.



Overfitting and Model Complexity

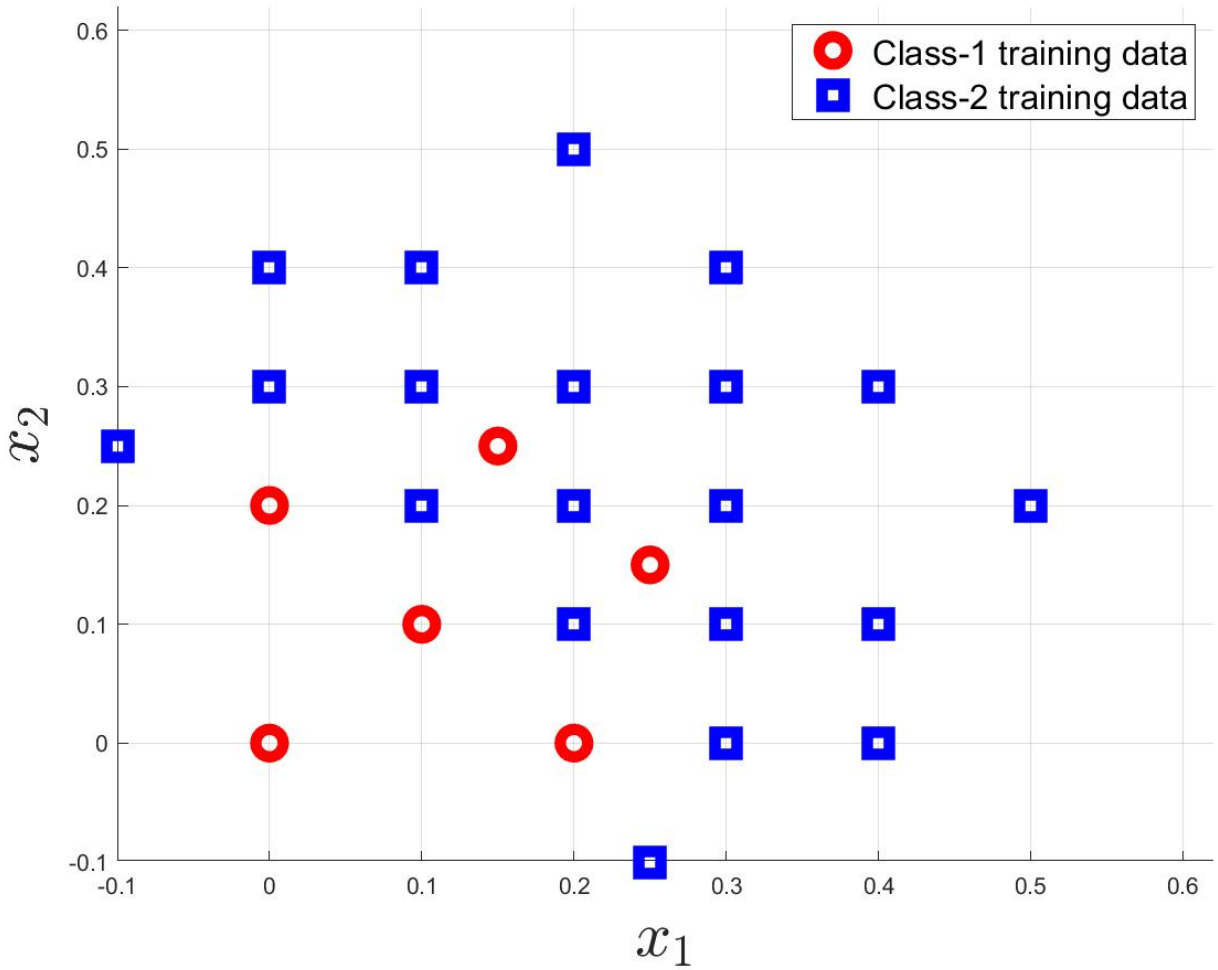
Test data revealed: polynomial regression of different orders

- Here the "x-points" are the test data that you are going to predict.
- Under this circumstance, which model do you think will predict better?



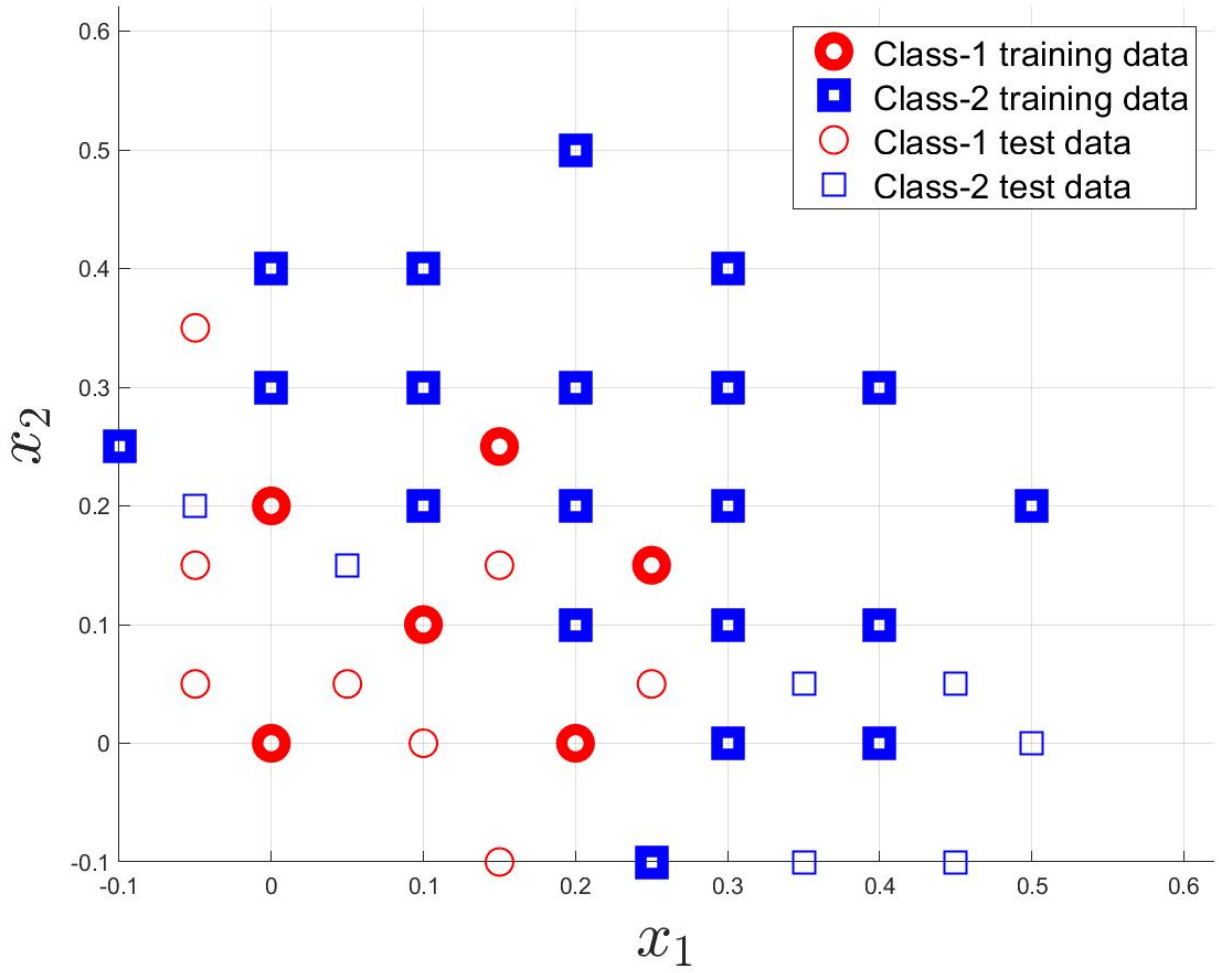
Overfitting and Model Complexity

Training Data (2 dimensional)



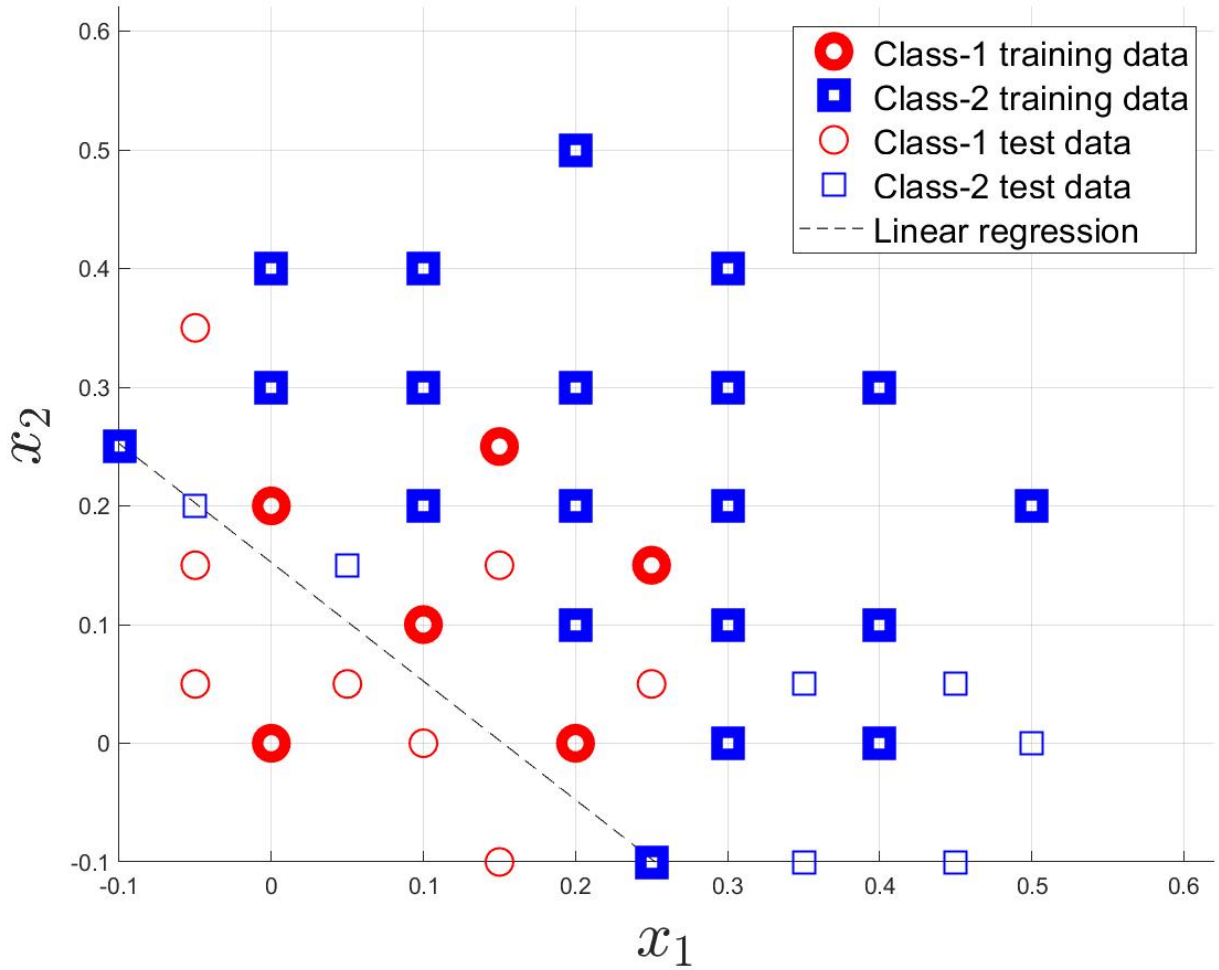
Overfitting and Model Complexity

Training and Test Data



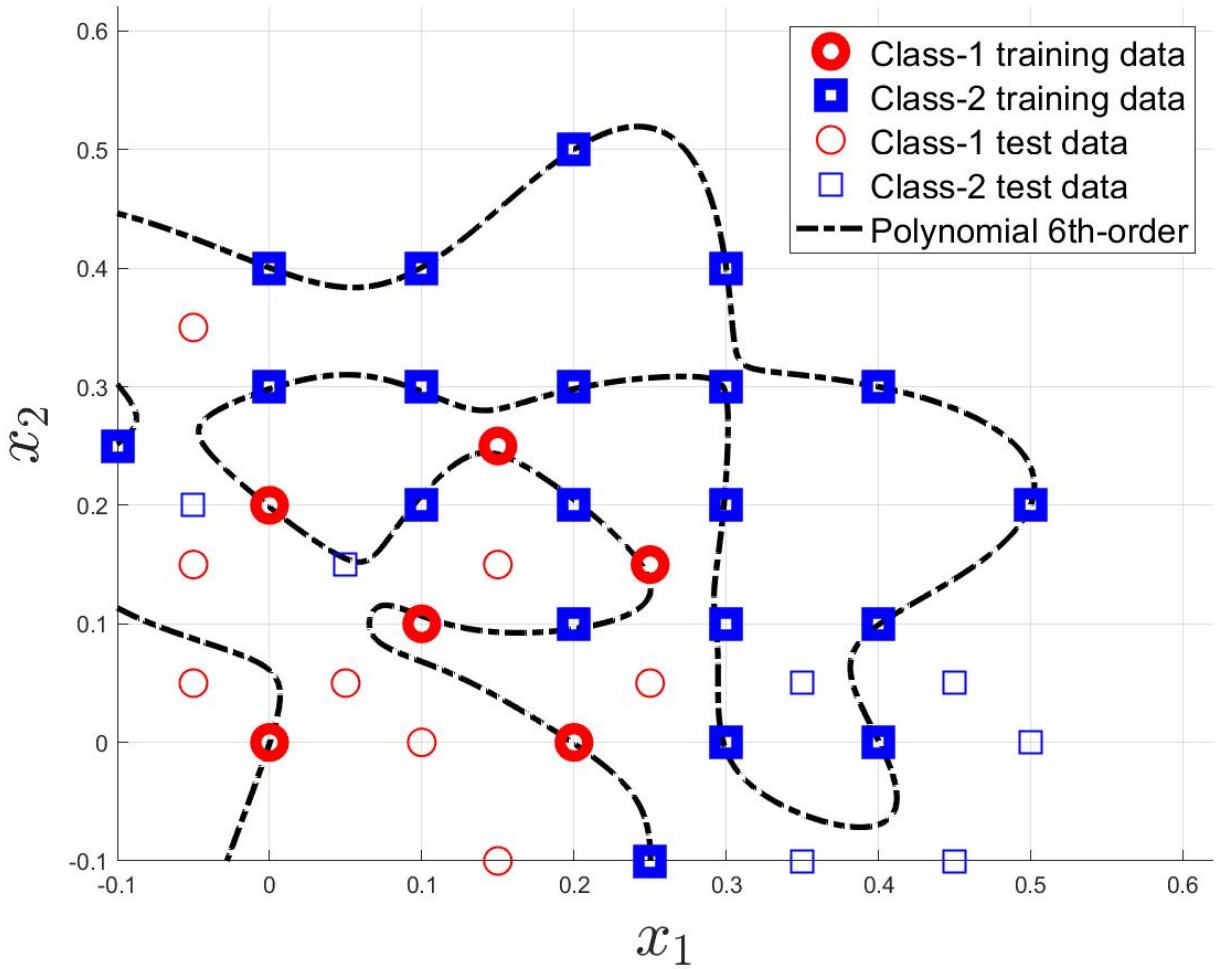
Overfitting and Model Complexity

Underfit



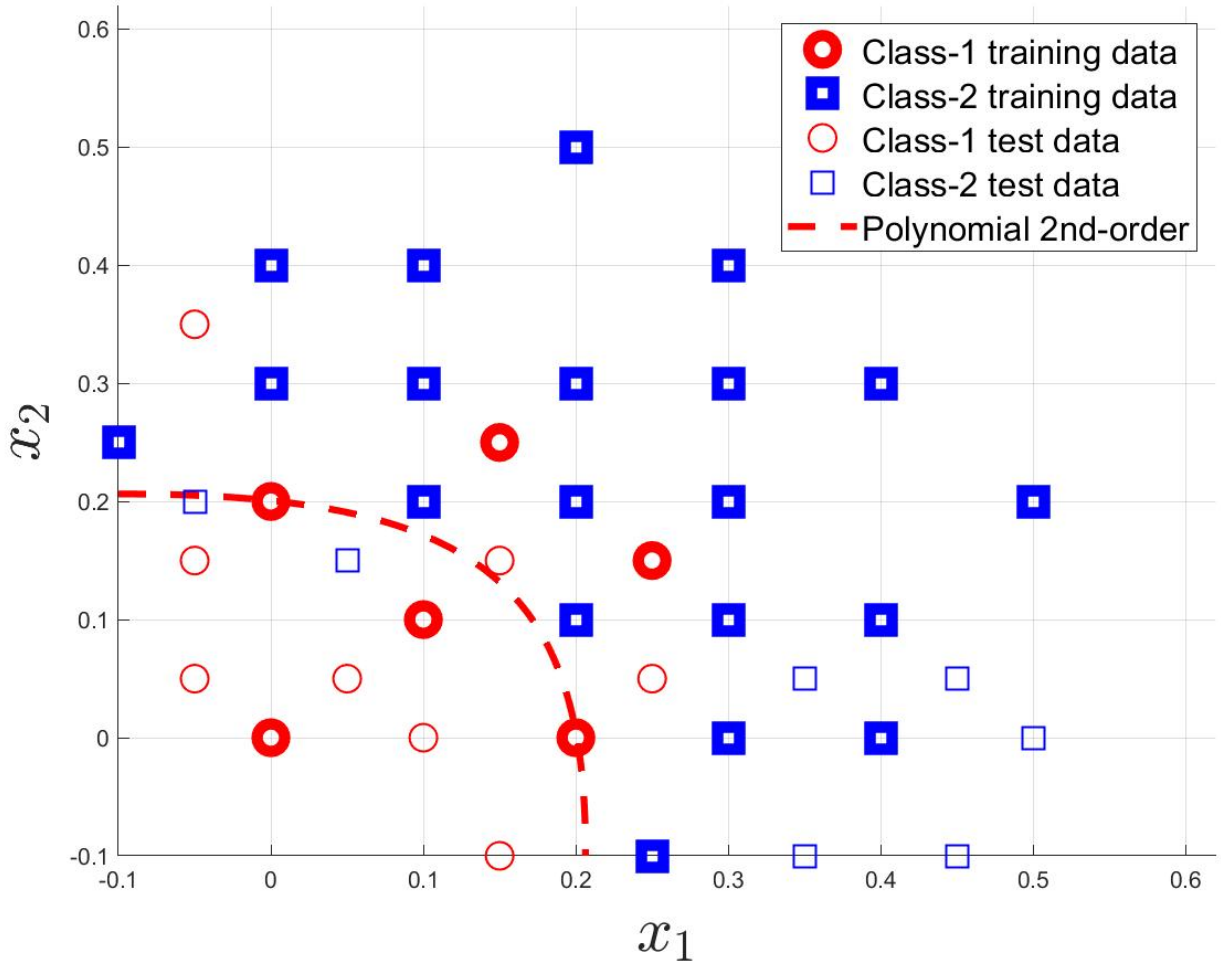
Overfitting and Model Complexity

Overfit



Overfitting and Model Complexity

Good fit



1. Overfitting, underfitting and model complexity

2. Bias-variance trade-off
   - Experimental observations
   - Statistical analysis

1. Overfitting, underfitting and model complexity

2. Bias-variance trade-off
   - Experimental observations
   - Statistical analysis

Bias-variance Trade-off

Bias and Variance

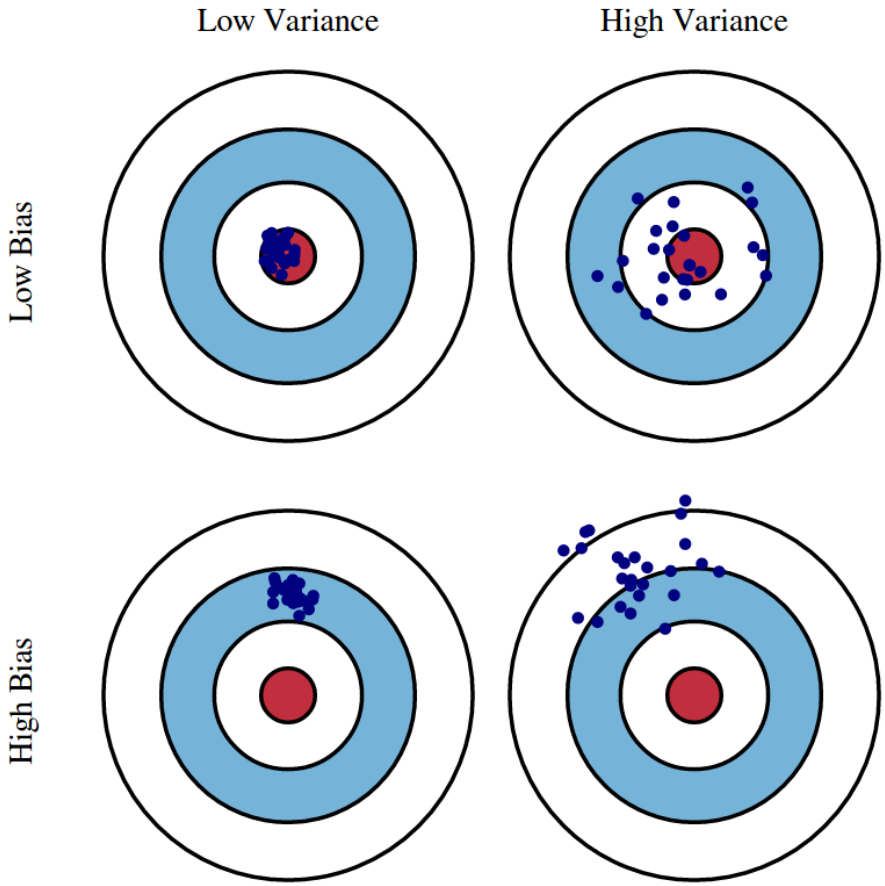


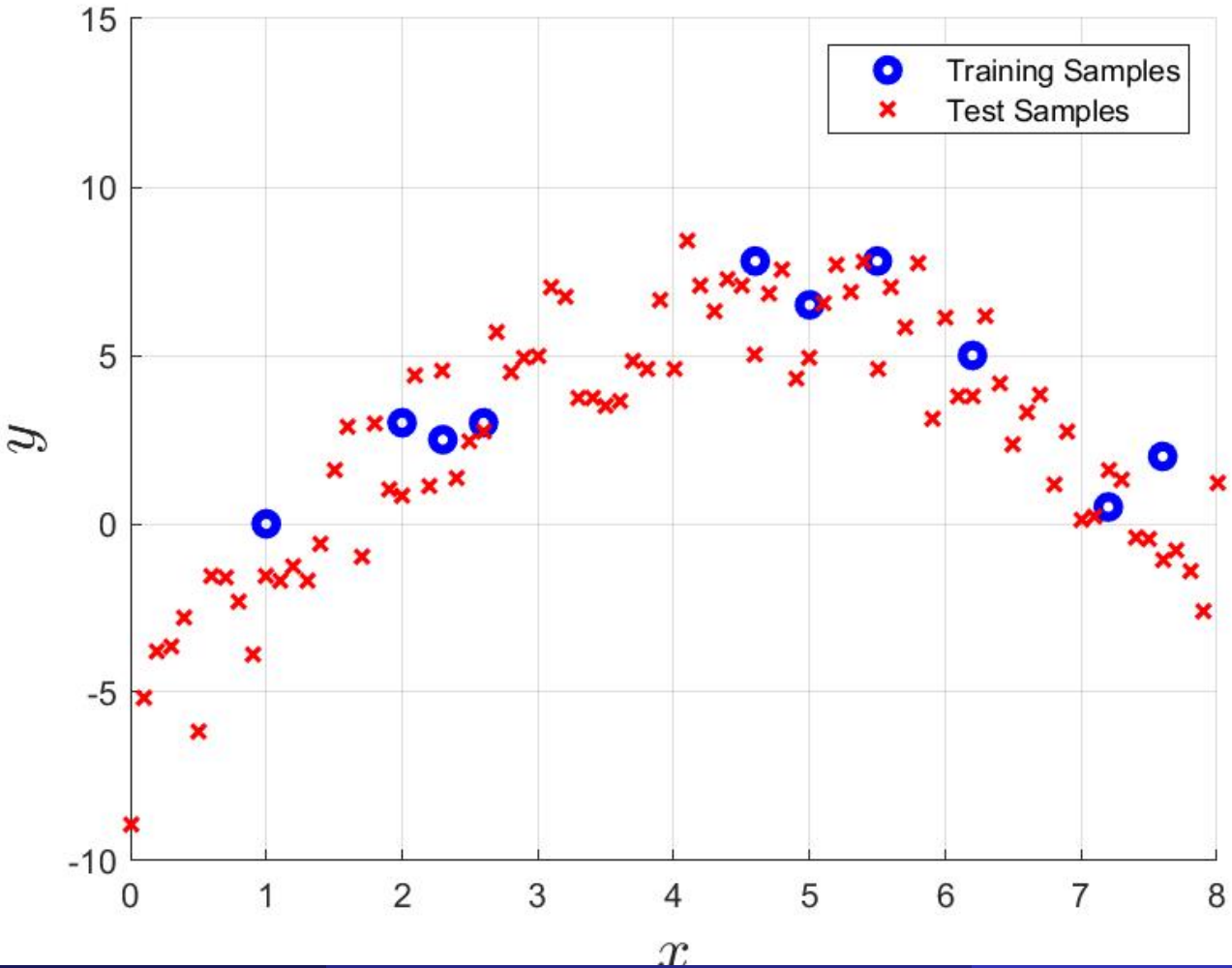
Figure: Graphical illustration of bias and variance

Motivation: To characterize the learning behavior from statistical perspective

Bias-variance Trade-off

Underfitting: Example

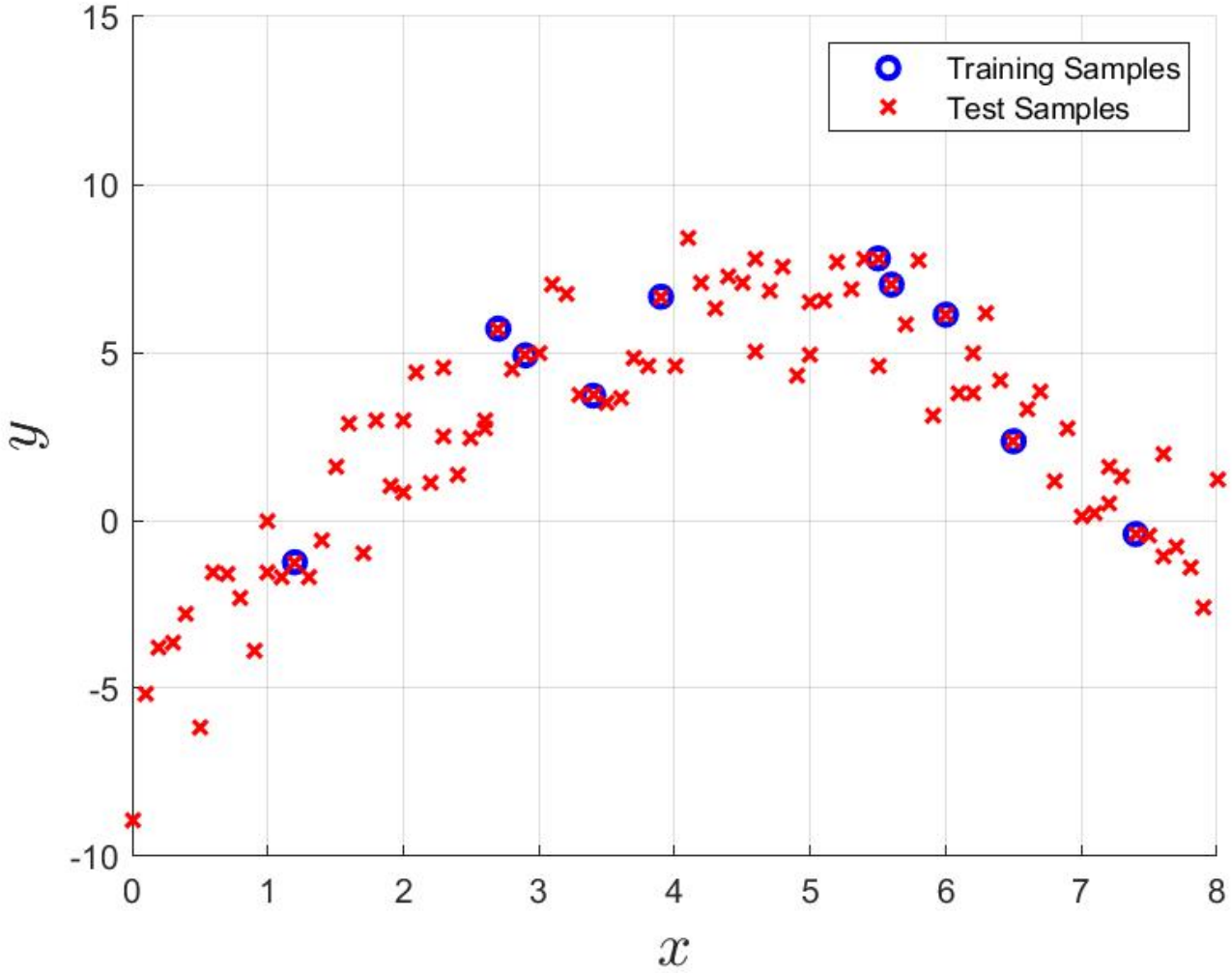
For example, let's repeat the previous 1D experiments for multiple trials. We select these blue samples for training and use the remaining for testing.



Bias-variance Trade-off

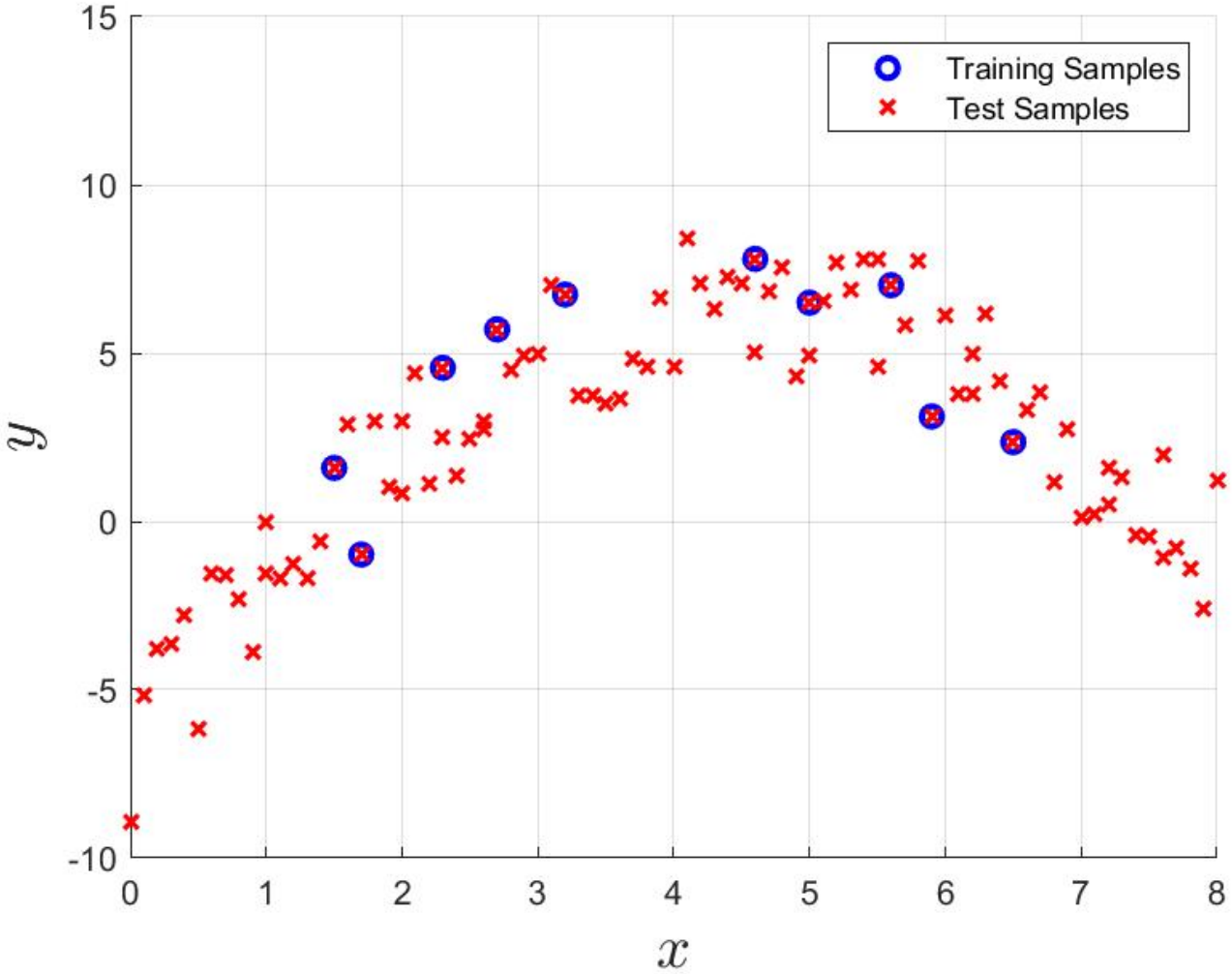
Underfitting: Example

Next, we select another set of blue samples for training and then do the testing.

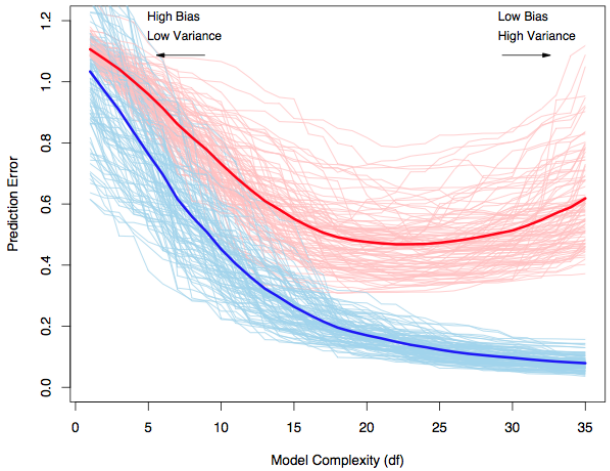


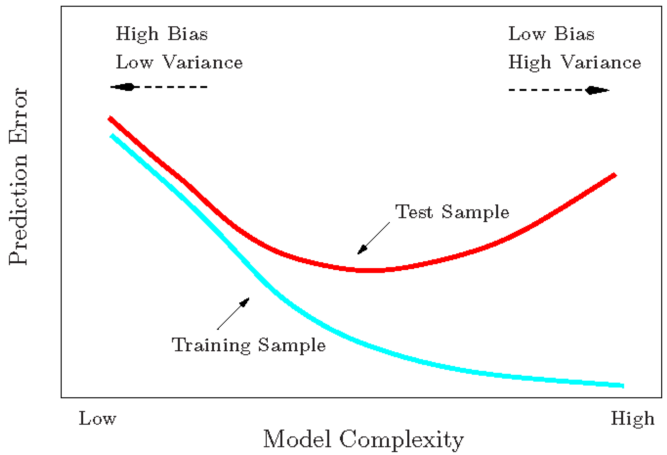
Underfitting: Example

And then another set . . .



Bias-variance Trade-off





When a lot of experiments have been conducted, we observe:

- In general, the training error decreases (towards zero) with the model complexity (*e.g.*, polynomial order).
- In terms of the testing error, for not too big training data, we have the following observations:
  - When the model complexity is low: the test prediction error is high, and it has high bias and low variance;
  - When the model complexity is high (*e.g.*, high order polynomial models): the test prediction error is low, and it has low bias and high variance;
  - However, the test prediction error does not decrease after certain point.

The observation is called bias-variance trade-off. How to explain it?

1. Overfitting, underfitting and model complexity

2. Bias-variance trade-off
   - Experimental observations
   - Statistical analysis

Bias-variance tradeoff

- We are provided by a training dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, which is drawn i.i.d. from some distribution $P(\mathcal{X}, \mathcal{Y})$
- The relationship between the input features $\boldsymbol{x}$ and the output y is

$$y = t(\mathbf{x}) + e, \ e \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

$$p(y|\mathbf{x}) = \mathcal{N}(t(\mathbf{x}), \sigma^2 \mathbf{I}), \tag{2}$$

  where $t(\mathbf{x})$ can be seen as the unknown target function and the mean of $p(y|\mathbf{x})$.
- The goal of machine learning is to learn a hypothesis function based on the training dataset D using some learning algorithm $\mathcal{A}$ (*e.g.*, logistic regression or SVM), *i.e.*,

$$h_D = \mathcal{A}(D)$$

Bias-variance tradeoff

- **Expected hypothesis function** (given $\mathcal{A}$):

$$\bar{h} = E_{D \sim P^n}[h_D] = \int_D h_D p(D) dD$$

- Given the test pair $(\mathbf{x}, y) \sim P(\mathcal{X}, \mathcal{Y})$ and h_D, the **expected test error** is defined as

$$E_{(\mathbf{x},y)\sim P}\left[(h_D(\mathbf{x}) - y)^2\right] = \int_{\mathbf{x}} \int_y (h_D(\mathbf{x}) - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy$$

- Given the test pair $(\mathbf{x}, y) \sim P(\mathcal{X}, \mathcal{Y})$ and $\mathcal{A}$, the **expected test error** is defined as

$$E_{(\mathbf{x},y)\sim P, D\sim P^n}\left[(h_D(\mathbf{x}) - y)^2\right] = \int_D \int_{\mathbf{x}} \int_y (h_D(\mathbf{x}) - y)^2 p(\mathbf{x}, y) p(D) d\mathbf{x} dy dD$$

- We are interested in evaluating the quality of a machine learning algorithm $\mathcal{A}$ with respect to a data distribution $P(\mathcal{X}, \mathcal{Y})$. In the following we will show that this expression decomposes into three meaningful terms.

Bias-variance tradeoff

- The expected test error can be decomposed as follows

$$\begin{aligned}
&E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-y)^2\right] = E_{(\mathbf{x},y),D}\left[[h_D(\mathbf{x})-\bar{h}(\mathbf{x}))+(\bar{h}(\mathbf{x})-y)]^2\right] \\
=&E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-\bar{h}(\mathbf{x}))^2\right] + 2E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-\bar{h}(\mathbf{x}))(\bar{h}(\mathbf{x})-y)\right] \\
&+ E_{(\mathbf{x},y),D}\left[(\bar{h}(\mathbf{x})-y)^2\right]
\end{aligned}$$

- We have

$$\begin{aligned}
&E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-\bar{h}(\mathbf{x}))\cdot(\bar{h}(\mathbf{x})-y)\right] \\
=&E_{(\mathbf{x},y)}\left[E_D[h_D(\mathbf{x})-\bar{h}(\mathbf{x})]\cdot(\bar{h}(\mathbf{x})-y)\right] \\
=&E_{(\mathbf{x},y)}\left[(E_D[h_D(\mathbf{x})]-\bar{h}(\mathbf{x}))\cdot(\bar{h}(\mathbf{x})-y)\right] = 0
\end{aligned}$$

- Replace the above equation to the expected test error, then we have

$$E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-y)^2\right] = E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-\bar{h}(\mathbf{x}))^2\right] + E_{(\mathbf{x},y),D}\left[(\bar{h}(\mathbf{x})-y)^2\right]$$

Bias-variance tradeoff

- We also have

$$E_{(\mathbf{x},y),D}\left[(\bar{h}(\mathbf{x})-y)^2\right]=E_{(\mathbf{x},y),D}\left[[(\bar{h}(\mathbf{x})-t(\mathbf{x}))+(t(\mathbf{x})-y)]^2\right]$$
$$=E_{(\mathbf{x},y)}\left[(t(\mathbf{x})-y)^2\right]+E_{(\mathbf{x},y)}\left[\bar{h}(\mathbf{x})-t(\mathbf{x}))^2\right]+2E_{(\mathbf{x},y)}\left[(t(\mathbf{x})-y)(\bar{h}(\mathbf{x})-t(\mathbf{x})\right.$$
$$=E_{(\mathbf{x},y)}\left[(t(\mathbf{x})-y)^2\right]+E_{(\mathbf{x},y)}\left[\bar{h}(\mathbf{x})-t(\mathbf{x}))^2\right],$$

  where we utilize

$$E_{(\mathbf{x},y)}\left[(t(\mathbf{x})-y)(\bar{h}(\mathbf{x})-t(\mathbf{x}))\right]=E_{\mathbf{x}}\left[E_{y|\mathbf{x}}\left[(t(\mathbf{x})-y)(\bar{h}(\mathbf{x})-t(\mathbf{x}))\right]\right]$$
$$=E_{\mathbf{x}}\left[(t(\mathbf{x})-E_{y|\mathbf{x}}(y))(\bar{h}(\mathbf{x})-t(\mathbf{x}))\right]$$
$$=E_{\mathbf{x}}\left[(t(\mathbf{x})-t(\mathbf{x}))(\bar{h}(\mathbf{x})-t(\mathbf{x}))\right]=0$$

- Finally, we have

$$E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-y)^2\right]$$
$$=E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x})-\bar{h}(\mathbf{x}))^2\right]+E_{(\mathbf{x},y)}\left[(\bar{h}(\mathbf{x})-t(\mathbf{x}))^2\right]+E_{(\mathbf{x},y)}\left[(t(\mathbf{x})-y)^2\right]$$

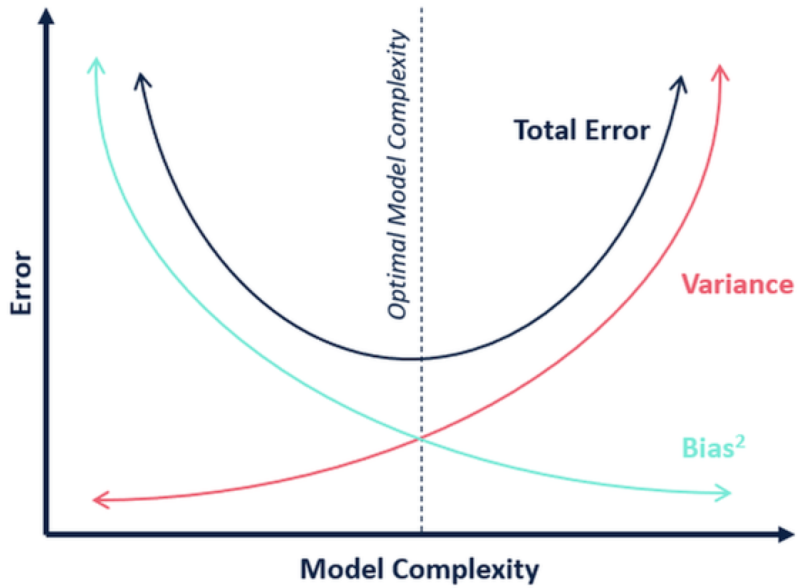
- Above three terms are variance, bias^2, noise, respectively.

Bias-variance tradeoff

$$
\begin{aligned}
& E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x}) - y)^2\right] \\
=& E_{(\mathbf{x},y),D}\left[(h_D(\mathbf{x}) - \bar{h}(\mathbf{x}))^2\right] + E_{(\mathbf{x},y)}\left[(\bar{h}(\mathbf{x}) - t(\mathbf{x}))^2\right] + E_{(\mathbf{x},y)}\left[(t(\mathbf{x}) - y)^2\right]
\end{aligned}
$$

- Variance: Captures how much your classifier changes if you train on a different training set, how "over-specialized" is your classifier to a particular training set (overfitting). In other words, variance is due to your fixed **training set**.
- bias^2: What is the inherent error that you obtain from your classifier even with infinite training data? This is due to your classifier being "biased" to a particular kind of solution (*e.g.*, linear classifier). In other words, bias is inherent to your **model**.
- Noise: How big is the data-intrinsic noise? This error measures ambiguity due to your data distribution and feature representation. You can never beat this, it is an aspect of the data.

Bias-variance tradeoff

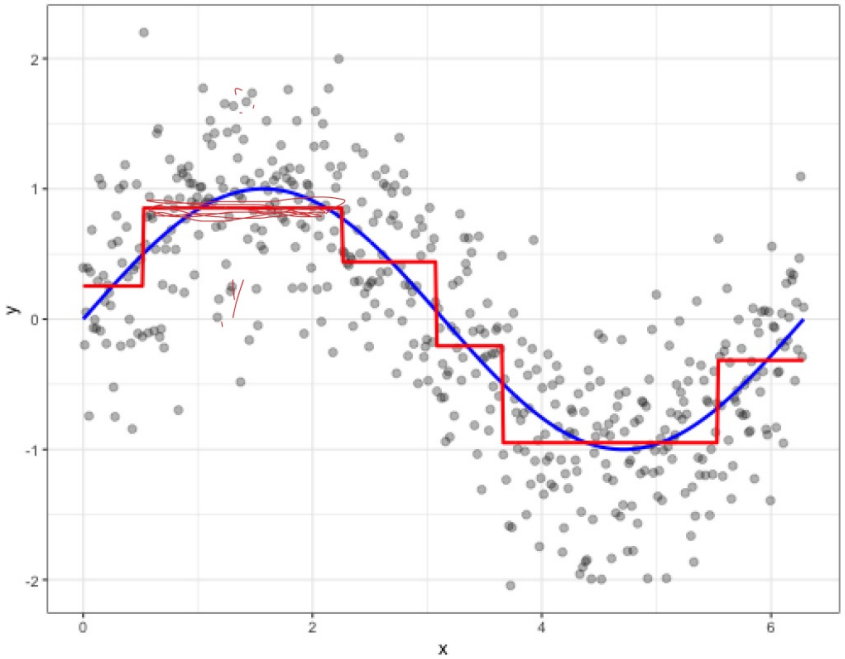


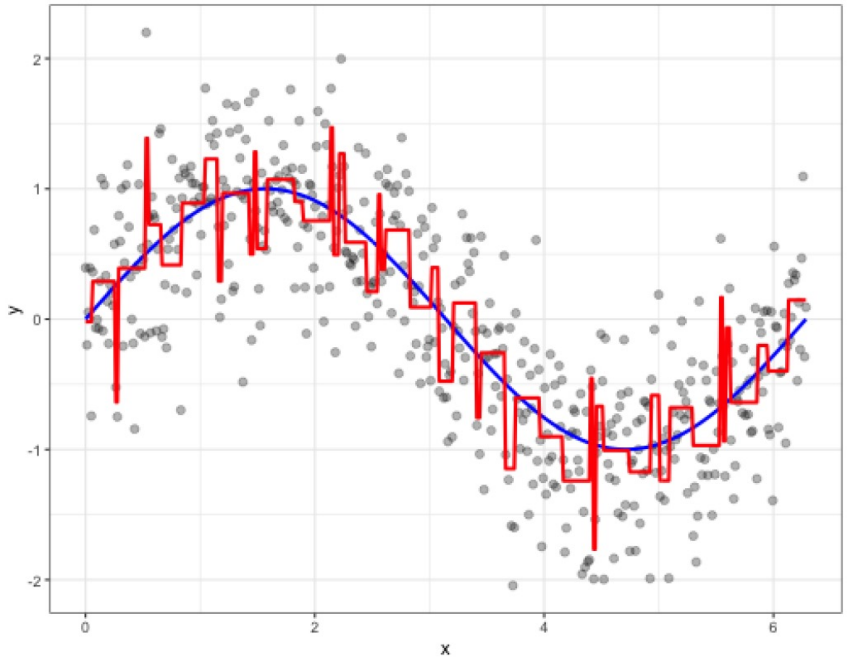
- When the model complexity increases, the differences of the models trained on different training sets will increase, *i.e.*, the variance increases
- When the model complexity increases, the average prediction of the models trained on different training sets will be more close to the target value, *i.e.*, the bias decreases
- Thus, the total test error will firstly decrease then increase along with the increase of model complexity. It implies that you should choose a model with suitable complexity.

Bias-variance tradeoff

Example: Single Decision Trees

- Single pruned trees have high bias and low variance, *i.e.*, underfitting
- Single deep trees have low bias and high variance, *i.e.*, overfitting

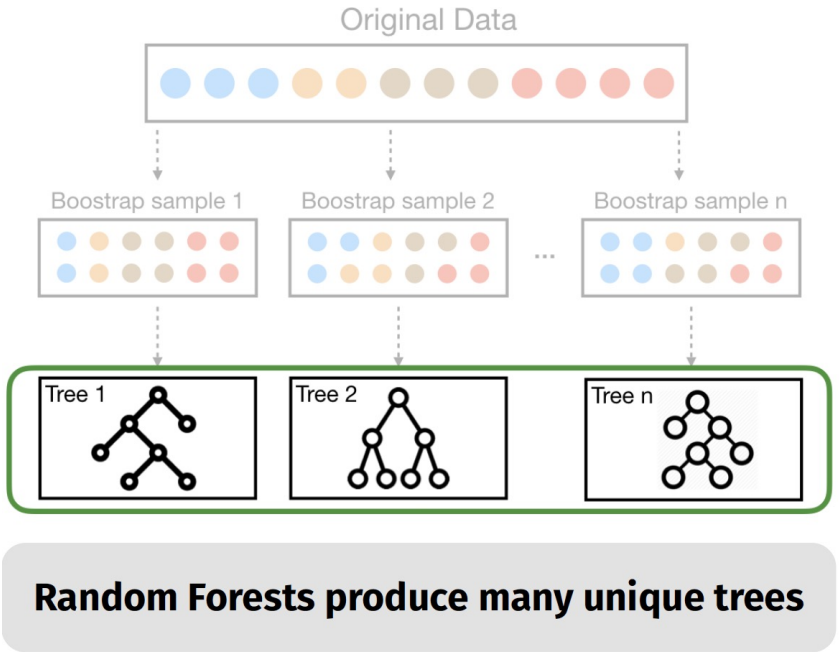




Bias-variance tradeoff

Example: Random Forests

- Random forests introduce data and model randomness (but not change the model complexity). Thus, its variance is significantly reduced, compared to single trees.
- However, there is no guarantee of reducing bias, as different trees are independent and the overall model complexity is not increased.
- Actually, there is an ensemble model called Boosting, which could reduce the bias. If interested, you can read this blog.



Bias-variance tradeoff

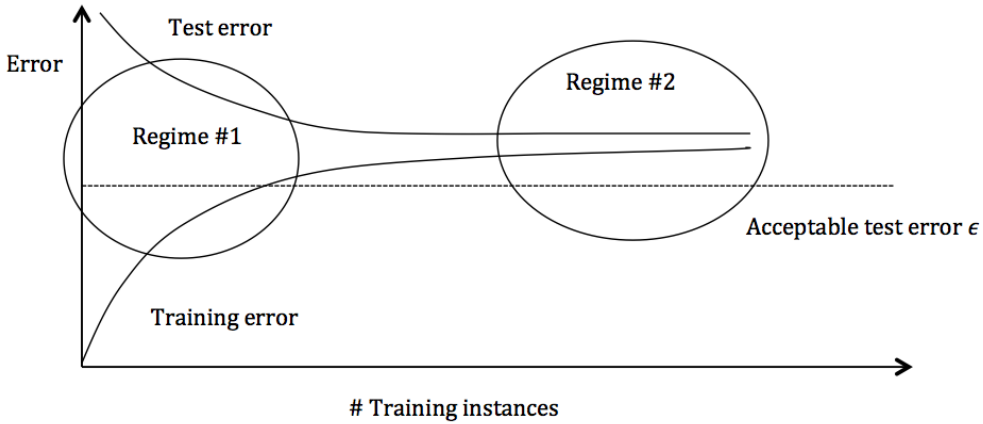


Figure: Given the same model, the changes of training and test error along with the increase of the number of training instances.

Regime 1 (High variance)

In Regime 1, the cause of the poor performance is high variance, *i.e.*, overfitting.

Symptoms:

- Training error is much lower than test error
- Training error is lower than a ϵ
- Test error is above ϵ

Remedies:

- Add more training instances
- Reduce model complexity – complex models are prone to high variance

Bias-variance tradeoff

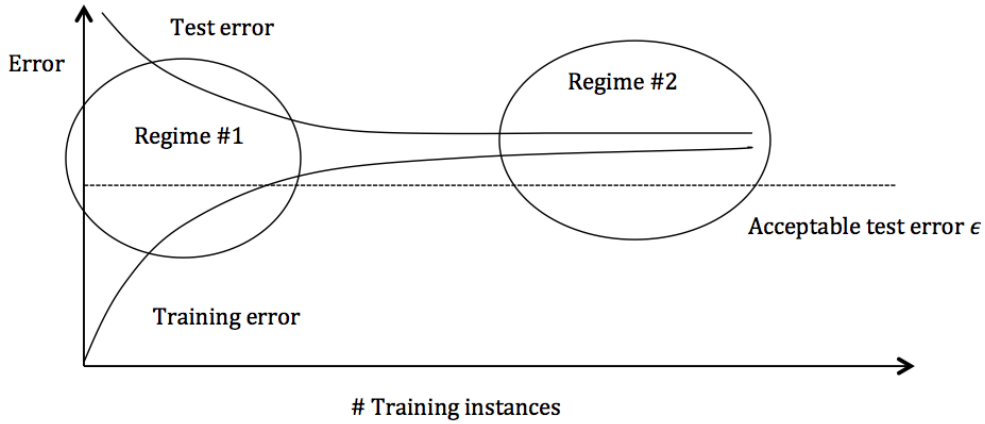


Figure: Given the same model, the changes of training and test error along with the increase of the number of training instances.

Regime 2 (High bias)

Unlike the first regime, the second regime indicates high bias: the model being used in not powerful enough to produce an accurate prediction, *i.e.*, underfitting.

Symptoms:

- Training error is higher than a ϵ

Remedies:

- Add more features
- Use more complex model (*e.g.*, kernel model, non-linear models)

Bias-variance tradeoff

Exercise:

- Suppose we randomly sample a training set D from some unknown distribution. For each training set D we sample, we train a regression model h_D to predict y from x (one dimensional). We repeat this process 10 times resulting in 10 trained models.
- Recall that $y = t(x) + \epsilon$, where $\epsilon \in \mathcal{N}(0, \sigma^2)$. Here, we specify $\sigma^2 = 0.5$.
- For a new test sample (x, y), we define its mean squared error (MSE) by different models as

$$MSE(x, y) = E_D[(h_{D_i}(x) - y)^2].$$

- It can observed that $E_{(x,y),D}[(h_{D_i}(x) - y)^2] = E_{(x,y)}[MSE(x, y)]$.
- The empirical estimation of MSE based on above 10 trained models is

$$\widehat{MSE}(x, y) = \frac{1}{10}\sum_{i=1}^{10}(h_D(x) - y)^2.$$

Bias-variance tradeoff

Exercise:

- For a new test sample $(x, y) = (5, 10)$ sampled from the same distribution that generated the training sets, we suppose $t(x = 5) = 9.5$, and ϵ is instantiated as 0.5.
- Suppose the predictions of this new test sample based on the 10 trained models are 9, 11, 23, 6, 8, 12, 10, 4, 13, 7, respectively.
- Based on this 10 trials, please compute the empirical mean squared error (MSE), Bias^2 and Variance on this test sample.

Bias-variance tradeoff

Exercise:

- For a new test sample $(x, y) = (5, 10)$ sampled from the same distribution that generated the training sets, we suppose $t(x = 5) = 9.5$, and ϵ is instantiated as 0.5.
- Suppose the predictions of this new test sample based on the 10 trained models are 9, 11, 23, 6, 8, 12, 10, 4, 13, 7, respectively.
- Based on this 10 trials, please compute the empirical mean squared error (MSE), Bias^2 and Variance on this test sample.
- Average Prediction $= (9+11+23+6+8+12+10+4+13+7)/10 = 10.3$
  Bias^2 $= (10.3–9.5)^2 = 0.64$
  Variance $= \frac{1}{10}[(9-10.3)^2+(11-10.3)^2+(23-10.3)^2+(6-10.3)^2+(8-10.3)^2+(12-10.3)^2+(10-10.3)^2+(4-10.3)^2+(13-10.3)^2+(7-10.3)^2] = 24.81$
  Empirical MSE $= \frac{1}{10}[(9-10)^2+(11-10)^2+(23-10)^2+(6-10)^2+(8-10)^2+(12-10)^2+(10-10)^2+(4-10)^2+(13-10)^2+(7-10)^2] = 24.9.$

Bias-variance tradeoff

References:

- https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lectur
  html
- http://scott.fortmann-roe.com/docs/BiasVariance.html#fn:3