Introduction to Machine Learning

Wenhu Chen

Lecture 5

Outline

Learning Goals

- Identify reasons for building an agent that can learn.
- Describe different types of learning.
- Define supervised learning, classification, and regression.
- Define bias, variance, and describe the trade-off between them.
- Understand unsupservised learning and reinforcement learning
- Describe how to prevent overfitting by performing cross validation.

Applications

- Medical diagnosis
- Spam filtering
- Facial recognition
- Speech understanding
- Handwriting recognition

Agents that learn

Learning is the ability of an agent to improve its performance on future tasks based on experience.

$\rightarrow$ The agent needs to remember its past in a way that is useful for its future.

We want an agent to

- Do more $\rightarrow$ The range of behaviours is expanded.
- Do things better $\rightarrow$ The accuracy on tasks is improved.
- Do things faster $\rightarrow$ The speed is improved.

Why would we want an agent to learn?

- Cannot anticipate all possible situations
- Cannot anticipate all changes over time
- No idea how to program a solution

The Learning Architecture

- Problem/Task

  → The behaviour or task that is being improved.

- Experiences/Data

  → The experiences that are being used to improve performance in the task.

- Background knowledge/Bias

- Measure of improvement

  → How can the improvement be measured? e.g. increasing accuracy in prediction, new skills that were not present initially, improved speed.

Types of learning problems

- **Supervised learning:**

  Given input features, target features, and training examples, predict the value of the target features for new examples given their values on the input features.

- **Unsupervised learning:**

  Learning classifications when the examples do not have targets defined.

  E.g. clustering, dimensionality reduction

- **Reinforcement Learning:**

  Learning what to do based on rewards and punishments.

Q: Supervised or Unsupervised Learning

Q #1: We are given information on a user's credit card transactions. We would like to detect whether some of the transactions are fraudulent by finding some transactions that are different from the other transactions. We have no information on whether any particular transaction is fraudulent or not.

Is this a supervised or unsupervised learning problem?

(A) Supervised learning

(B) Unsupervised learning

Q: Supervised or Unsupervised Learning

Q #1: We are given information on a user's credit card transactions. We would like to detect whether some of the transactions are fraudulent by finding some transactions that are different from the other transactions. We have no information on whether any particular transaction is fraudulent or not.

Is this a supervised or unsupervised learning problem?

(A) Supervised learning

(B) Unsupervised learning

→ (B) is correct. We do not have values of the target features.

Q: Supervised or Unsupervised Learning

Q #2: You are given historical data on the weather condition (sunny, cloudy, rain, or snow) on a particular day of the year. You want to predict the weather condition on this day next year.

Is this a supervised or unsupervised learning problem?

(A) Supervised learning

(B) Unsupervised learning

Q: Supervised or Unsupervised Learning

Q #2: You are given historical data on the weather condition (sunny, cloudy, rain, or snow) on a particular day of the year. You want to predict the weather condition on this day next year.

Is this a supervised or unsupervised learning problem?

(A) Supervised learning

(B) Unsupervised learning

$\rightarrow$ (A) is correct. We have values of the target feature (i.e. weather condition).

Two types of supervised learning problems

- **Classification:** target features are discrete.

  $\rightarrow$ E.g. Predicting whether an image contains a dog or a cat

- **Regression:** target features are continuous.

  $\rightarrow$ E.g. Predicting tomorrow's temperature

Q: Classification or regression

Q #3: Is the following problem an instance of classification or regression?

You are given historical data on the weather condition (sunny, cloudy, rain, or snow) on a particular day of the year. You want to predict the weather condition of this day next year.

(A) Classification

(B) Regression

(C) This is not supervised learning.

Q: Classification or regression

Q #3: Is the following problem an instance of classification or regression?

You are given historical data on the weather condition (sunny, cloudy, rain, or snow) on a particular day of the year. You want to predict the weather condition of this day next year.

(A) Classification

(B) Regression

(C) This is not supervised learning.

$\rightarrow$ A) is correct. We are predicting the value of a discrete variable.

Q: Classification or regression

Q #4: Is the following problem classification or regression?

You are given historical data on the price of a house at several points in time. You want to predict the price of this house next month.

(A) Classification

(B) Regression

(C) This is not supervised learning.

Q: Classification or regression

Q #4: Is the following problem classification or regression?

You are given historical data on the price of a house at several points in time. You want to predict the price of this house next month.

(A) Classification

(B) Regression

(C) This is not supervised learning.

$\rightarrow$ B) is correct. We are predicting the value of a continuous variable.

Supervised Learning

- Given training examples of the form $(x, f(x))$

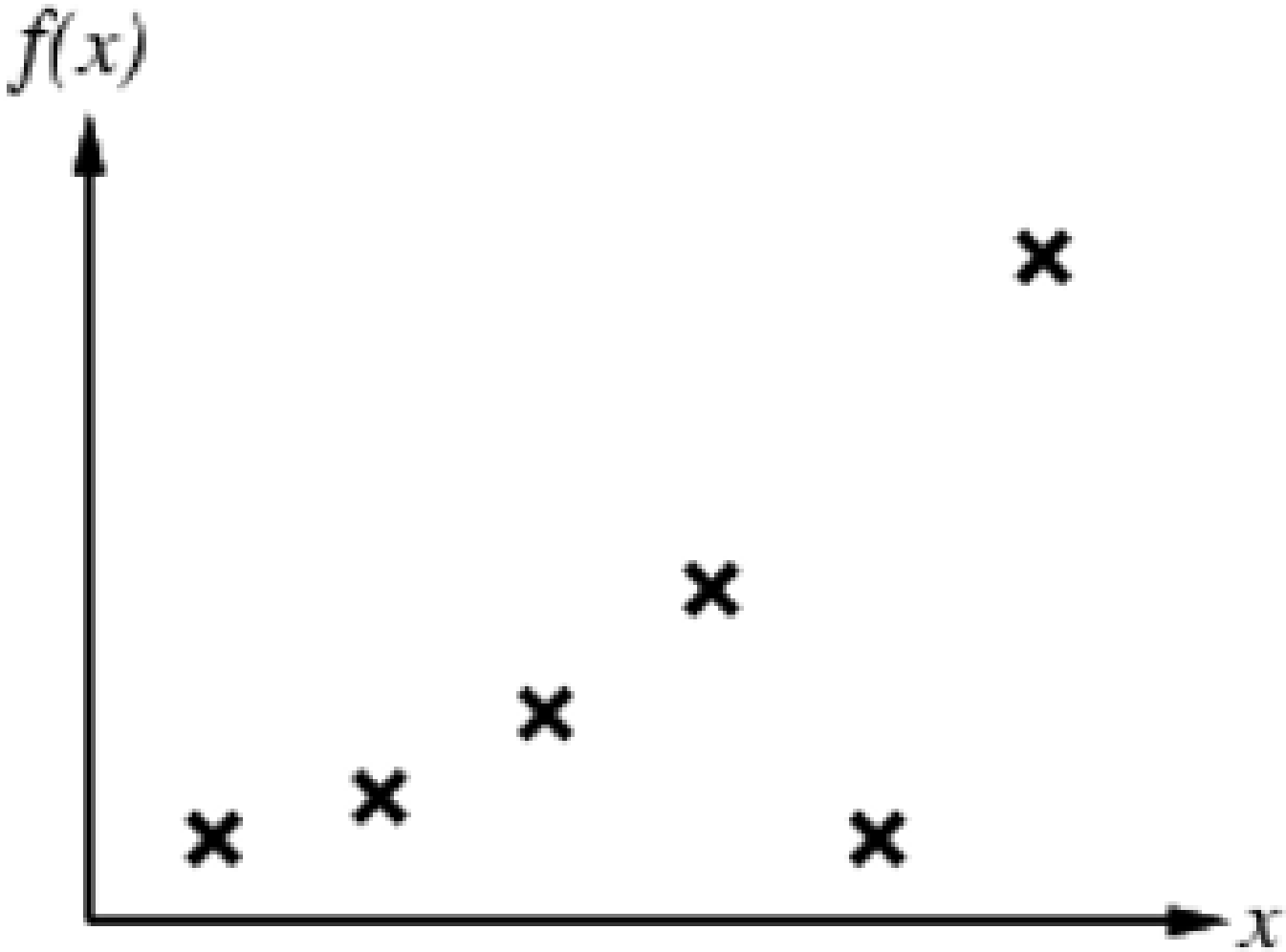
  $\rightarrow$ We assume that f exists. We don't have f. We never observe f.

- Return a function h (a.k.a a hypothesis) that approximates the true function f.

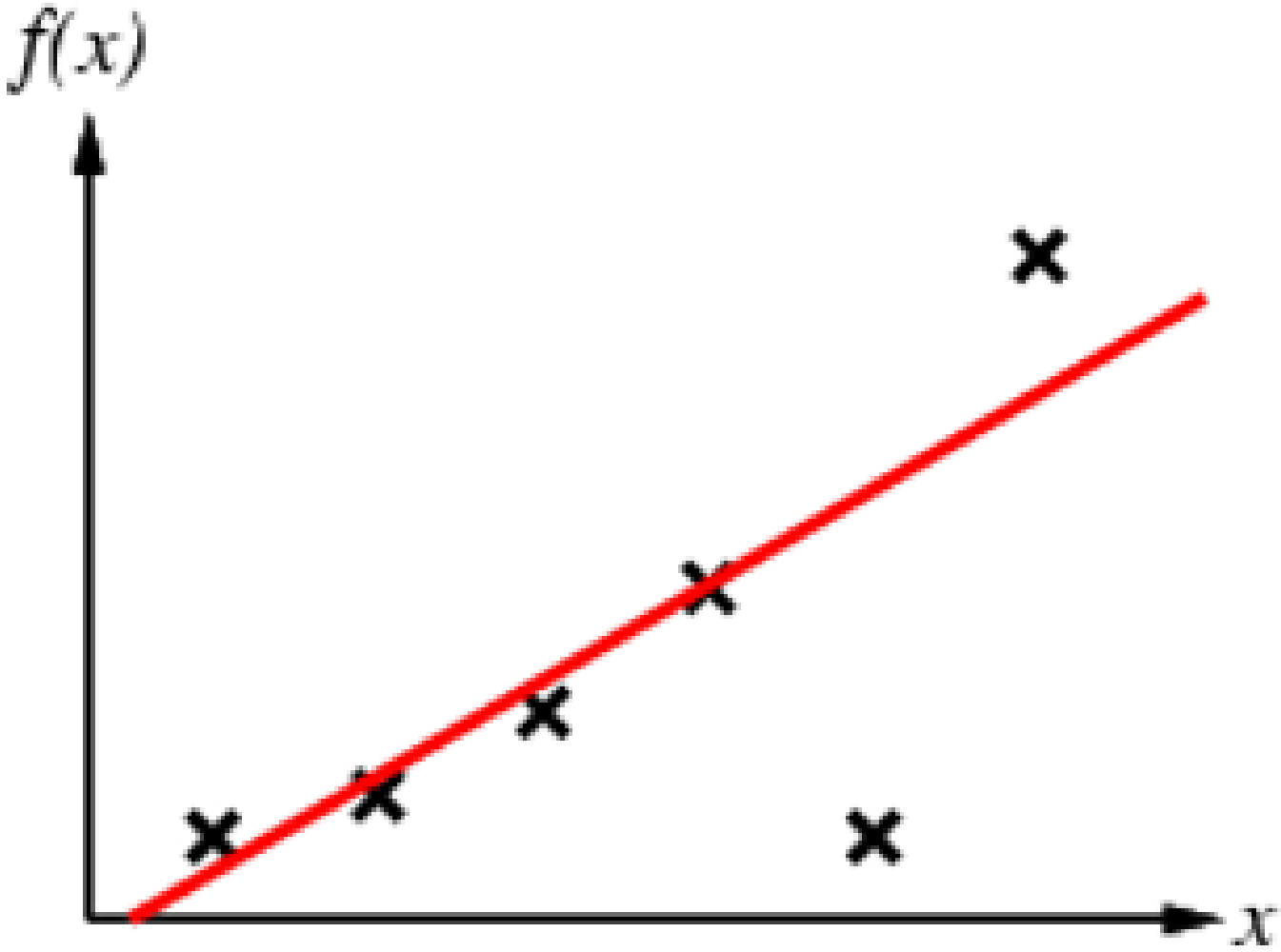
Learning as a search problem

- Given a hypothesis space, learning is a search problem.
- Search space is prohibitively large for systematic search.
- ML techniques are often some forms of local search.

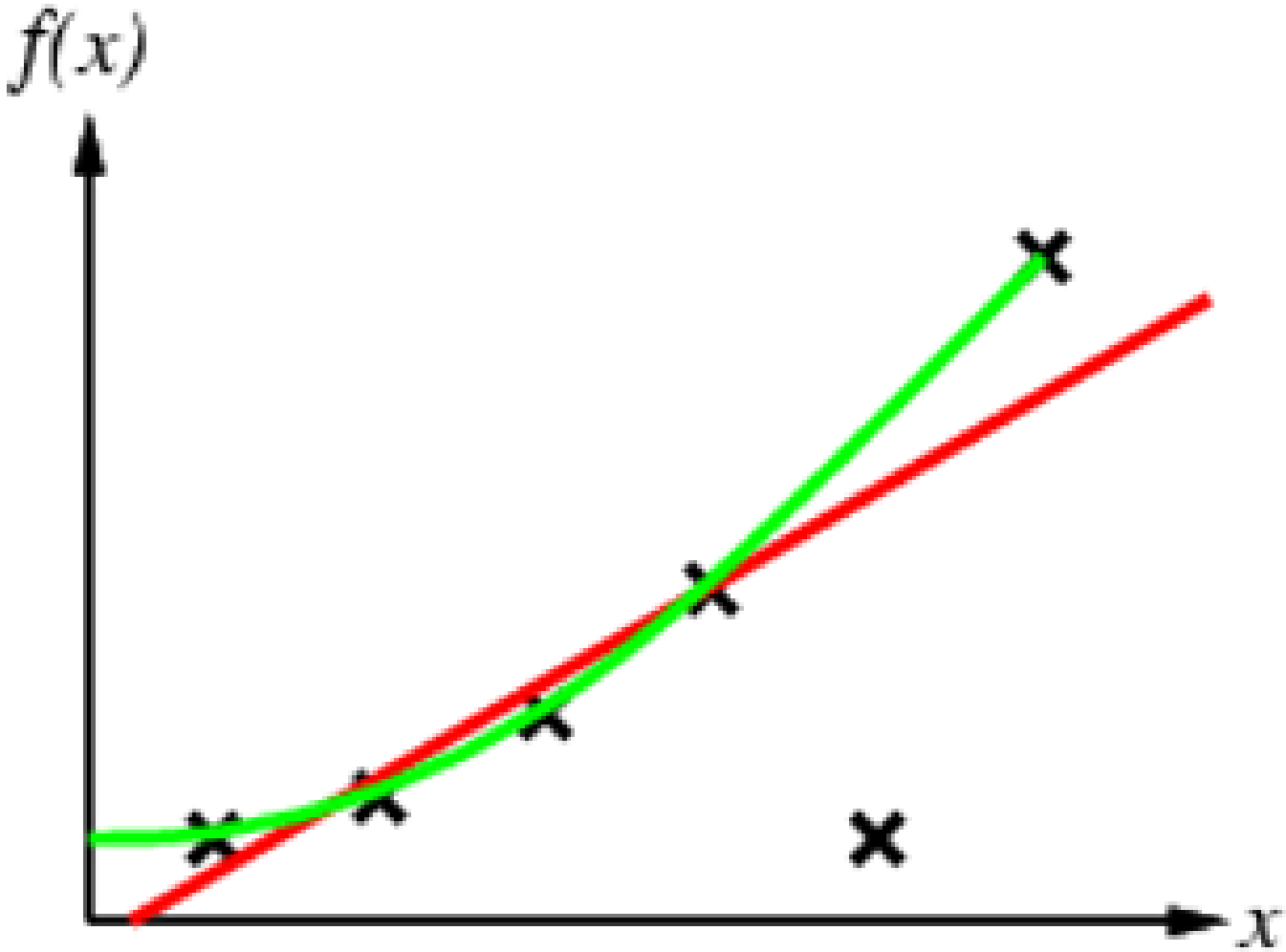
Example: A prediction task. Fit a polynomial.



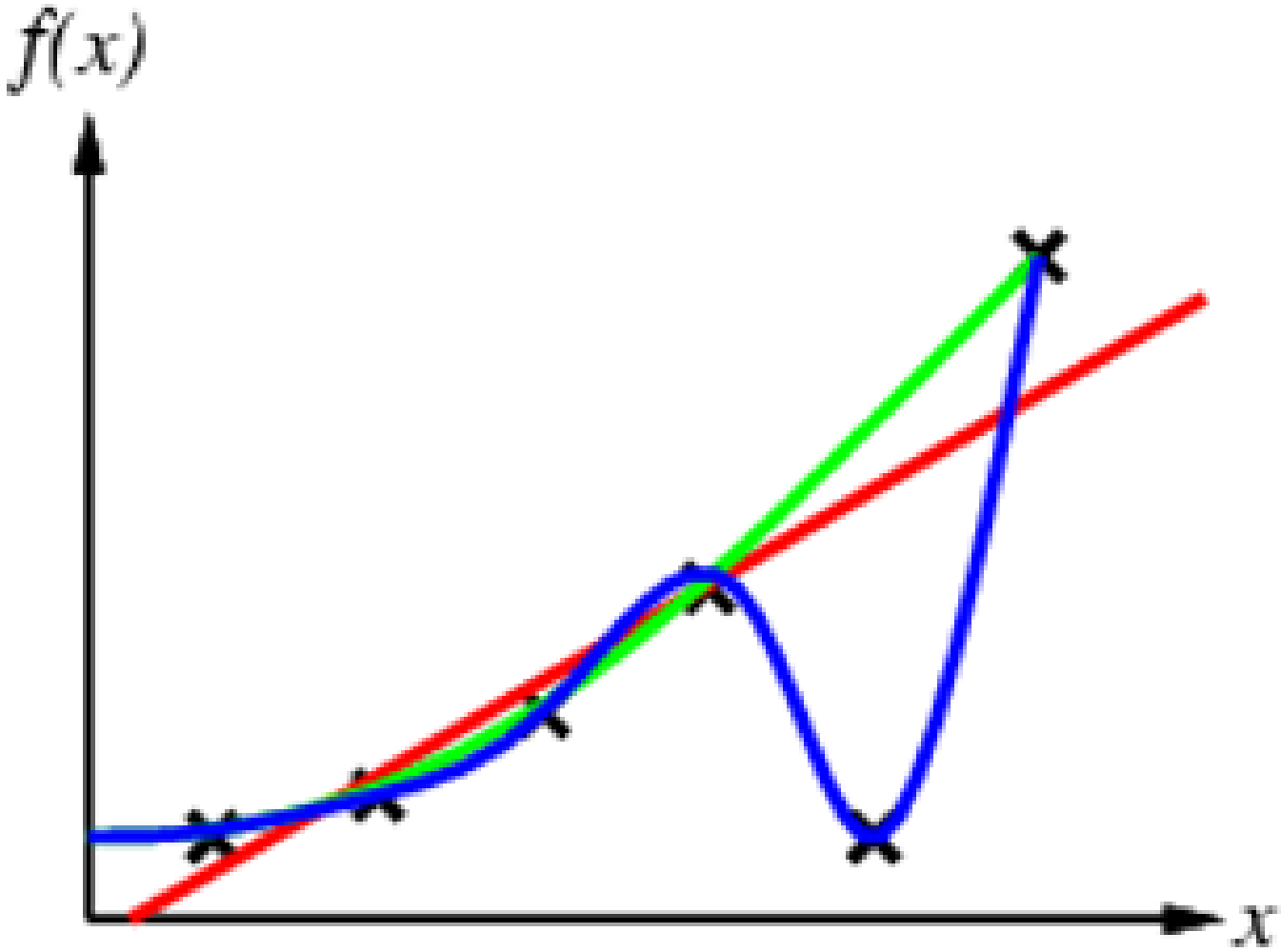
Example: A prediction task. Fit a polynomial.



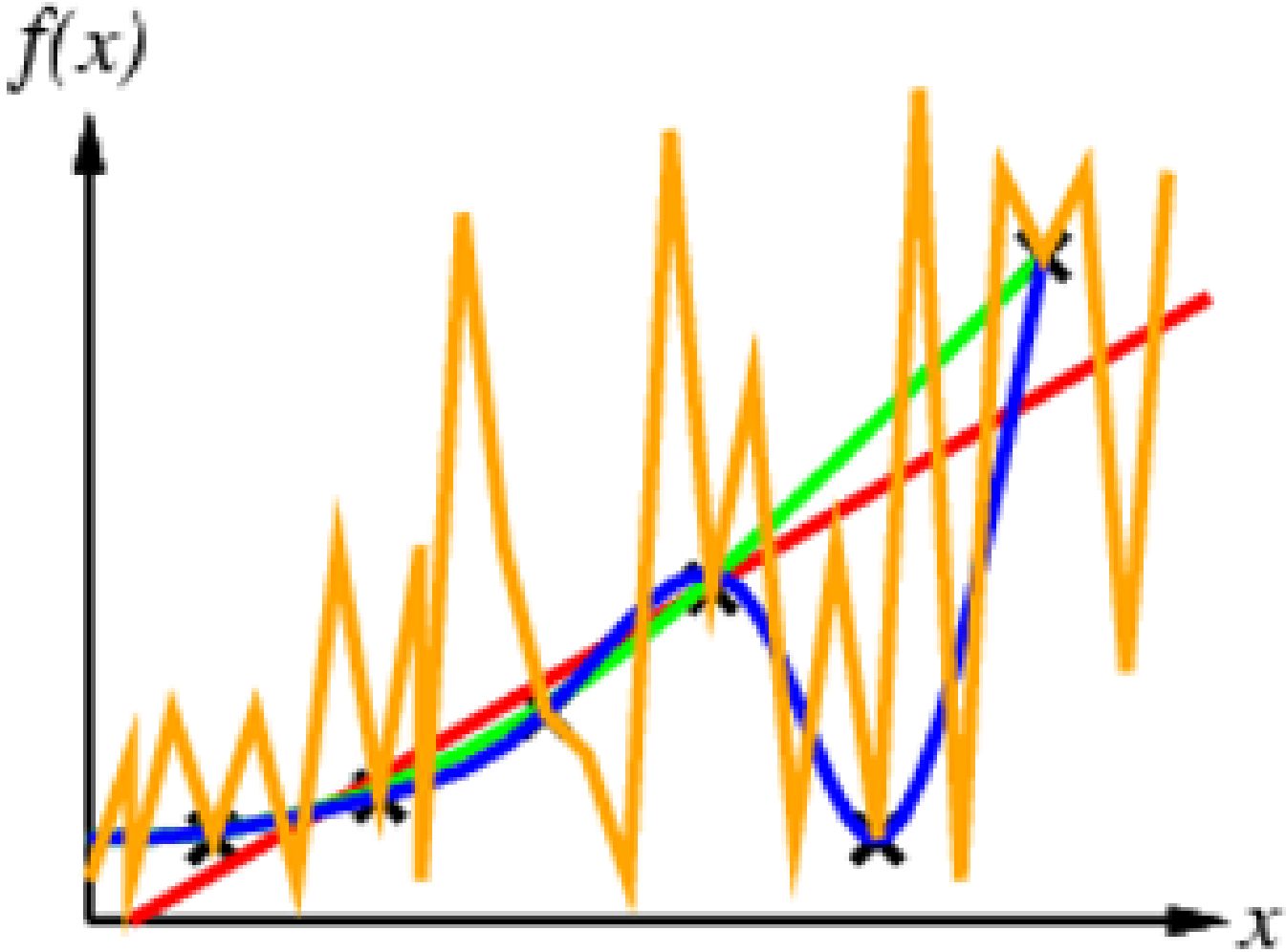
Example: A prediction task. Fit a polynomial.



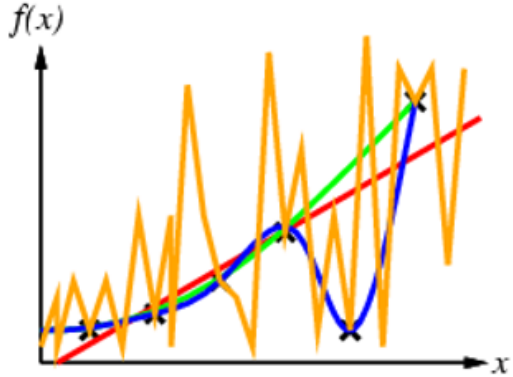
Example: A prediction task. Fit a polynomial.



Example: A prediction task. Fit a polynomial.



Example: A prediction task. Fit a polynomial.



- Which function is the correct one?
- If some data points are outliers, or the data is noisy, then the simpler curves are better.
- If we are predicting stock market prices, perhaps the more complex curve is better.
- There is no perfect answer. All curves can be justified as the correct one from some perspective.
- *No free lunch theorem:* In order to learn something useful, we have to make some assumptions — have an inductive bias.
- Assumptions can include: Do I have any outliers? Does the curve follow a particular parametric form?

Generalization

- Goal of ML is to find a hypothesis that can predict unseen examples correctly.

  $\rightarrow$ Goal is not to predict the data we already have correctly. This makes ML difficult but exciting.

- How do we choose a hypothesis that generalizes well?
  - *Ockham's razor*

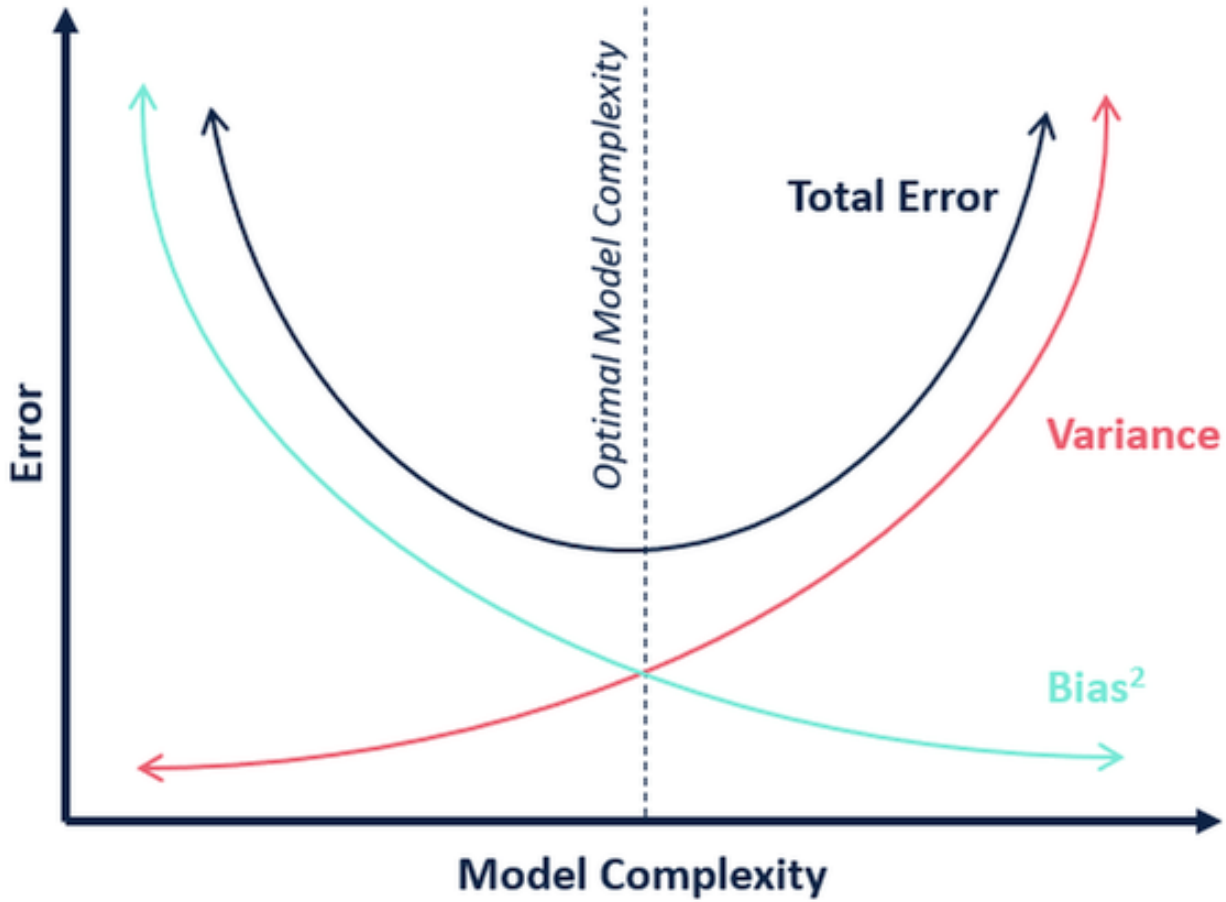
    $\rightarrow$ prefer the simplest hypothesis consistent with the data.
  - *Cross-validation*

    $\rightarrow$ a more principled approach to choose a hypothesis.

- A trade-off between
  - complex hypotheses that fit the training data well
  - simpler hypotheses that may generalize better

Bias-Variance Trade-off

How well does the hypothesis fit the data as the hypothesis becomes more complex?



Bias-Variance Trade-off

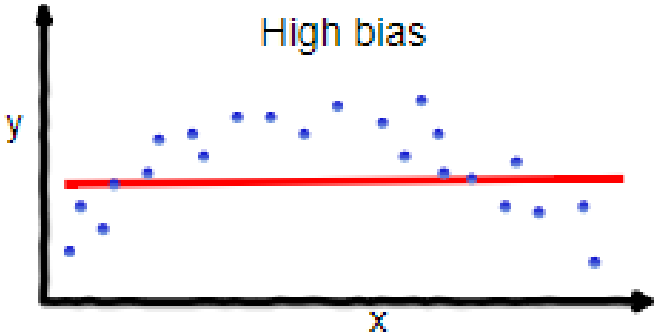
- **Bias:** If I have infinite data, how well can I fit the data with my learned hypothesis?

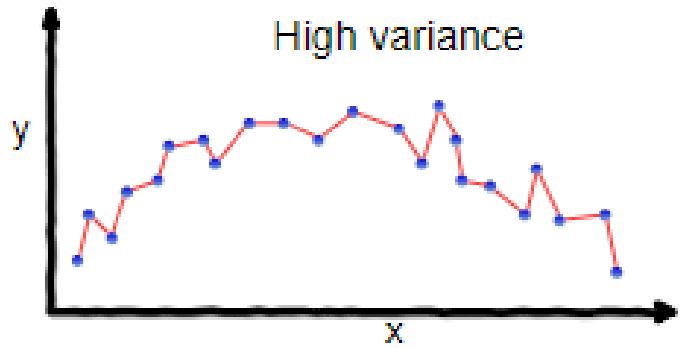
  A hypothesis with high bias makes strong assumptions, too simplistic, has few degrees of freedom, does not fit the training data well.

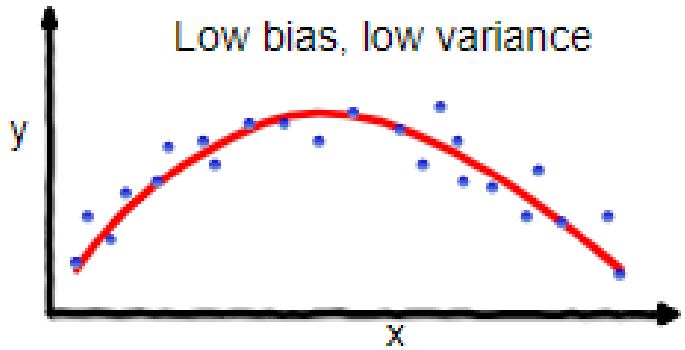
- **Variance:** How much does the learned hypothesis vary given different training data?

  A hypothesis with high variance has a lot of degrees of freedom, is very flexible, and fits the training data well. Whenever the training data changes, the hypothesis changes a lot.

Bias-Variance Trade-off







Bias-Variance Equation

Suppose that we have a training set consisting of set of points $x_1, \cdots, x_n$, and real values y_i associated with each point. We assume that the data is generated by a function $f(x)$ such as $y = f(x) + \epsilon$, where the noise is Gaussian.

We want to find a function $\hat{f}(x; D)$ that approximates the true function $f(x)$ as well as possible, by means of some learning algorithm based on training dataset $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Finding an $\hat{f}$ that generalizes to points outside of the training set can be done with any of the countless algorithms.

Bias-Variance Equation

It turns out that whichever function $\hat{f}$ we select, we can decompose its expected error on an unseen sample x as follows:

$$\mathbb{E}_{D,\epsilon}[(y-\hat{f}(x;D))^2] = (Bias_D[\hat{f}(x;D)])^2 + Var[\hat{f}(x;D)] + \sigma^2 \quad (1)$$

where

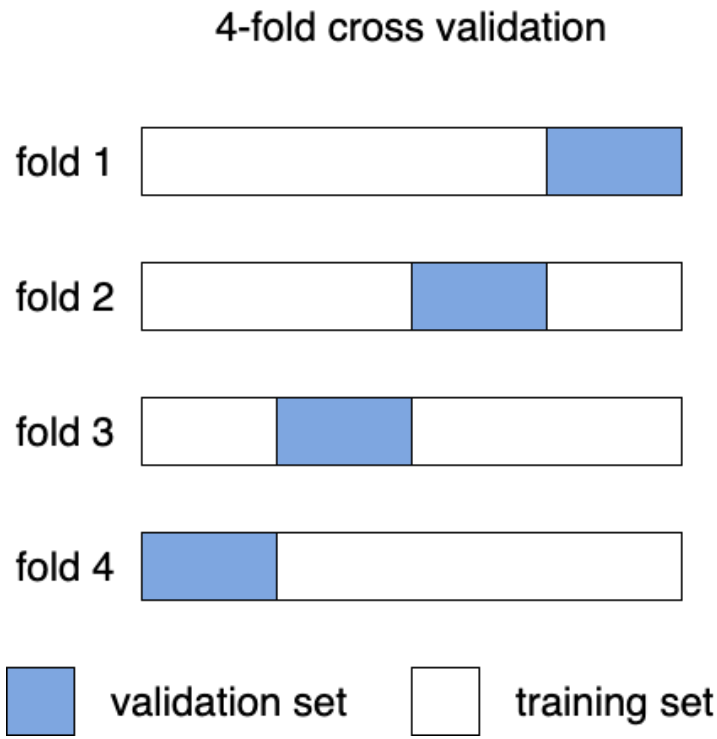
$$Bias_D[\hat{f}(x;D)] = \mathbb{E}_D[\hat{f}(x;D)] - f(x) \quad (2)$$

$$Variance_D[\hat{f}(x;D)] = \mathbb{E}_D[(\mathbb{E}_D(\hat{f}(x;D)) - \hat{f}(x;D))^2] \quad (3)$$

The total error is the bias + variance + irreducible noise.

Cross-validation

How do we find a hypothesis that has low bias and low variance?
Use cross validation.



→ Use part of the training data as a surrogate for test data (called validation data).

Use validation data to choose the hypothesis.

K-fold Cross Validation

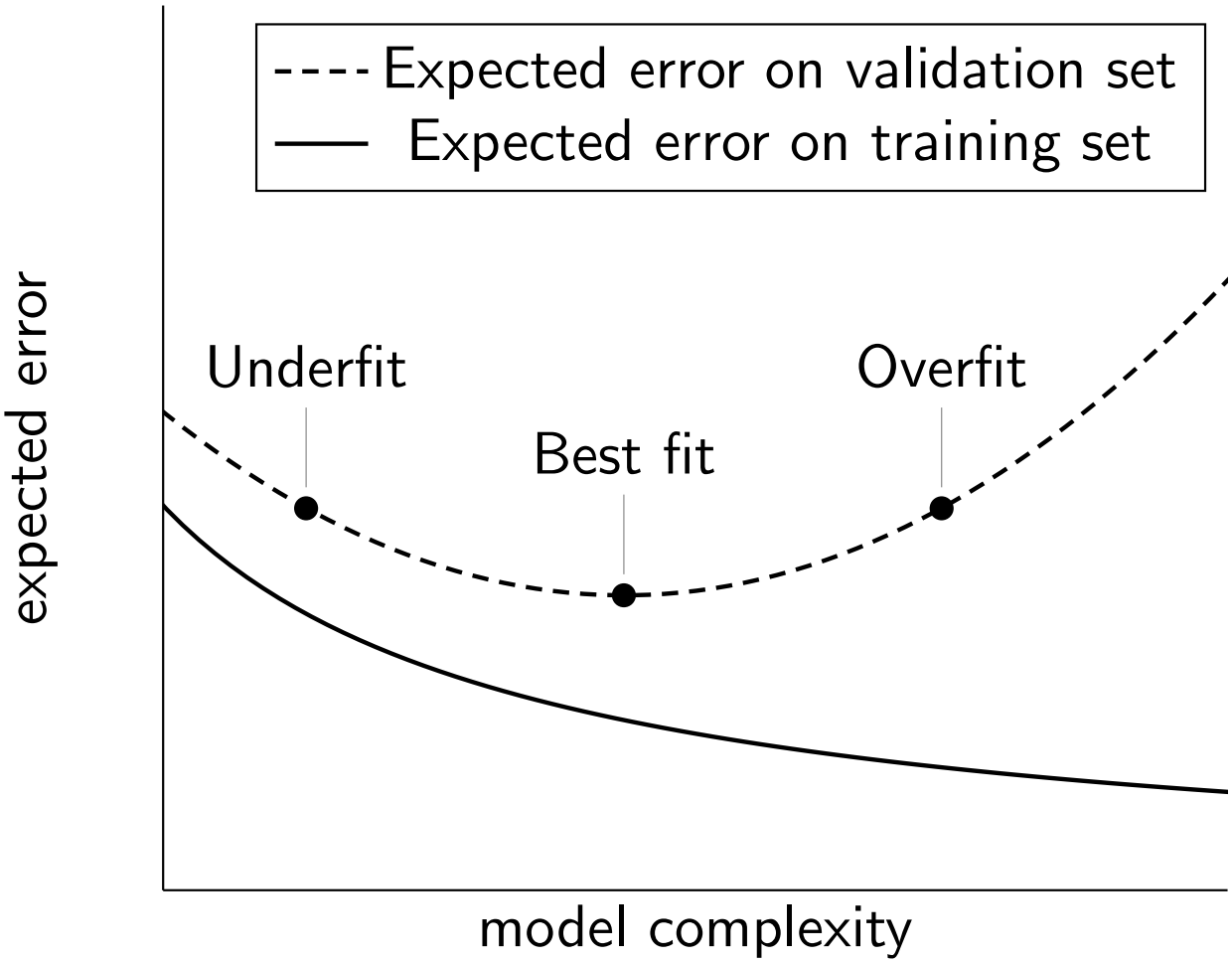
1. Break training data into K equally sized partitions.
2. Train a learning algorithm on $K-1$ partitions (training set).
3. Test on the remaining 1 partition (validation set).
4. Do this K times, each time testing on a different partition.
5. Calculate the average error on the K validation sets.

After cross validation

After running cross validation, you can:

- Select one of the K trained hypotheses as your final hypothesis.
- Train a new hypothesis on all of the data, using parameters selected by cross validation.

Overfitting



Q: Which hypothesis is prone to overfitting?

Q #5: Suppose that we are considering a simple hypothesis (a straight line) and a complex hypothesis (a 4^{th} degree polynomial). Which of the two is more likely to overfit the training data?

(A) The simple hypothesis

(B) The complex hypothesis

(C) I don't know

Q: Which hypothesis is prone to overfitting?

Q #5: Suppose that we are considering a simple hypothesis (a straight line) and a complex hypothesis (a 4^{th} degree polynomial). Which of the two is more likely to overfit the training data?

(A) The simple hypothesis

(B) The complex hypothesis

(C) I don't know

$\rightarrow$ (B) is the correct answer.

Learning Problem

- Let's start with a simple supervised learning classification problem. Let us say that we want to classify spam emails, probably the most often used example in machine learning.
- We denote the domain space with X and the label space with Y, we also need a function for mapping the domain set space to the label set space, $f : X \to Y$, this is just a formal definition of a learning task.

Learning Problem

- We need a model that is going to make our predictions. We denote our hypothesis as h.
- The hypothesis, in this case, is nothing else than a function that takes input from our domain X and produces a label 0 or 1, i.e. a function h: $X \rightarrow Y$.
- Looking at it from a probabilistic perspective we say that we sample S from the domain set X, with D being the distribution over X.

We can define the true error:

$$L_{D,f}(h) = P_{x \in D}[h(x) \neq f(x)] = D(\{x : h(x) \neq f(x)\}) \qquad (4)$$

Learning Problem

Since we only have access to S, a subset of domain D. We can only compute the empirical error:

$$L_S(h) = \frac{\{i \in [m] : h(x_i) \neq y_i\}}{m} \tag{5}$$

m denotes the number of training examples. You can see from the equation that we effectively define the empirical error as the fraction of misclassified examples in the set S.

We want to generalize based on S, also called inductive learning. This error is also called the risk, hence the term risk in empirical risk minimization.

Empirical Loss Function

Given a training dataset containing examples $S = \{X_i, Y_i\}_1^m$, one realization of empirical risk minimization is:

$$\hat{h} = argmin_{h:X \to Y} \frac{1}{M} \sum_{i=1}^{M} l(h(X_i), Y_i) \tag{6}$$

We hope to minimize the empirical risk/training error on a given dataset, which is likely to minimize the true error.

Empirical Loss Function

Given a training dataset containing examples $S = \{X_i, Y_i\}_1^m$, one realization of empirical risk minimization is:

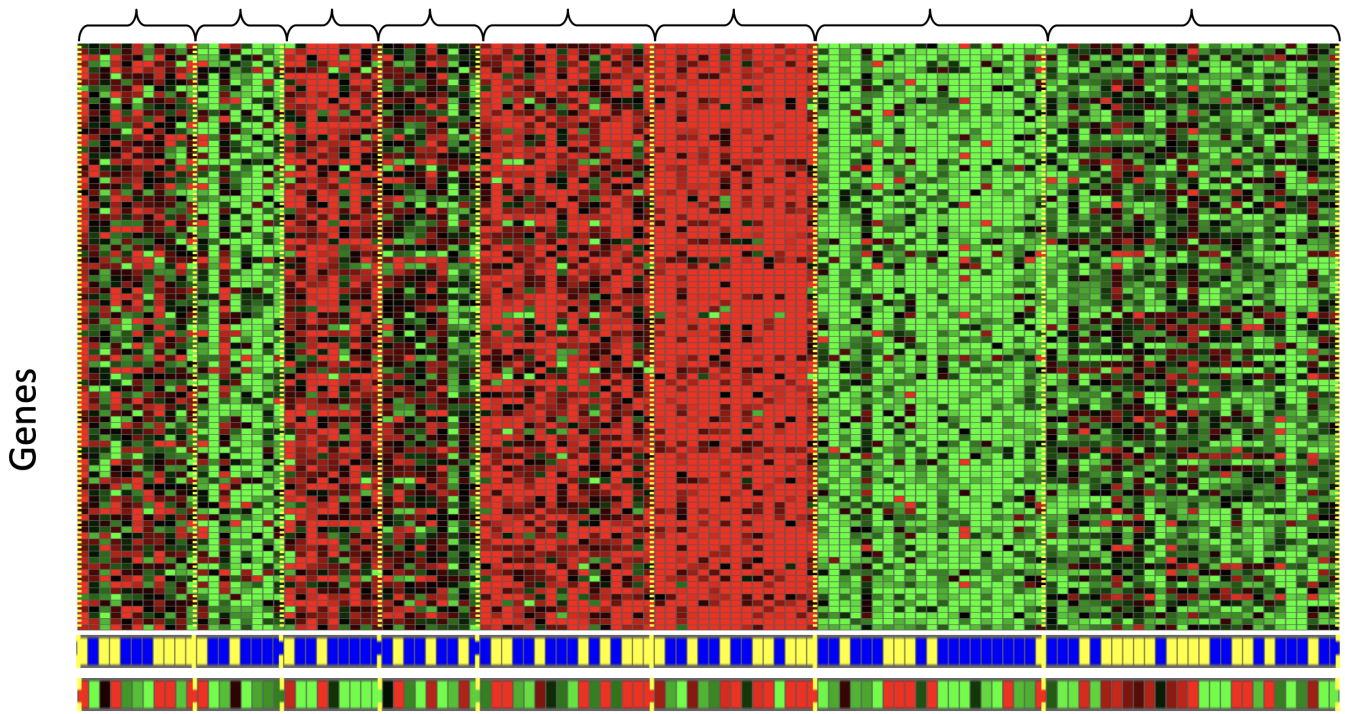
$$\hat{h} = argmin_{h:X \to Y} \frac{1}{M} \sum_{i=1}^{M} l(h(X_i), Y_i) \tag{7}$$

Important Factors for EMR:

- The size of the training dataset S. The larger, the better.
- The hypothesis space is important. Your need to trade off between bias and variance.
- The loss function - It can imply certain bias.

Learning Problem

Group individuals by genetic similarity.



Supervised & Unsupservised Learning

- Supervised Learning: discover patterns in the data that relate data attributes with target class
  - These patterns are then utilized to predict the values of the target class in future instances.
- Unsupervised Learning: the data have no target class
  - We want to explore the data to find intrinsic structures in them. These structures can help us understand the data.
- Due to historical reasons, clustering is often considered synonymous with unsupservised learning.

Supervised & Unsupervised Learning

- Clustering is a technique for finding similarity groups in the data, called clusters.
  - It groups data instances that are similar to each other in one cluster and data instances that are very different from each other into different clusters.
- Clustering is often called an unsupervised learning task as no class values denoting an a priori grouping of the data instances are given.

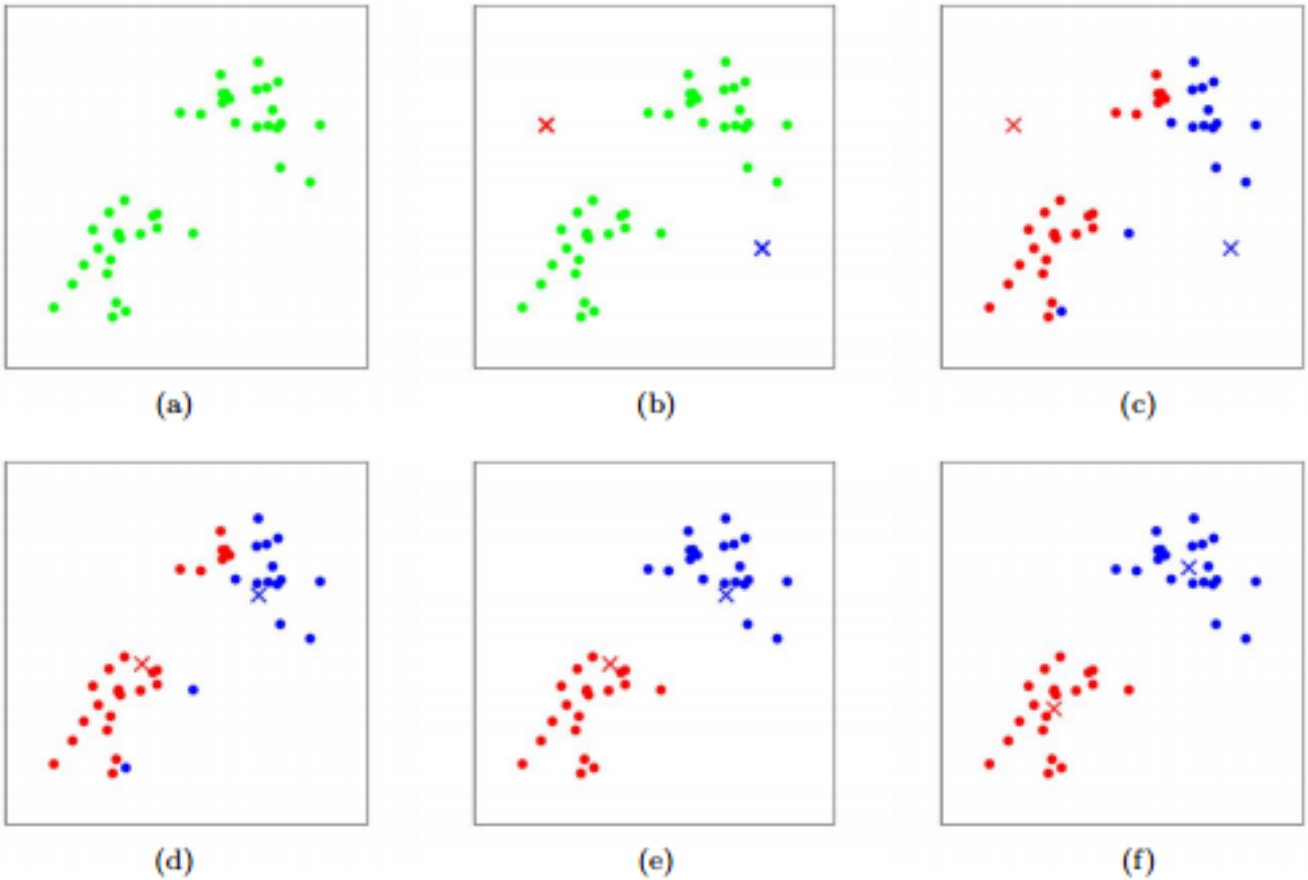
What is clustering?

- Example 1: group people of similar sizes together to make 'small', 'medium' and 'large' T-shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit well
- Example 2: In marketing, segment customers according to their similarities
  - Customized recommendation for each group of customers
- Example 3: Given a collection of text documents, we want to organize them according to the similarities
  - Topic Modeling

K-Means algorithm

- Given k, the k-means algorithm works as follows:
  - Randomly select k data points as initial centroids.
  - Assign each data point to the closest centroid.
  - Re-compute the centroid using the curretn cluster memberships.
  - If convergence is not met, go to step 2).

K-Means algorithm



Independence Component Analysis

- Observed data $x_i(t)$ is modeled by using hidden variables $s_i(t)$:

$$x_i(t) = \sum_{j=1}^{m} a_{ij} s_j(t), i = 1...n \tag{8}$$

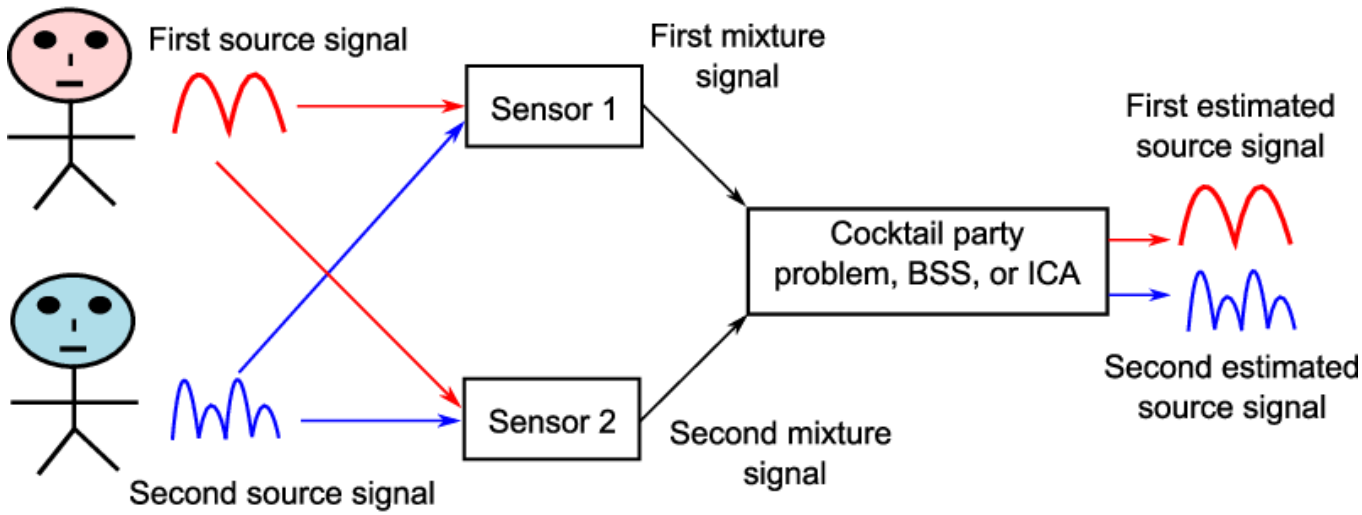
or as a matrix decomposition

$$X = AS \tag{9}$$

- Matrix of a_{ij} is constant parameter called 'mixing matrix'.
- Hidden random factors $s_i(t)$ are called 'independent components' or 'source signals'.
- Estimate both a_{ij} and $s_j(t)$, observing only $x_i(t)$.

Independence Component Analysis

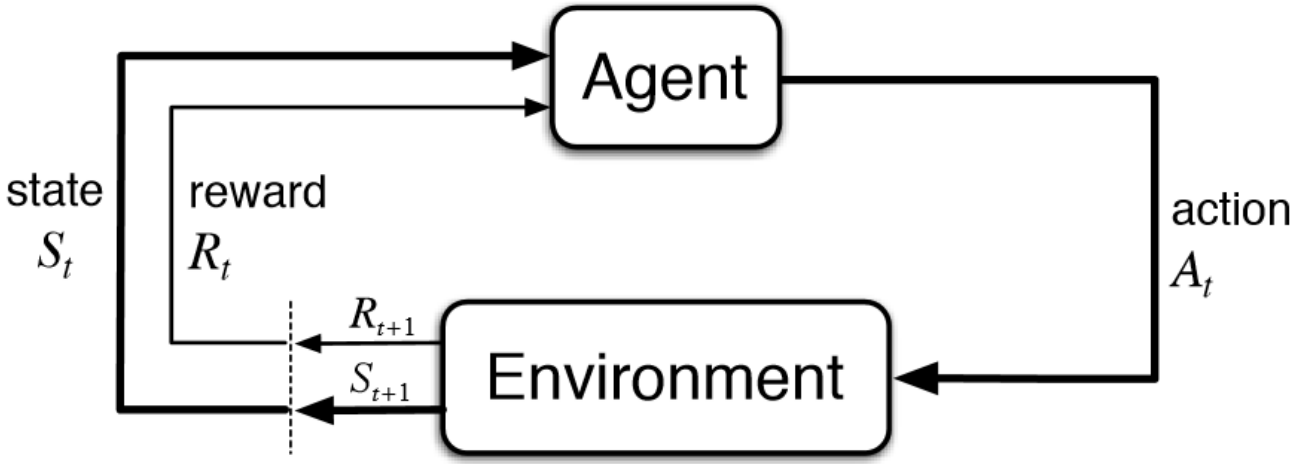
Disentangle different speakers in a cocktail party.



Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states to actions that tells you what to do in a given state.
  - We need to have an environment for the agent to interact with.
- Examples
  - Game Playing
  - Robot Control
  - Portfolio Management

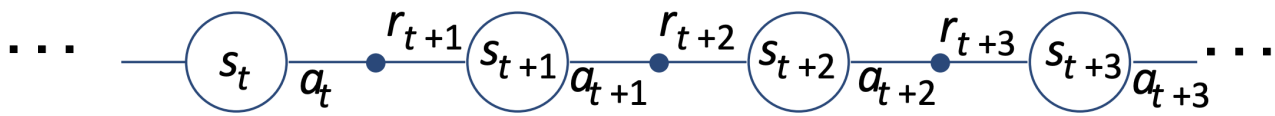
The Agent-Environment Interface



Agent and environment interact at discrete time steps:

- Agent observes state at step t: s_t
- produces action at step t: a_t
- gets resulting reward: r_{t+1}
- resulting in next state: s_{t+1}

The Goal of Reinforcement Learning



Given the trace of (s_t, a_t, r_t, s_{t+1}) quadruple. We want to learn a policy that maximizes the long-term reward:

$$argmax_\pi \mathbb{E}_{a_t \sim \pi(s_t)} G \tag{10}$$

G is the discounted reward:

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + ... \tag{11}$$

The algorithm must find a policy with maximum expected discounted return.

Revisiting Learning Goals

- Identify reasons for building an agent that can learn.
- Describe different types of learning.
- Define supervised learning, classification, and regression.
- Define bias, variance, and describe the trade-off between them.
- Understand unsupservised learning and reinforcement learning
- Describe how to prevent overfitting by performing cross validation.