Where we are

- Supervised learning and classification problems
  - Predict a class label based on other attributes
- Rules
  - To start, we guessed simple rules that might explain the data
  - But the relationships are complex, so we need to automate
  - The 1-rule algorithm
  - A sequential cover algorithm for sets of rules with complex conditions
  - But: Sets of rules are hard to interpret
- Decision trees
  - Each path from the root is a tree; easy to interpret
  - Use entropy to choose best attribute at each node
  - Extensions for numeric attributes
  - *But: Decision Trees are prone to* ***overfitting***

Overfitting

What if the knowledge and data we have are not sufficient to completely determine the correct classifier? Then we run the risk of just hallucinating a classifier (or parts of it) that is not grounded in reality, and is simply encoding random quirks in the data.

This problem is called *overfitting*, and is the bugbear of machine learning. When your learner outputs a classifier that is 100% accurate on the training data but only 50% accurate on test data, when in fact it could have output one that is 75% accurate on both, it has overfit.

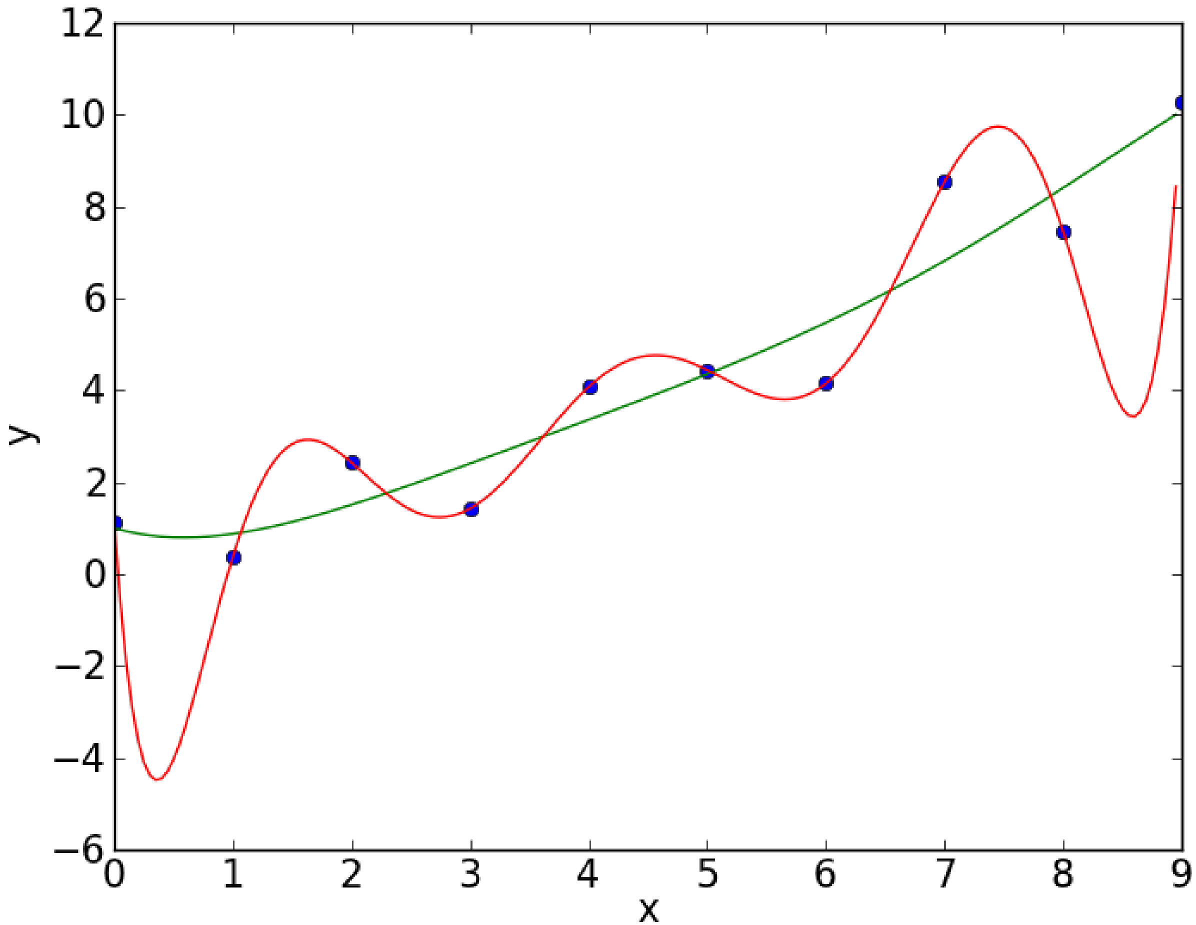
Pedro Domingos, A Few Useful Things to Know About Machine Learning, CACM 55(10), 2012

Overfitting

Low error on training data
and high error on test data

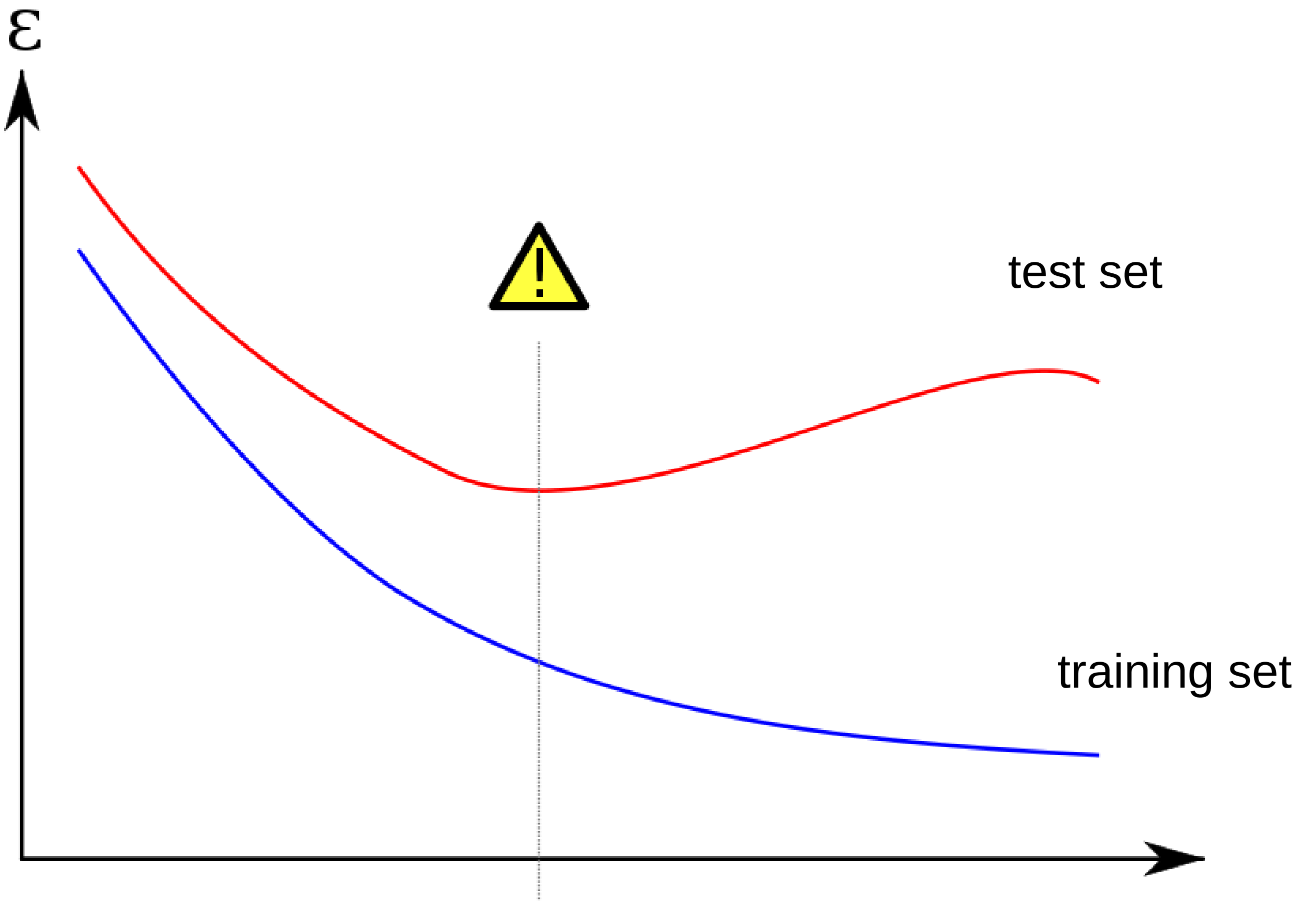
Overfitting

Perhaps not the most useful intuition



Overfitting

A better image to remember



Overfitting.png by Dake.

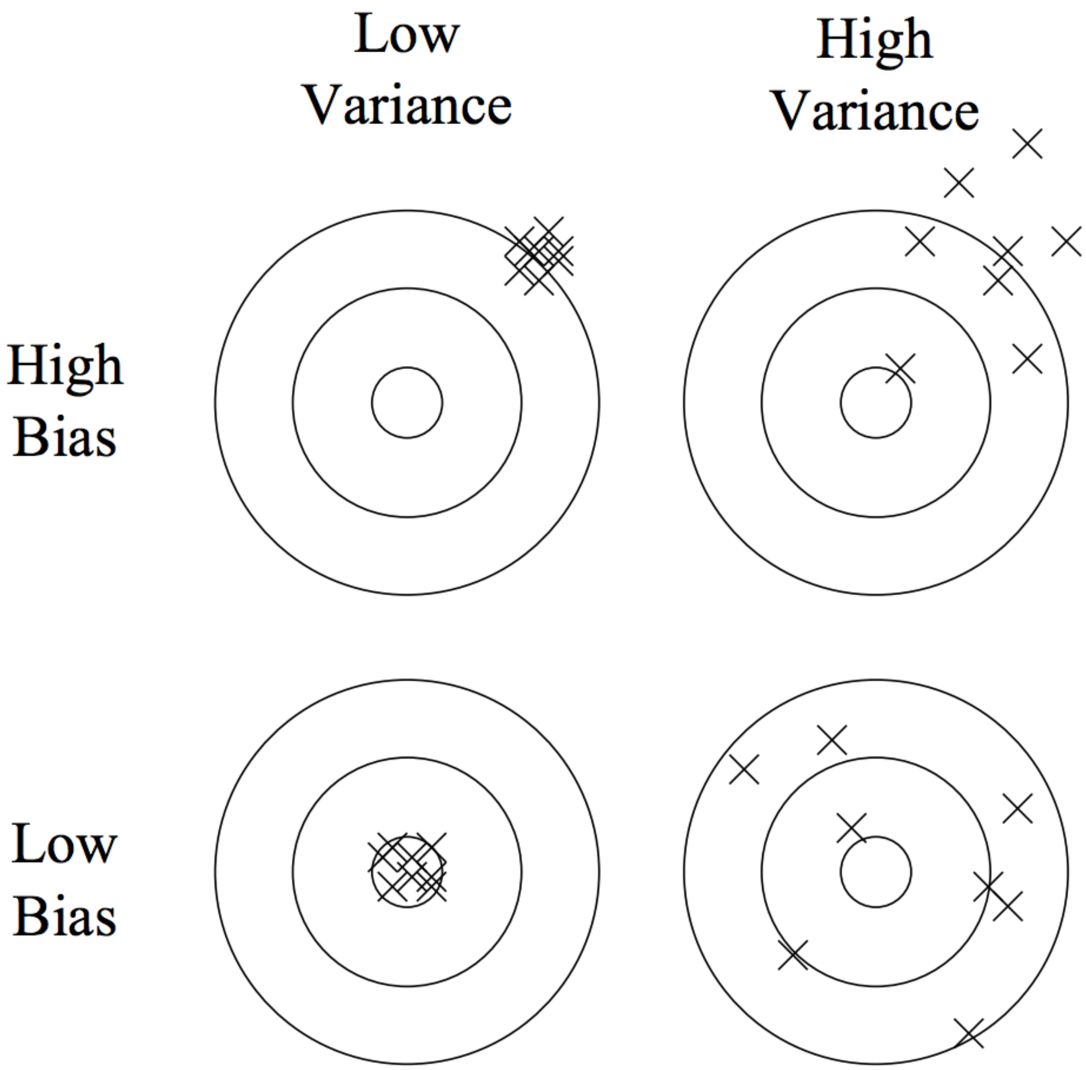
Is the model able to *generalize*? Can it deal with unseen data, or does it overfit the data? Test on hold-out data:

- split data to be modeled in training and test set
- train the model on training set
- evaluate the model on the training set
- evaluate the model on the test set
- difference between the fit on training data and test data measures the model's ability to *generalize*

slide src: Frank Keller

Underfitting:
- High Bias
- Low Variance

Overfitting:
- Low Bias
- High Variance



src: Domingo 2012