CS 4700:
Foundations of Artificial Intelligence

Prof. Bart Selman
selman@cs.cornell.edu

Module:
Intro Learning

Part V R&N --- Learning

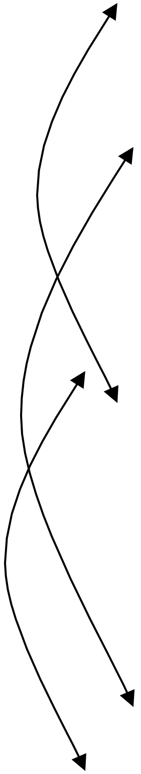
Chapter 18: Learning from Examples

Intelligence

Intelligence:

- “the capacity to learn and solve problems” (Webster dictionary)
- the ability to act rationally (requires reasoning)

What's involved in Intelligence?



A) Ability to interact with the real world

to perceive, understand, and act

speech recognition and understanding

image understanding (computer vision)

B) Reasoning and Planning

Part I and PartII

modelling the external world

problem solving, planning, and decision making

ability to deal with unexpected problems, uncertainties

C) Learning and Adaptation

Part III

We are continuously learning and adapting.

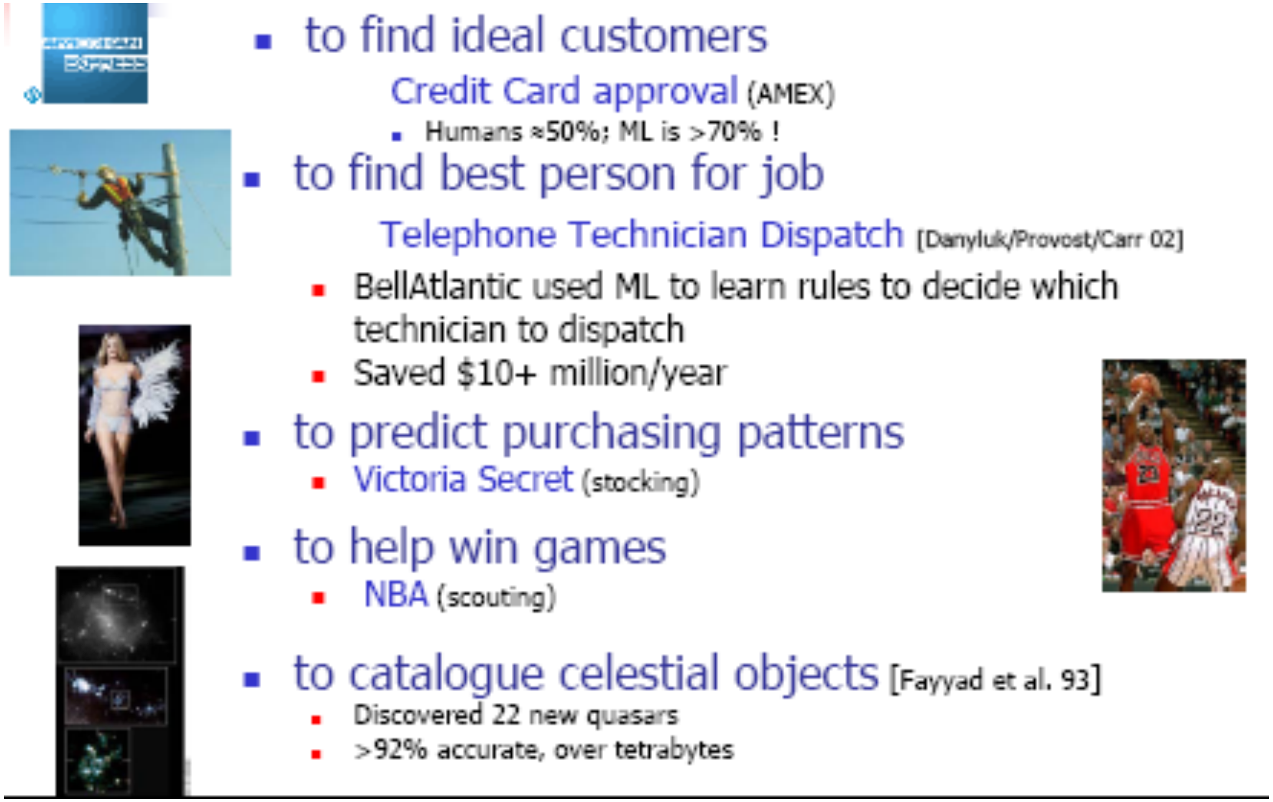
We want systems that adapt to us!

Learning

Examples

- Walking (motor skills)
- Riding a bike (motor skills)
- Telephone number (memorizing)
- Playing backgammon (strategy)
- Develop scientific theory (abstraction)
- Language
- Recognize fraudulent credit card transactions
- Etc.

Different Learning tasks

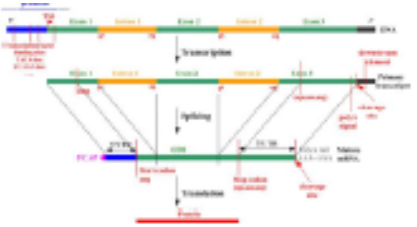


- to find ideal customers
  - Credit Card approval (AMEX)
    - Humans ≈50%; ML is >70% !
- to find best person for job
  - Telephone Technician Dispatch [Danyluk/Provost/Carr 02]
  - BellAtlantic used ML to learn rules to decide which technician to dispatch
  - Saved $10+ million/year
- to predict purchasing patterns
  - Victoria Secret (stocking)
- to help win games
  - NBA (scouting)
- to catalogue celestial objects [Fayyad et al. 93]
  - Discovered 22 new quasars
  - >92% accurate, over tetrabytes

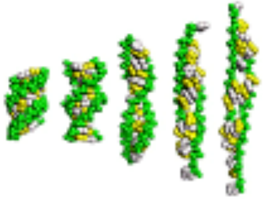
Source: R. Greiner

Different Learning Tasks

- **BioInformatics 1:** identifying genes
  - Glimmer [Delcher et al, 95]
  - identifies 97⁺% of genes, automatically!



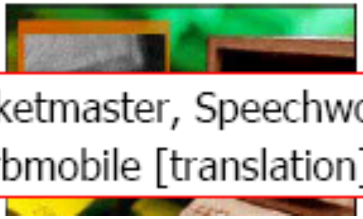
- **BioInformatics 2:** Predicting protein function, …



- **Recognizing Handwriting**

US postal service zip code reader

- **Recognizing Spoken Words**
  - **"How to wreck a nice beach"**

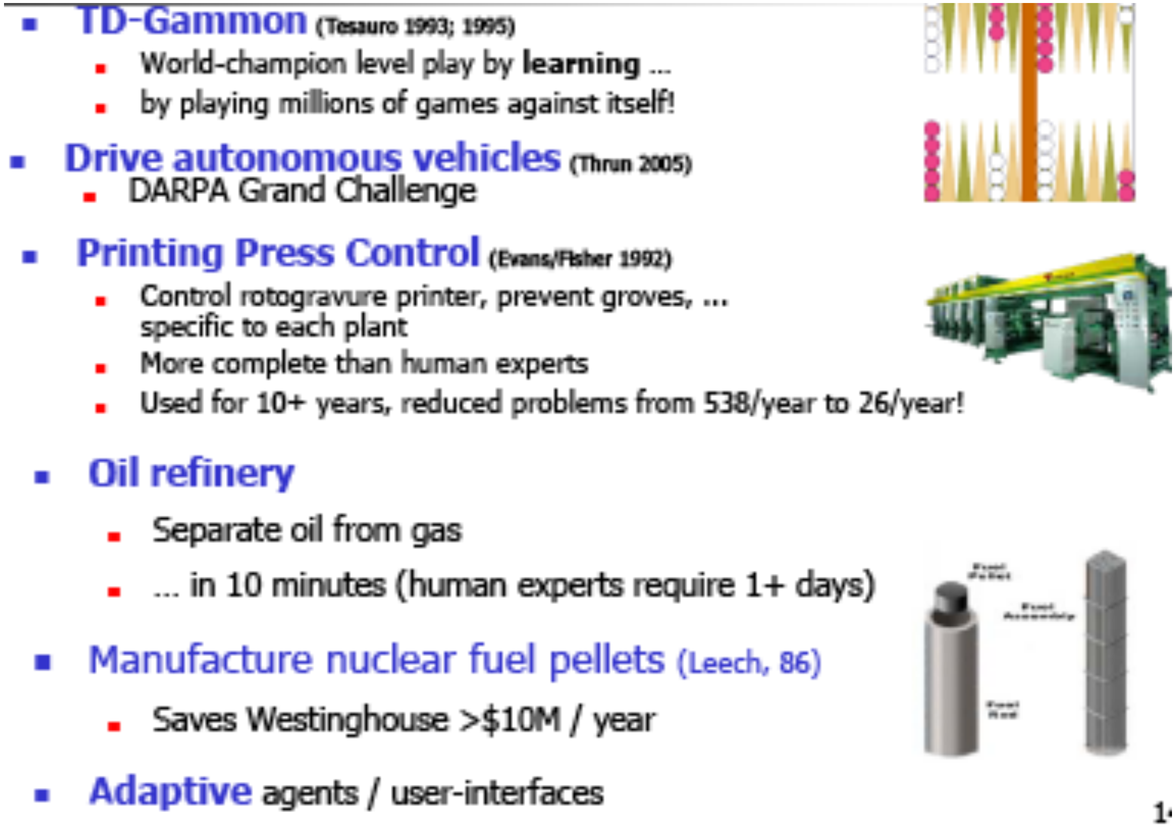


Ticketmaster, Speechworks, Bell, Verbmobile [translation]

????

problems in developing systems that recognize spontaneous speech → How to recognize speech

Different Learning Tasks



- **TD-Gammon** (Tesauro 1993; 1995)
  - World-champion level play by **learning** ...
  - by playing millions of games against itself!
- **Drive autonomous vehicles** (Thrun 2005)
  - DARPA Grand Challenge
- **Printing Press Control** (Evans/Fisher 1992)
  - Control rotogravure printer, prevent groves, ... specific to each plant
  - More complete than human experts
  - Used for 10+ years, reduced problems from 538/year to 26/year!
- **Oil refinery**
  - Separate oil from gas
  - ... in 10 minutes (human experts require 1+ days)
- Manufacture nuclear fuel pellets (Leech, 86)
  - Saves Westinghouse >$10M / year
- **Adaptive** agents / user-interfaces

(One) Definition of Learning

Definition [Mitchell]:

A computer program is said to learn from

- experience E with respect to some class of
- tasks T and
- performance measure P,

if its performance at tasks in T, as measured by P, improves with experience E.

Examples

Spam Filtering

- T: Classify emails HAM / SPAM
- E: Examples (e_1,HAM),(e_2,SPAM),(e_3,HAM),(e_4,SPAM), ...
- P: Prob. of error on new emails

Personalized Retrieval

- T: find documents the user wants for query
- E: watch person use Google (queries / clicks)
- P: # relevant docs in top 10

Play Checkers

- T: Play checkers
- E: games against self
- P: percentage wins

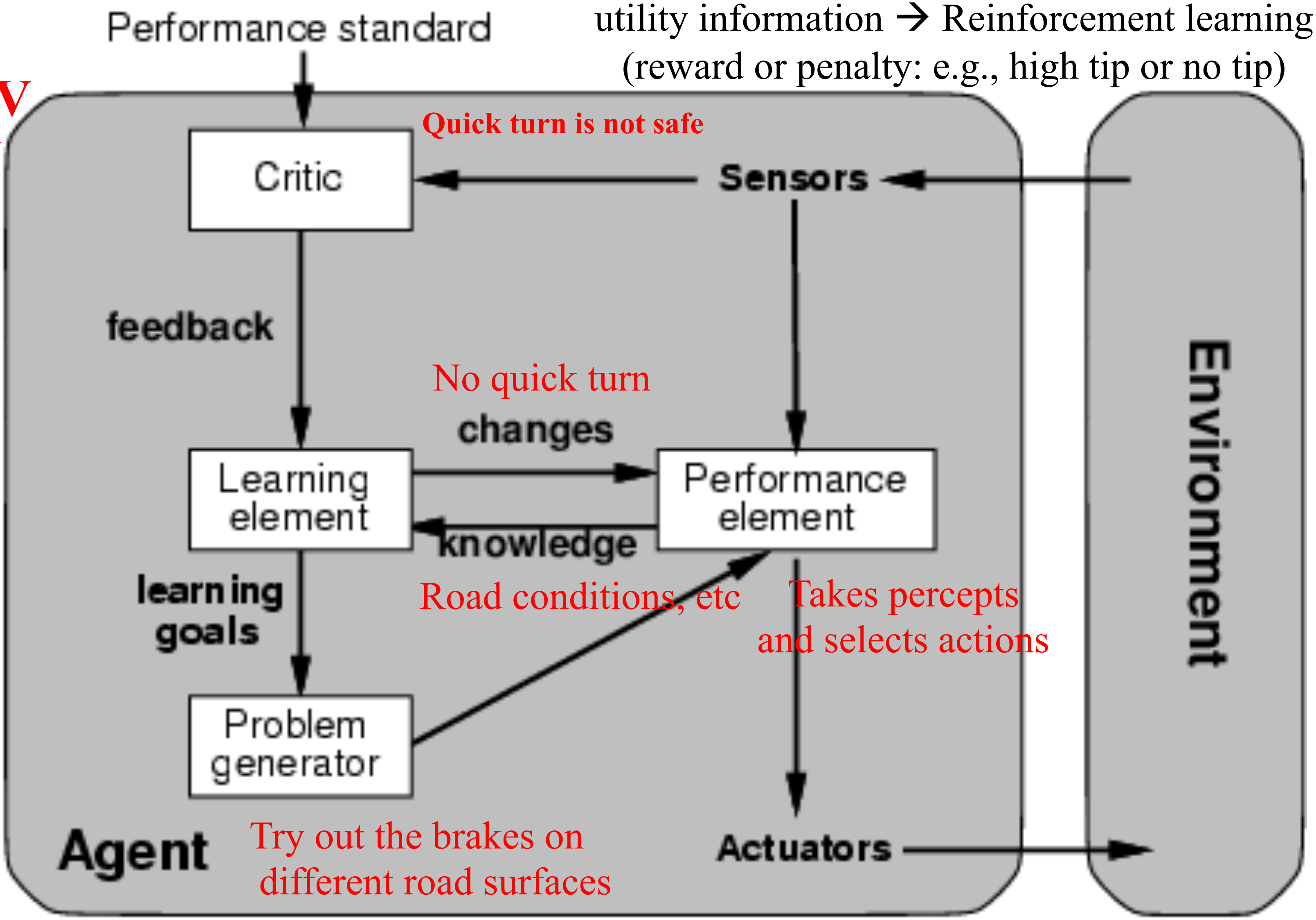
Module: Learning

Learning enables an agent to modify its decision mechanisms to improve performance

Learning agents

More complicated when agent needs to learn utility information → Reinforcement learning (reward or penalty: e.g., high tip or no tip)

Part V
R&N



Quick turn is not safe

No quick turn

Road conditions, etc

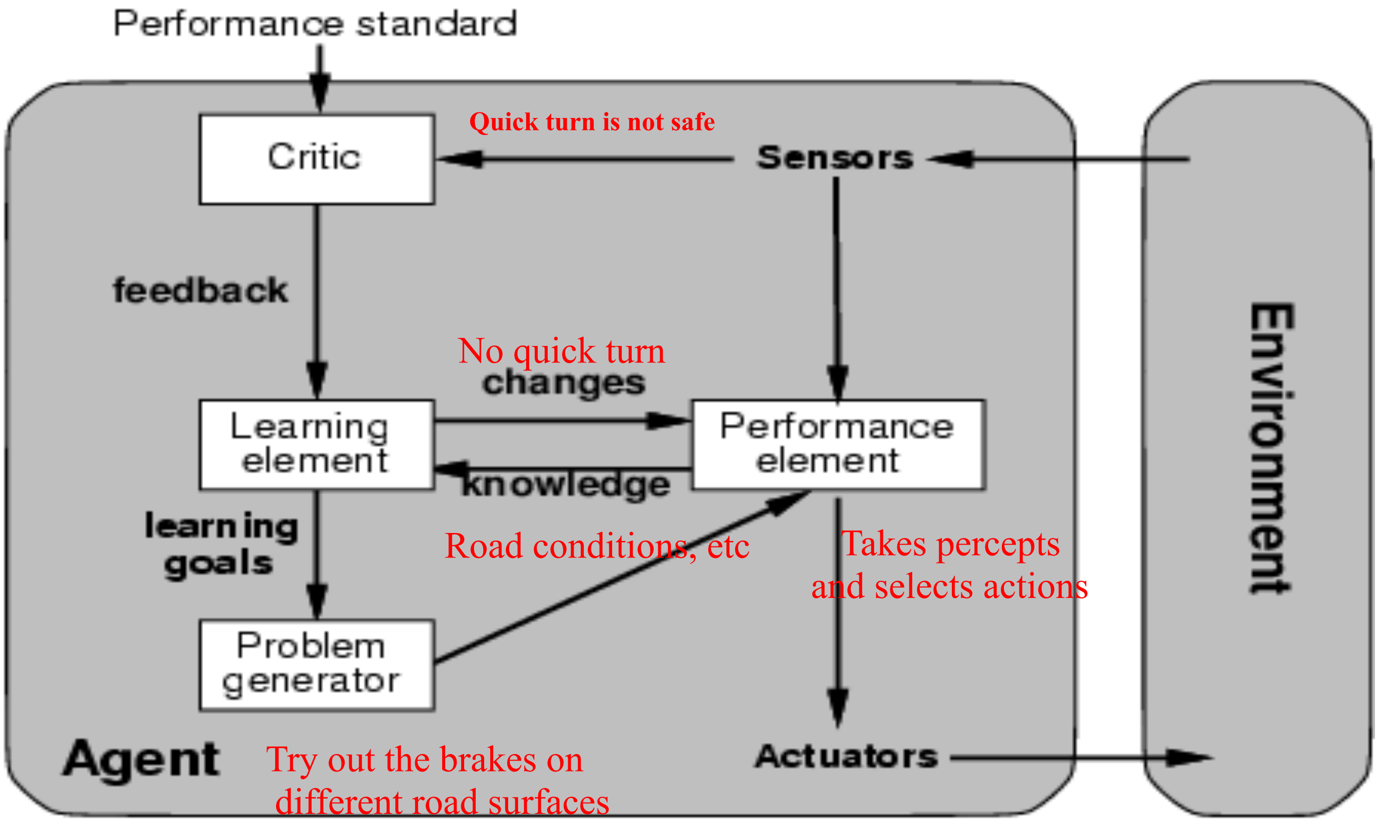
Takes percepts and selects actions

Try out the brakes on different road surfaces

A General Model of Learning Agents

Design of a learning element is affected by

- •What feedback is available to learn these components
- •Which components of the performance element are to be learned
- •What representation is used for the components



Learning: Types of learning

rote learning - (memorization) -- storing facts – no inference.

learning from instruction - Teach a robot how to hold a cup.

learning by analogy - transform existing knowledge to new situation;

→ learn how to hold a cup and learn to hold objects with a handle.

learning from observation and discovery – unsupervised learning; ambitious → goal of science! → cataloguing celestial objects.

→ **learning from examples –** special case of inductive learning - well studied in machine learning. Example of good/bad credit card customers.

–Carbonell, Michalski & Mitchell.

Learning:
Type of feedback

Supervised Learning

→ learn a **function** from examples of its inputs and outputs.

→ Example – an agent is presented with many camera images and is told which ones contain buses; the agent learns a function from images to a Boolean output (whether the image contains a bus)

→ Learning decision trees is a form of supervised learning

Unsupervised Learning

→ learn a patterns in the input when no specific output values are supplied

→ Example: Identify communities in the Internet; identify celestial objcets

Reinforcement Learning

→ learn from reinforcement or (occasional) rewards --- most general form of learning

→ Example: An agent learns how to play Backgammon by playing against itself; it gets a reward (or not) at the end of each game.

Learning: Type of representation and Prior Knowledge

Type of representation of the learned information

- → Propositional logic (e.g., Decision Trees)
- → First order logic (e.g., Inductive Logic Programming)
- → Probabilistic descriptions (E.g. Bayesian Networks)
- → Linear weighted polynomials (E.g., utility functions in game playing)
- → Neural networks (which includes linear weighted polynomials as special case; (E.g., utility functions in game playing)

Availability of Prior Knowledge

- → No prior knowledge (majority of learning systems)
- → Prior knowledge (E.g., used in statistical learning)

Inductive Learning Example

Food	Chat	Fast	Price	Bar	BigTip
(3)	(2)	(2)	(3)	(2)	
great	yes	yes	normal	no	yes
great	no	yes	normal	no	yes
mediocre	yes	no	high	no	no
great	yes	yes	normal	yes	yes

Instance Space X: Set of all possible objects described by attributes (often called features).

Target Function f: Mapping from Attributes to Target Feature (often called label) (f is unknown)

Hypothesis Space H: Set of all classification rules h_i we allow.

Training Data D: Set of instances labeled with Target Feature

Inductive Learning / Concept Learning

Task:

- Learn (to imitate) a function f: X → Y

Training Examples:

- Learning algorithm is given the correct value of the function for particular inputs → **training examples**
- An **example** is a pair (*x*, *f*(*x*)), where *x* is the input and *f*(*x*) is the output of the function applied to *x*.

Goal:

- Learn a function h: X → Y that approximates f: X → Y as well as possible.

Classification and Regression Tasks

Naming:

If Y is a discrete set, then called “classification”.

If Y is a not a discrete set, then called “regression”.

Examples:

Steering a vehicle: **road image → direction to turn the wheel (how far)**

Medical diagnosis: **patient symptoms → has disease / does not have disease**

Forensic hair comparison: **image of two hairs → match or not**

Stock market prediction: **closing price of last few days → market will go up or down tomorrow (how much)**

Noun phrase coreference: **description of two noun phrases in a document → do they refer to the same real world entity**

Inductive Learning Algorithm

Task:

- Given: collection of examples
- Return: a function *h* (*hypothesis*) that approximates *f*

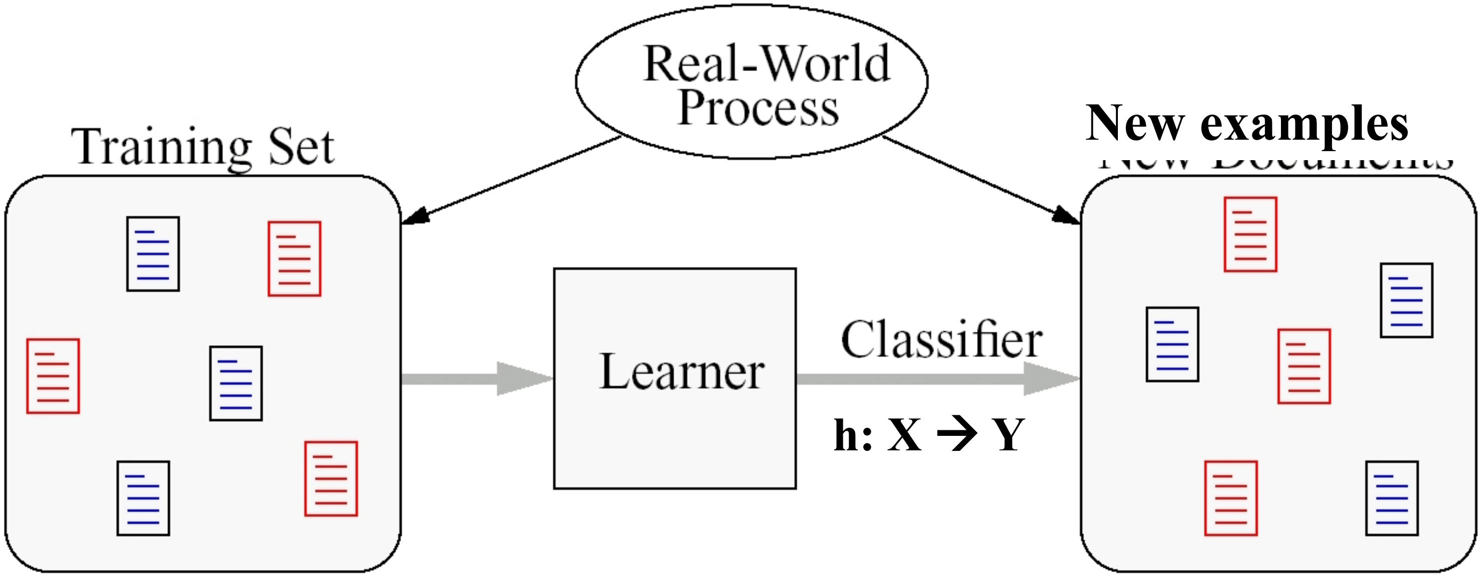
Inductive Learning Hypothesis:

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over any other unobserved examples.

Assumptions of Inductive Learning:

- The training sample represents the population
- The input features permit discrimination

Inductive Learning Setting



Task:

Learner (or inducer) induces a general rule h from a set of observed examples that classifies new examples accurately. An algorithm that takes as input specific instances and produces a model that generalizes beyond these instances.

Classifier - A mapping from unlabeled instances to (discrete) classes.

Classifiers have a form (e.g., decision tree) plus an interpretation procedure (including how to handle unknowns, etc.)

Inductive learning: Summary

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:

- Ignores prior knowledge
- Assumes examples are given)

→ Learning a discrete function is called classification learning.

→ Learning a continuous function is called regression learning.

Inductive learning method

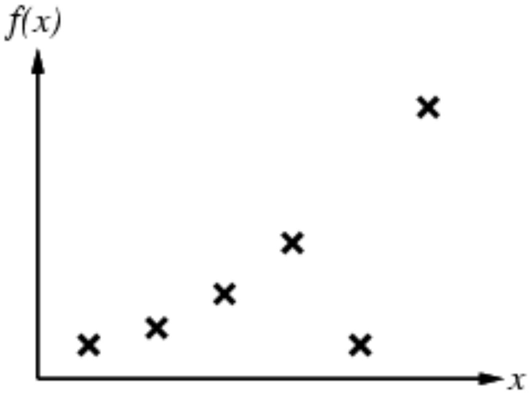
Fitting a function of a single variable to some data points

Examples are (x, f(x) pairs;

Hypothesis space H – set of hypotheses we will consider for function f, in this case **polynomials of degree at most k**

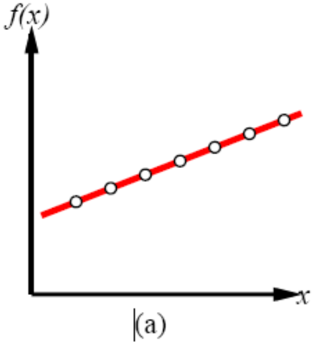
Construct/adjust *h* to agree with *f* on training set

(*h* is consistent if it agrees with *f* on all examples)

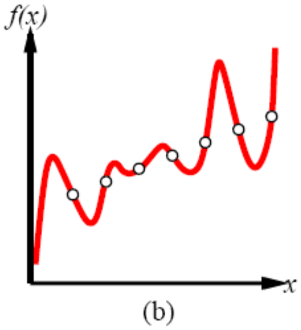


Multiple consistent hypotheses?

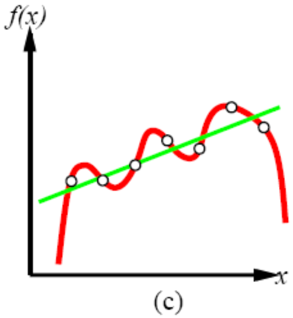
Polynomials of degree at most k



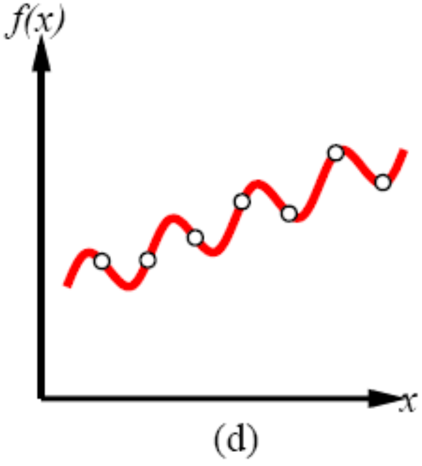
Linear hypothesis



Degree 7 polynomial hypothesis



Degree 6 polynomial and approximate linear fit



Sinusoidal hypothesis

How to choose from among multiple consistent hypotheses?

Ockham's razor: ***maximize a combination of consistency and simplicity***

Preference Bias: Ockham's Razor

Aka Occam's Razor, Law of Economy, or Law of Parsimony

Principle stated by William of Ockham (1285-1347/49), an English philosopher, that

- "*non sunt multiplicanda entia praeter necessitatem*"
- or, entities are not to be multiplied beyond necessity.

The simplest explanation that is consistent with all observations is the best.

- E.g, the smallest decision tree that correctly classifies all of the training examples is the best.
- Finding the provably smallest decision tree is NP-Hard, so instead of constructing the absolute smallest tree consistent with the training examples, construct one that is pretty small.

Different Hypothesis Spaces

Learning can be seen as fitting a function to the data. We can consider different functions as the target function and therefore different hypothesis spaces. Examples:

Propositional if-then rules
Decision Trees
First-order if-then rules
First-order logic theory
Linear functions
Polynomials of degree at most k
Neural networks
Java programs
Etc

Tradeoff in expressiveness and complexity

A learning problem is realizable if its hypothesis space contains the true function.

Why not pick the largest possible hypothesis space, say the class of all Turing machines?

Tradeoff between expressiveness of a hypothesis space and the complexity of finding simple, consistent hypotheses within the space (also risk of "overfitting").
Extreme overfitting: Just remember all training examples.

Summary

Learning needed for unknown environments.

Learning agent = performance element + learning element

For supervised learning, the aim is to find a simple hypothesis approximately consistent with training examples.