CS 461
Artificial Intelligence

Dr. Hashim Yasin

Function Approximation

Function Approximation

- Given a bunch of examples of
  - input
  - output

- *Find a function which does a good job of expressing the relationship between them.*

- The problem of learning a function from examples, is complicated.

Function Approximation

- Simplest form: learn a function from examples

  f is the target function

  An example is a pair (x, $f(x)$)

"Polynomial Game": Learn polynomial from examples

x	$f(x)$
1	1
2	4
3	9
4	16
5	25

$f(x)$ = ?

$f(x) = x^2$

Function Approximation

- Simplest form: learn a function from examples
  - *f* is the target function
  - An example is a pair (*x*, *f(x)*)

"Polynomial Game": Learn polynomial from examples

x	*f(x)*
1	1
2	1
3	1
4	1
5	121

$f(x) = ?$

$f(x) = (x-4)(x-3)(x-2)(x-1)x+1$

Function Approximation

Problem Setting:

- Set of possible instances (instance space with fixed distribution D_X) X
- A corresponding target space Y
- Unknown target function

$$f: X \rightarrow Y$$

- Set of function hypotheses

$$H = \{h \mid h: X \rightarrow Y\}$$

Function Approximation

Input:

- Training examples $\{x_i, y_i\}$ of unknown target function

Output:

- Hypothesis $h \in H$ that ***best approximates*** the target function f

$$error_{D_X}(h) = E_{D_X}\left(error(f(x), h(x))\right)$$

the expected (average) classification error on instances drawn according to D_X should be **minimal**

Function Approximation

- There are following different categories for the function approximation:
    - Memory
    - Averaging
    - Generalization

Example

When to drive the car? It depends on,

- Temperature
- Expected precipitation
- Day of the week
- Whether need to shop on the way back home
- What are you wearing

Memory

Temp	Precip	Day	Shop	Cloths	
80	None	Sat	No	Casual	Walk
19	Snow	Mon	Yes	Casual	Drive
65	none	Tue	No	Casual	Walk

Memory

Temp	Precip	Day	Shop	Cloths	
80	None	Sat	No	Casual	Walk
19	Snow	Mon	Yes	Casual	Drive
65	none	Tue	No	Casual	Walk
19	Snow	Mon	Yes	Casual	??

Averaging

Temp	Precip	Day	Shop	Cloths	
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Drive
80	None	Sat	No	Casual	Drive
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	?

Averaging

Temp	Precip	Day	Shop	Cloths	
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Drive
80	None	Sat	No	Casual	Drive
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk
80	None	Sat	No	Casual	Walk

Generalization

Temp	Precip	Day	Shop	Cloths	
71	None	Fri	Yes	Casual	Drive
36	None	Sun	Yes	Casual	Walk
62	Rain	Weds	No	Casual	Walk
93	None	Mon	No	Casual	Drive
55	None	Sat	No	Formal	Drive
80	None	Sat	No	Casual	Walk
19	Snow	Mon	Yes	Casual	Drive
65	None	Tues	no	Casual	Walk

Generalization

Temp	Precip	Day	Shop	Cloths	
71	None	Fri	Yes	Casual	Drive
36	None	Sun	Yes	Casual	Walk
62	Rain	Weds	No	Casual	Walk
93	None	Mon	No	Casual	Drive
55	None	Sat	No	Formal	Drive
80	None	Sat	No	Casual	Walk
19	Snow	Mon	Yes	Casual	Drive
65	None	Tues	No	Casual	Walk
58	Rain	Mon	No	Casual	??

Generalization

- He's going to **walk** because it's raining today and the only other time it rained, he walked.
- He's going to **drive** because he has always driven on Mondays.
- He's going to **walk** because he only drives if he is wearing formal clothes, or if the temperature is above 90 or below 20.

The question of which one to choose is hard.

Learning

Learning

- An agent is **learning** if it improves its performance on future tasks after making observations about the world.
- **Learning** is the ability of an agent to improve its behavior based on experience.
- This could mean the following:
  - The **range of behaviors** is expanded;
    - the intelligent agent can do more.
  - The **accuracy level to perform tasks** is improved;
    - the intelligent agent can do things in a better way.
  - The efficiency in terms of **speed** is improved;
    - the intelligent agent can do things faster.

Learning

- Any component of an agent can be improved by learning from data.

- The improvements may depend on four major factors:
  - Which ***component*** is to be improved.
  - What ***prior knowledge*** the agent already has.
  - What ***representation*** is used for the data and the component.
  - What ***feedback*** is available to learn from.

Inductive Learning

- One of the most common kinds of learning is the acquisition of information with the goal of making predictions about the future.
- **David Hume** first framed the problem of **induction** as follows:
  - Piece of bread number 1 was nourishing when I ate it.
  - Piece of bread number 2 was nourishing when I ate it.
  - Piece of bread number 3 was nourishing when I ate it.
  - Piece of bread number 100 was nourishing when I ate it.
  - *Therefore*, all pieces of bread will be nourishing if I eat them.

Inductive Learning

- Simplest form: learn a function from examples

  f is the target function

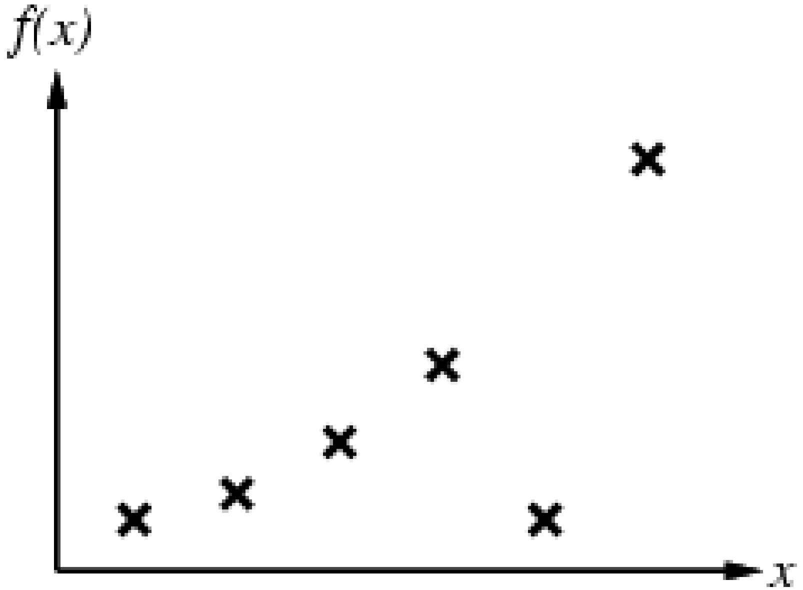
  An example is a pair $(x, f(x))$

- Problem: find a hypothesis h
  such that $h \approx f$
  given a training set of examples

- This is a highly simplified model of real learning:
  - Assumes examples are given

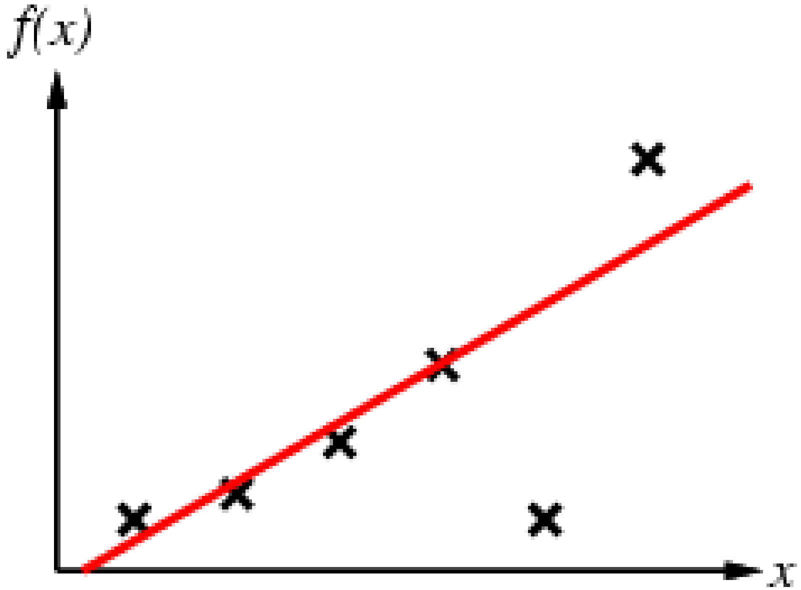
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



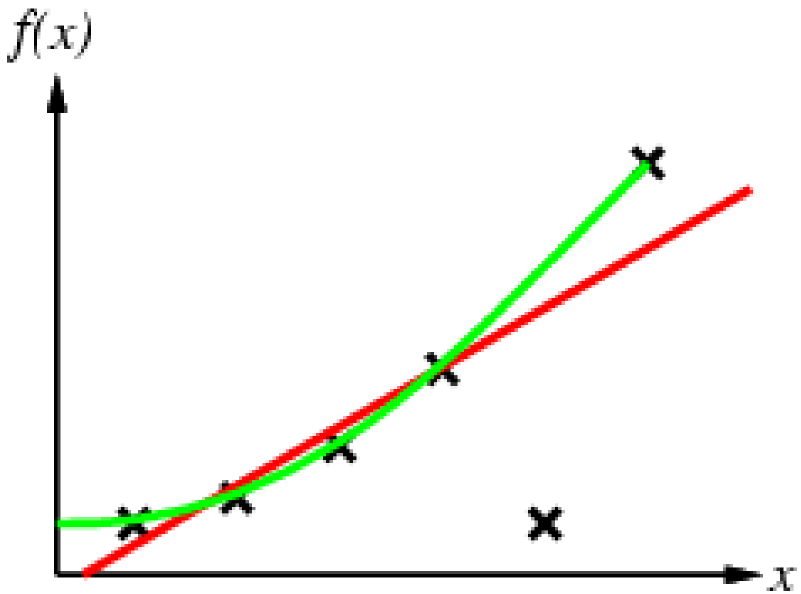
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



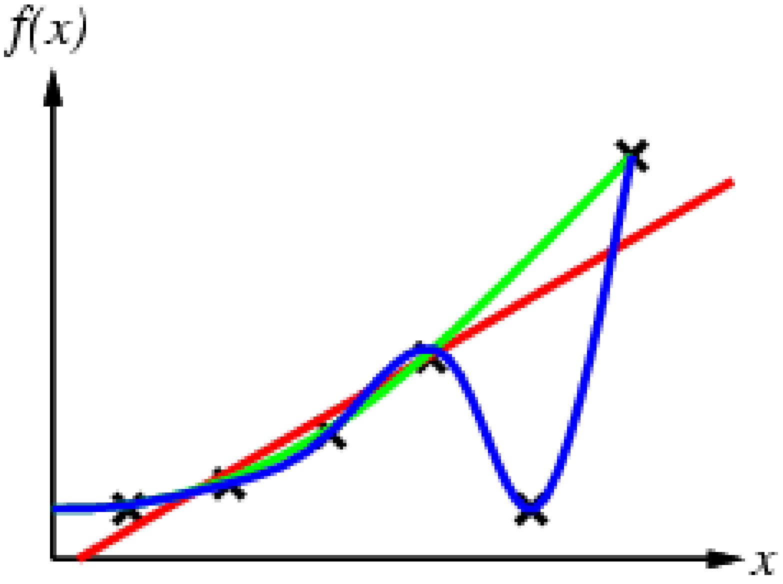
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



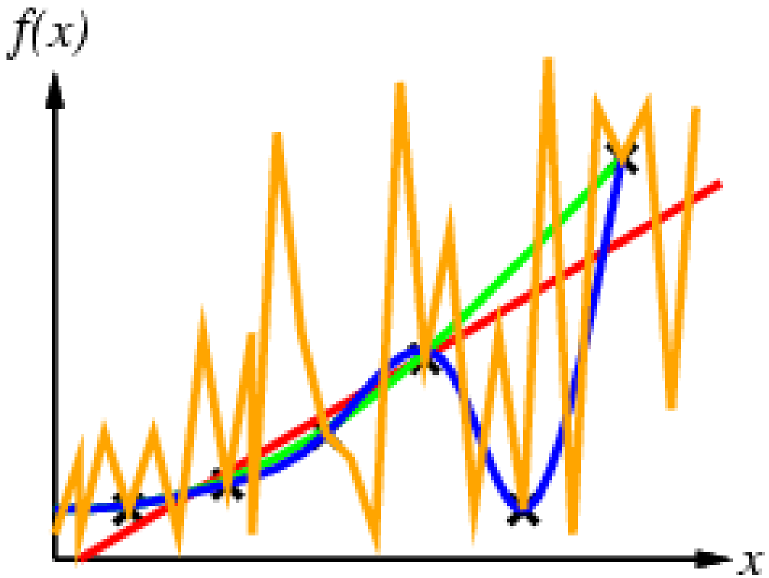
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



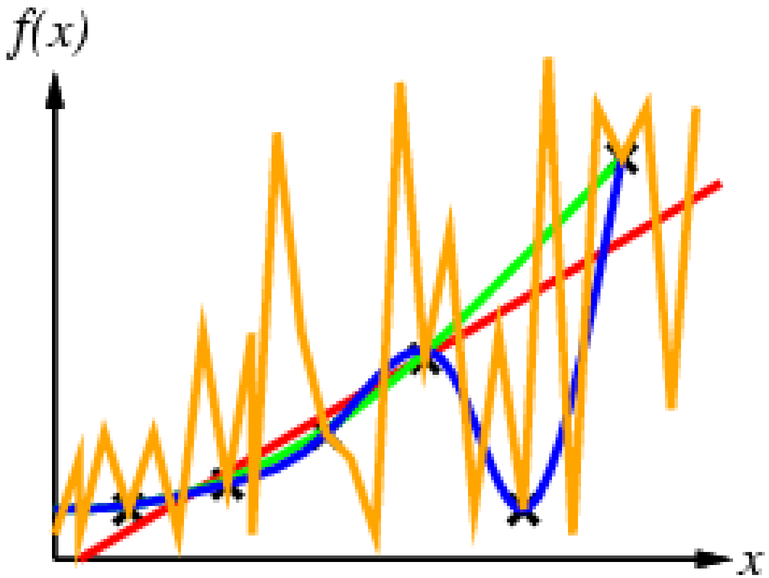
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



Ockham's razor: prefer the simplest hypothesis consistent with data

Inductive Learning Method

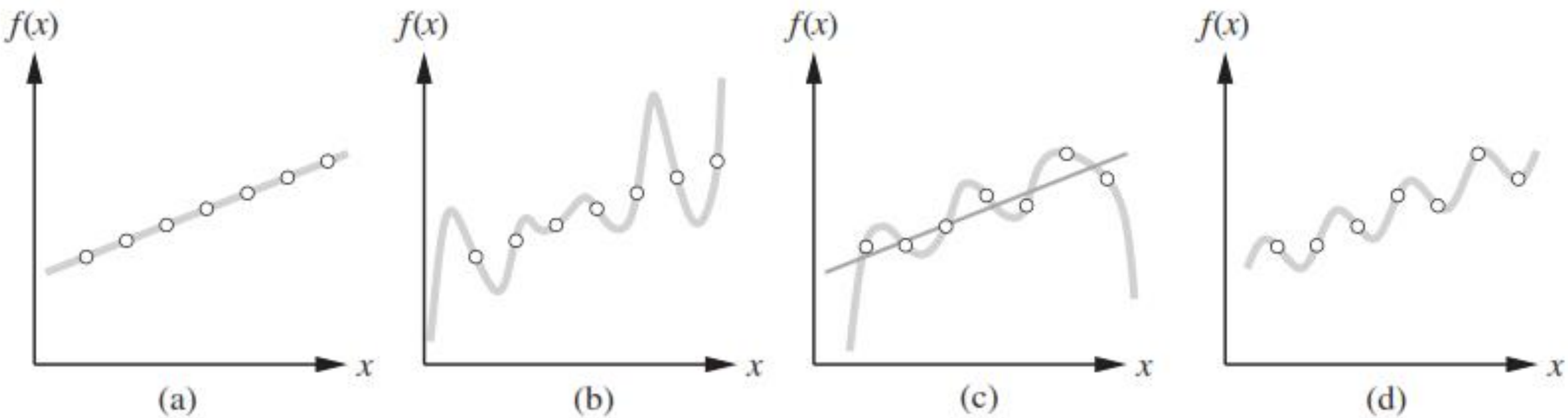


Figure 18.1 (a) Example $(x, f(x))$ pairs and a consistent, linear hypothesis. (b) A consistent, degree-7 polynomial hypothesis for the same data set. (c) A different data set, which admits an exact degree-6 polynomial fit or an approximate linear fit. (d) A simple, exact sinusoidal fit to the same data set.

Ockham's razor: prefer the simplest hypothesis consistent with data

Learning Types

Learning may be:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Supervised Learning

- In **supervised learning**, the agent observes some example input–output pairs and learns a function that maps from input to output. Supervised learning involves:
  - input features
  - target features
  - training examples

Supervised Learning

- The **training examples**
  - where the input features as well as the target features are specified.

- We have to predict the **target features** of a new example for which the input features are given.

- This is called,
  - **classification** when the target variables are discrete and
  - **regression** when the target features are continuous.

Supervised Learning

- Given a data set (training data)

$$D = \{\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \dots \langle x_m, y_m\rangle\}$$

Inputs

Outputs

- **Goal:** Find a hypothesis h in hypothesis class H that performs a good job of mapping x to y.
- When y_i is a boolean, or a member of a discrete set, the problem is a **classification** problem. When y_i is real-valued, we call this a **regression** problem.

Error Measure

Classification

- Y is discrete, a (small) finite, unordered set of classes

$$error\big(h(x), f(x)\big) = 0 \text{ if } h(x) = f(x) \text{ else } 1$$

0-1 Loss Error

Regression

- Y is continuous, a numeric set (typically real numbers)

$$error\big(h(x), f(x)\big) = (h(x) - f(x))^2$$

Squared Error

Unsupervised Learning

- In **unsupervised learning**, the agent learns patterns in the input even though **no explicit feedback** is supplied.
- **Unsupervised learning** occurs when no classifications are given and the learner must discover categories and regularities in the data.
- The most general example of unsupervised learning task is **clustering**:
  - potentially useful clusters developed from the input examples.
- For example, **a taxi agent** might gradually develop a concept of "good traffic days" and "bad traffic days".

Semi-Supervised Learning

- In **semi-supervised learning** we are given a few labeled examples and must make what we can of a large collection of unlabeled examples.
  - Some data is labeled but most of it is unlabeled and a mixture of supervised and unsupervised techniques can be used.
- Many real world machine learning problems fall into this type of learning.

Reinforcement Learning

- A supervised learning agent needs to be told the correct move for each position it encounters, but such feedback is seldom available.
- In the absence of feedback, **an agent can learn a transition model** for *its own moves* and can perhaps learn to predict the opponent's moves,
- *Without some feedback about what is good and what is bad, the agent will have no grounds for deciding which move to make*.

Reinforcement Learning

- In **reinforcement learning** the agent learns from a series of reinforcements—rewards or punishments.

- A win at the end of a chess game tells the agent it did something right.
    - It is up to the agent to decide which of the actions prior to the reinforcement were most responsible for it.

- The rewards **may come more frequently**, it depends upon the environment.

Reinforcement Learning

- Each percept(e) is enough to determine the State(the state is accessible)
- The agent can decompose the reward component from a percept.
- The **<u>agent task</u>**: *to find an optimal policy, mapping states to actions, that maximize long-run measure of the reinforcement*
  - Think of reinforcement as reward
- Can be modeled as **Markov Decision Processes** MDP model!

Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig
  - **Chapter 18.**