# Validation and Interpretation

# Bias-Variance Decomposition

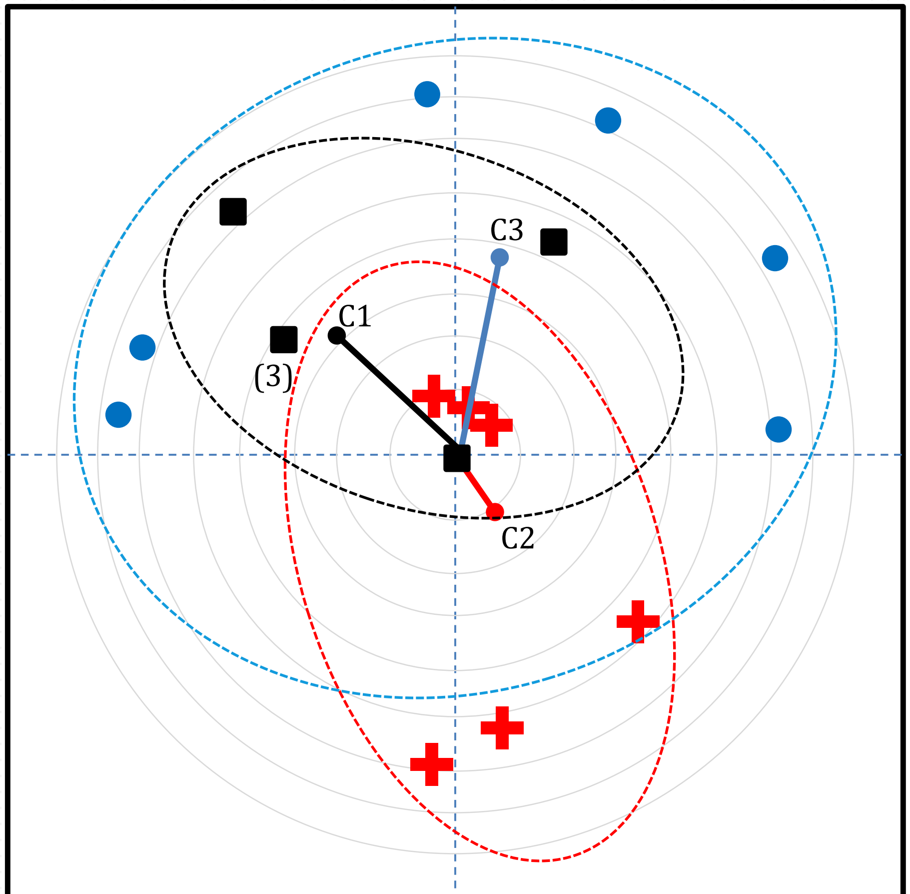
*A formal method for analyzing the prediction error of a predictive model.*

# The Intuition

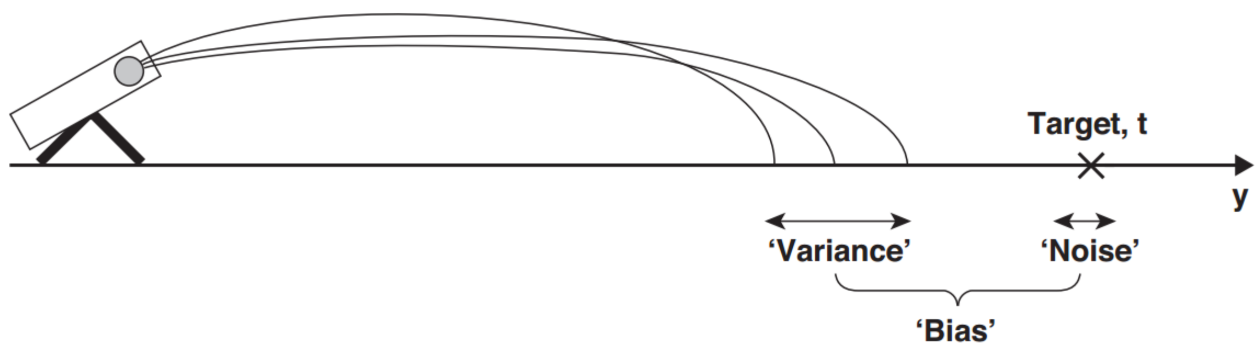
**Bias**

**Variance**

**Noise**



# The Intuition



# The Intuition

$$\boldsymbol{d_{f,\theta}(y,t) = \mathbf{Bias}_\theta + \mathbf{Variance}_f + \mathbf{Noise}_t}$$

- $f$ refers to the amount of force applied
- $\theta$ denotes the angle of the launcher
- $t$ corresponds to the location of the target

# Bias-Variance in Classifiers

- The task of predicting a class label can be analyzed using the same approach
- Predictions may turn out to be correct, while others can be way off the mark
- The error of a classifier can be decomposed as a sum of the three terms previously described
- Classifiers minimize the error in the training set but must be able to generalize to unseen instances

# Bias-Variance in Classifiers

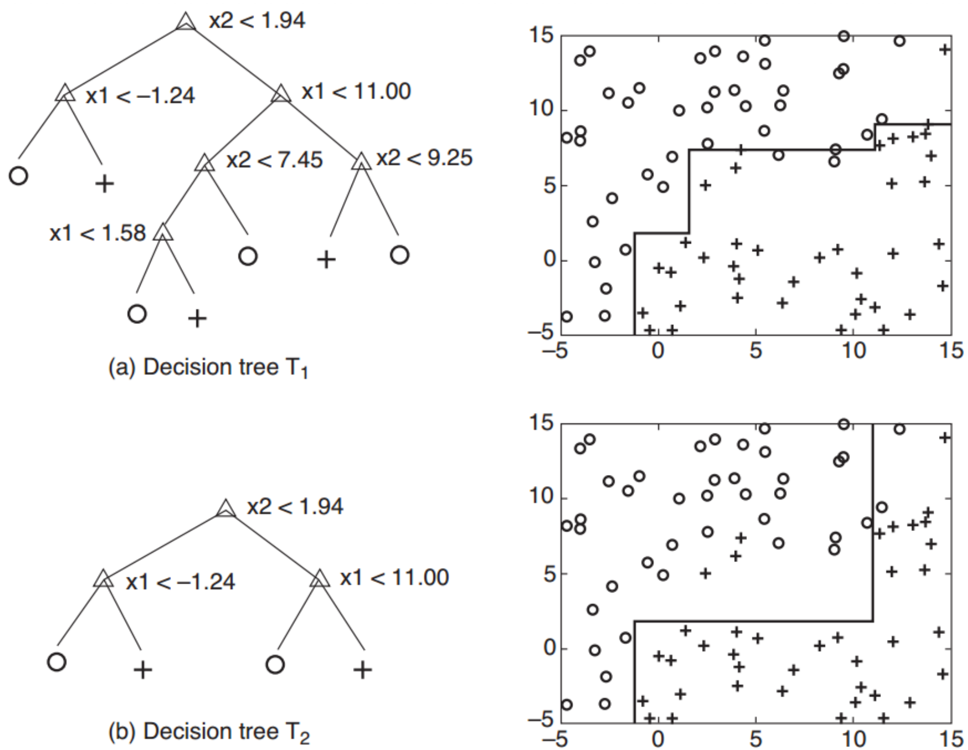



**Figure 5.33.** Two decision trees with different complexities induced from the same training data.

- $T_1$ and $T_2$ are generated from the same training data
- $T_2$ is obtained by pruning $T_1$
- These design choices introduce a bias analogous to that of the projectile launcher into the classifier
- The larger the assumptions made about the decision boundaries, the larger the **bias**
  - $T_2$ has a larger bias

# Bias-Variance in Classifiers

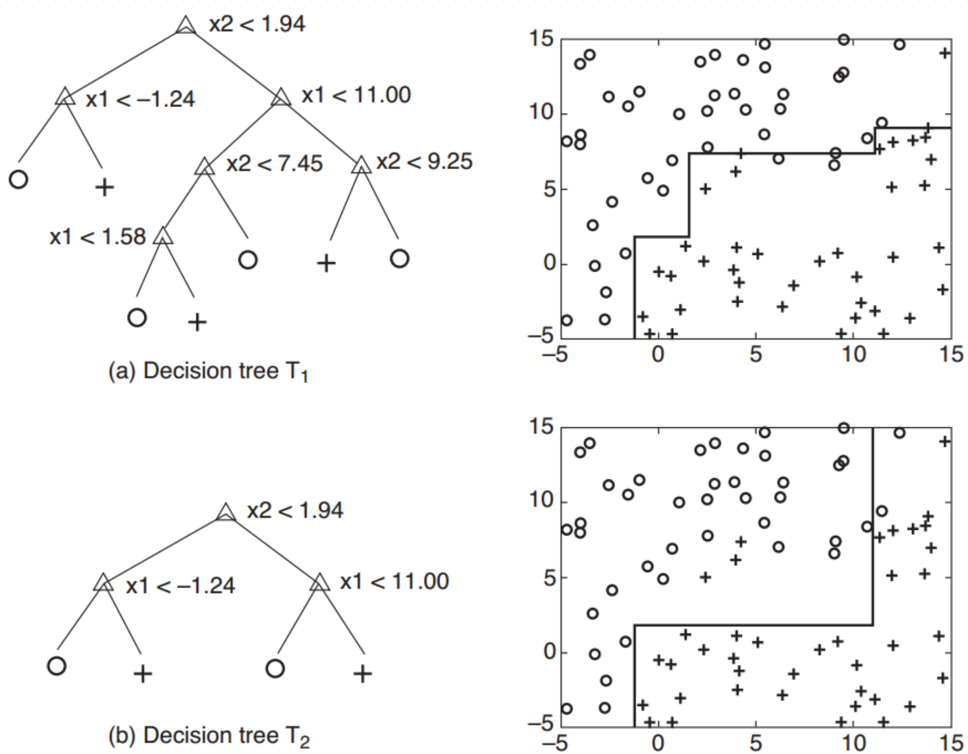
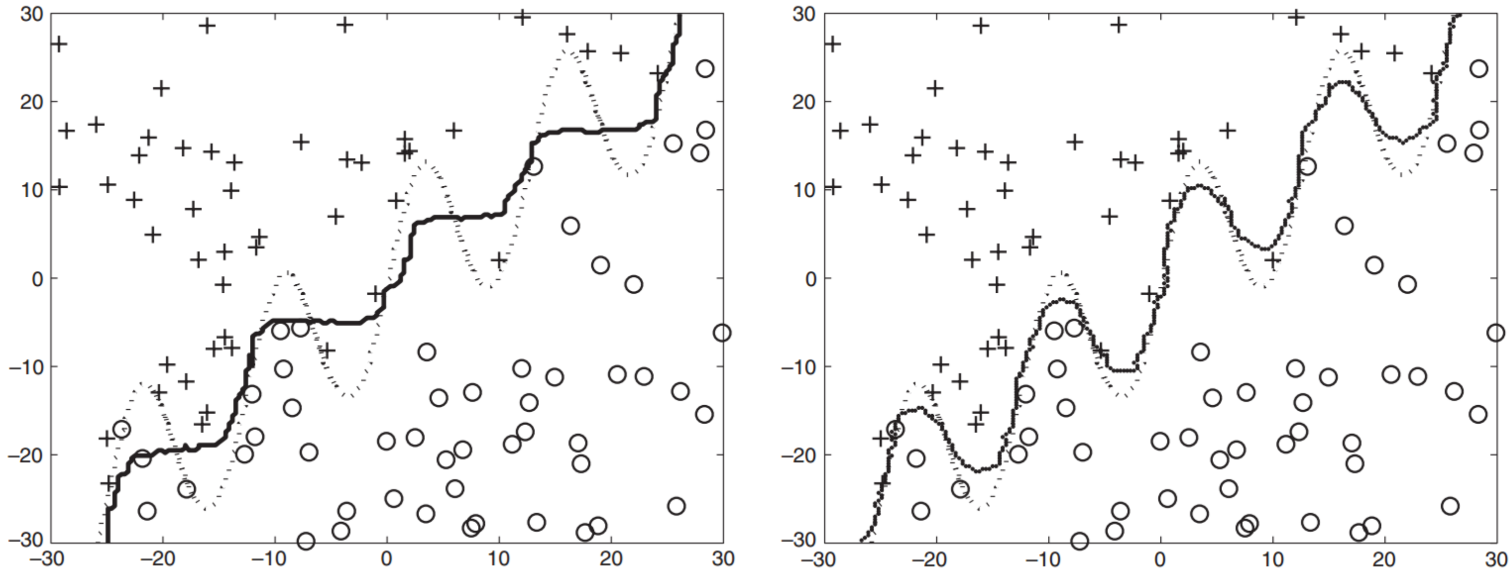



**Figure 5.33.** Two decision trees with different complexities induced from the same training data.

- The expected error of a classifier can be affected by different compositions of the training set leading to different decision boundaries. This is analogous to the **variance** when different amounts of force are applied to the projectile.
- The third component of the expected error (i.e., **noise**) is associated with the intrinsic noise in the class. That is, some instances with the same attributes may have different classes.

# Bias-Variance in Classifiers




(a) Decision boundary for decision tree.

(b) Decision boundary for 1-nearest neighbor.

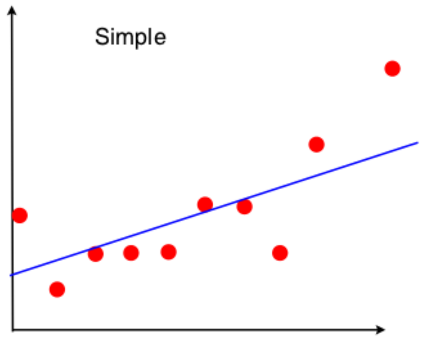
**Figure 5.34.** Bias of decision tree and 1-nearest neighbor classifiers.

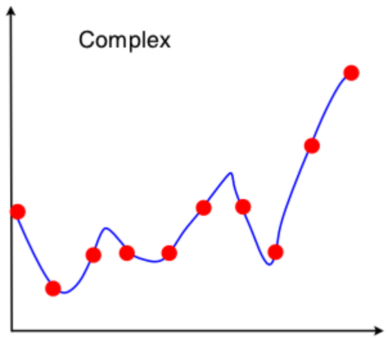
# Generalization

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Slide credit: L. Lazebnik

# Bias-Variance Tradeoff




- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Slide credit: D. Hoiem

# Bias-Variance Tradeoff

$$E(MSE) = \textbf{noise}^2 + \textbf{bias}^2 + \textbf{variance}$$

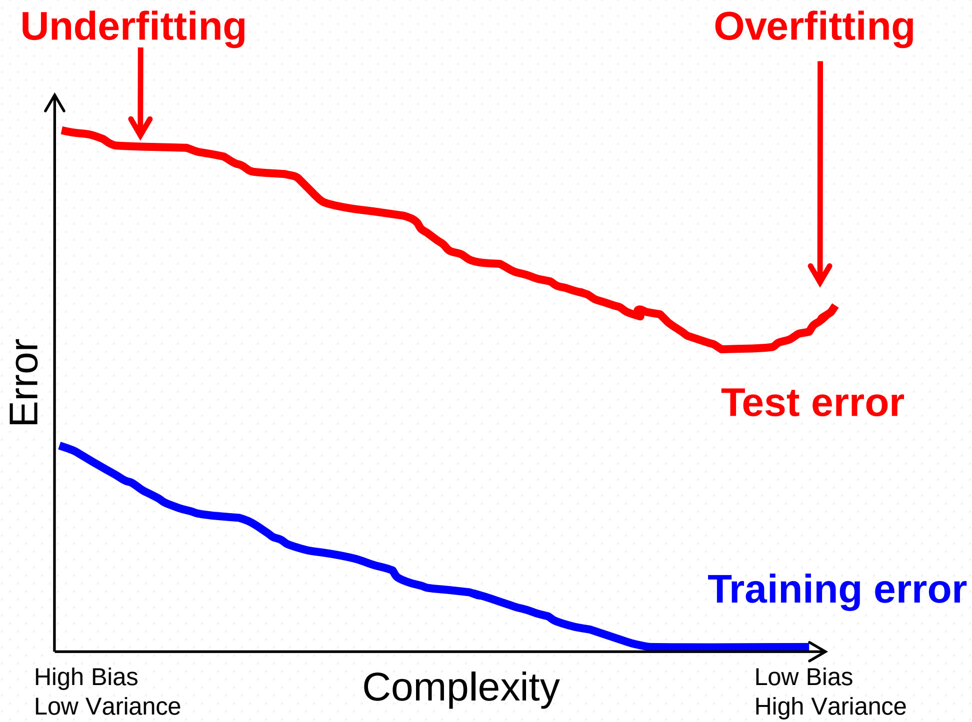
Unavoidable error

Error due to incorrect assumptions

Error due to variance of training samples

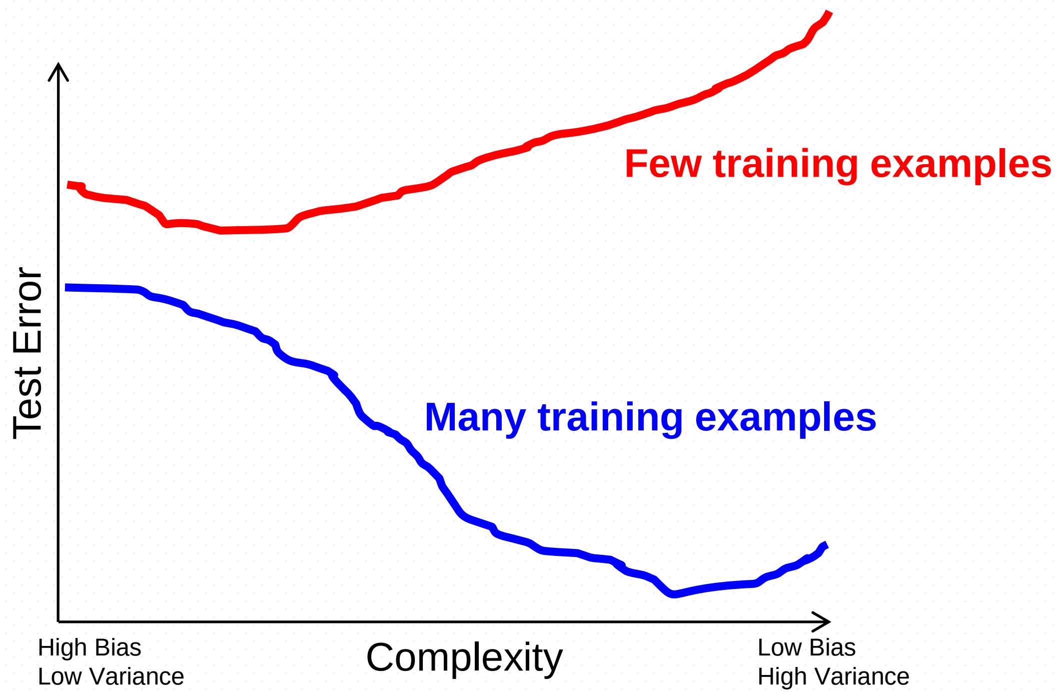
Slide credit: D. Hoiem

# Bias-Variance Tradeoff



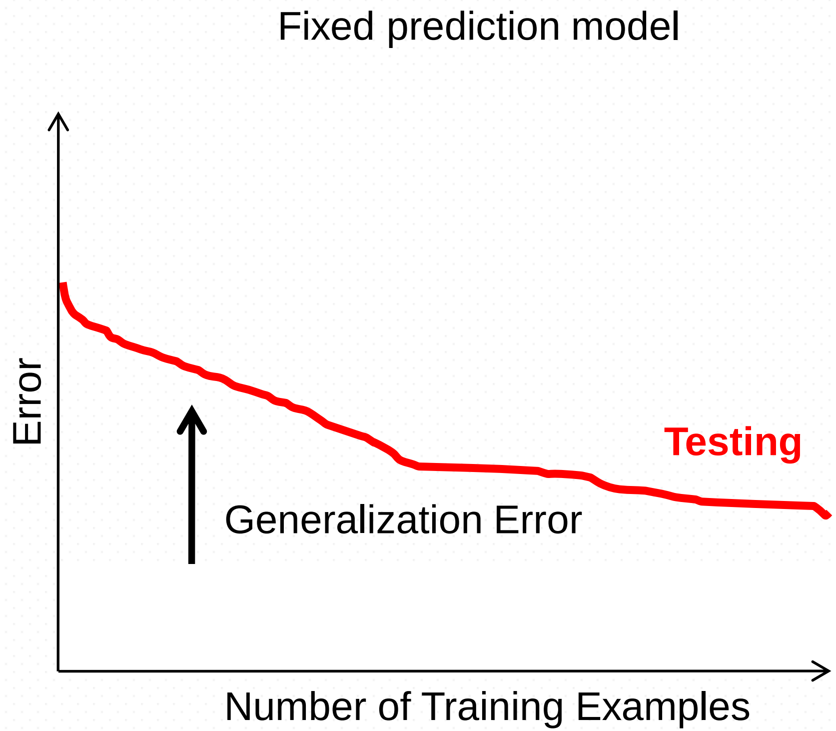

Slide credit: D. Hoiem

# Bias-Variance Tradeoff




Slide credit: D. Hoiem

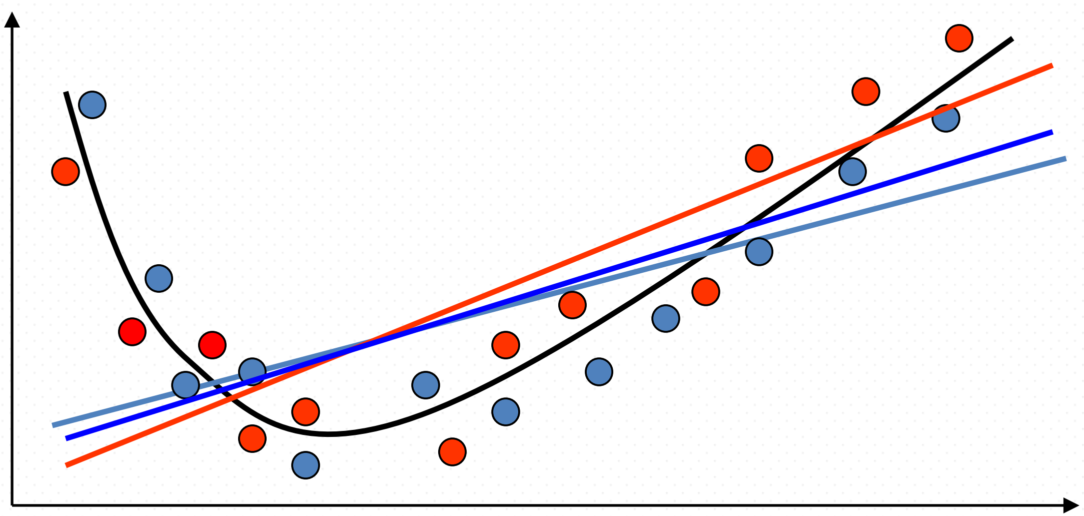
# Effect of Training Size




Slide credit: D. Hoiem

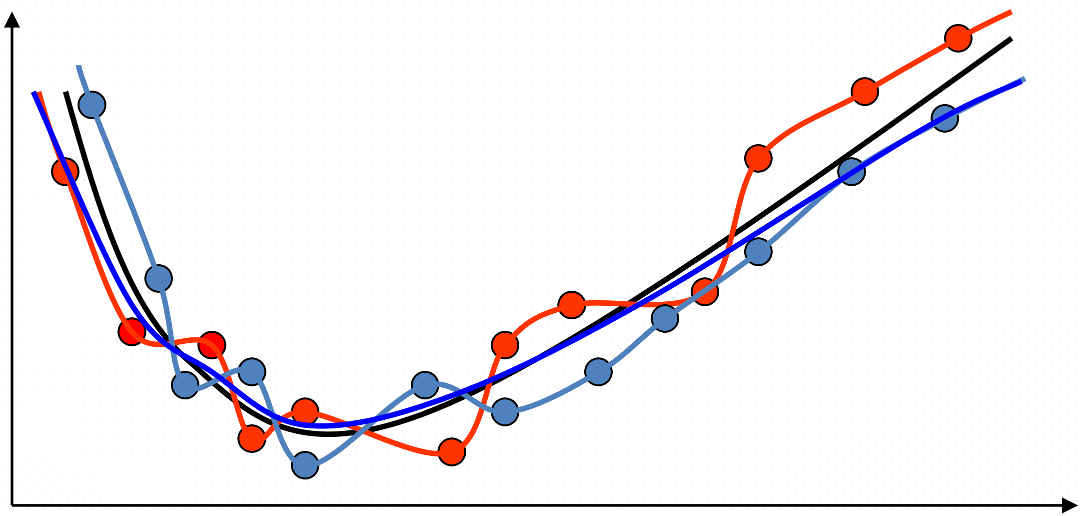
# Illustration (1)

Low variance, high bias method ⇒ underfitting



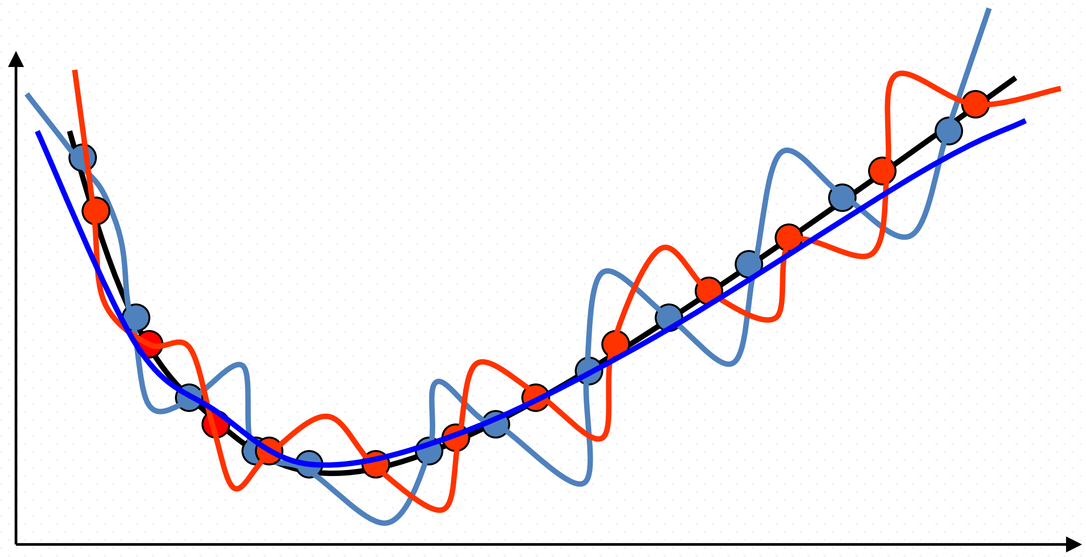
# Illustration (2)

Low bias, high variance method ⇒ overfitting



# Illustration (3)

No noise doesn't imply no variance (but less variance)



# Remember…

- No classifier is inherently better than any other: you need to make assumptions to generalize


- Three kinds of error
  - **Inherent:** unavoidable
  - **Bias:** due to over-simplifications
  - **Variance:** due to inability to perfectly estimate parameters from limited data

# How to Reduce Variance?

- Choose a simpler classifier
- Regularize the parameters
- Get more training data

# The Jackknife

- Sampling technique somewhat similar to Bootstrap

| | | Sample Size | |
|---|---|---|---|
| | | Subsample | Full Sample |
| Sampling Method | Without Replacement | Jackknife | Randomization Test |
| | With Replacement | | Bootstrap |

# Recall the Bootstrap

- The bootstrap uses sampling with replacement to form the training set.
  - Sample a dataset of $n$ instances $n$ times with replacement to form a new dataset of $n$ instances.
  - Use this data as the training set.
  - Use the instances from the original dataset that don't occur in the new training set for testing.

# The Jackknife

- Somewhat similar to bootstrap
- For single-elimination jackknife:
  - Create $n$ samples of size $n-1$
  - The $i^{th}$ instance is eliminated in the $i^{th}$ sample
  - Compute the mean (or wanted quantity) of each sample
- Can be used to estimate/reduce bias

# The Jackknife

- Estimating a parameter $\theta$:

$$\bar{\theta}_{\text{Jack}} = \frac{1}{n}\sum_{i=1}^{n}(\bar{\theta}_i)$$

- Estimating variance:

$$Var(\theta) = \sigma^2 = \frac{n-1}{n}\sum_{i=1}^{n}\left(\bar{\theta}_i - \bar{\theta}_{\text{Jack}}\right)^2$$

# The Jackknife

- Estimating and correcting bias:

$$\bar{\theta}_{\text{BiasCorrected}} = N\bar{\theta} - (N-1)\bar{\theta}_{\text{Jack}}$$

  – This reduces bias from $O(N^{-1})$ to $O(N^{-2})$