Introduction to Artificial Intelligence
COMP 3501 / COMP 4704-4
Lecture 13: Supervised Learning

Prof. Nathan Sturtevant

Today

- Learning
- Decision Trees

Learning

- Any aspect of an agent can (potentially) be improved through learning
- Depends on:
  - What component to be improved
  - Prior knowledge of the agent
  - Representation used for data & the component
  - What feedback is available for learning

Types of learning

- Unsupervised learning
  - Learning about data by looking at its features
  - No specific feedback from users
  - Usually entails clustering data

Types of learning

- Reinforcement learning
  - Agents with sensors experience the world
  - As they act they receive positive and negative rewards
  - The agent then learns value of (sensor) states

Types of learning

- Supervised learning
  - Agent is given example input and correct output
  - Goal is to build general model that will produce correct ouput on novel input

Types of learning

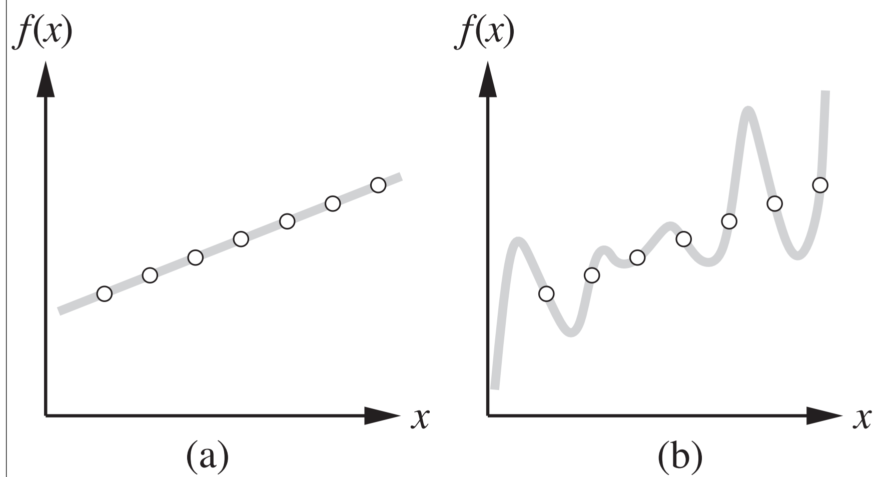
- Semi-supervised learning
  - Some labeled examples in data set
  - Some mislabeled examples
  - Learn generalized model

Supervised learning

- Given a *training set* of N example input & outputs
  - $(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$
  - Where each y_i comes from an unknown function
    - $y_i = f(x_i)$
  - Discover a function h such that $h(x_i) \approx f(x_i)$
- Think of *h* as a hypothesis, and we are searching for the “best” hypothesis

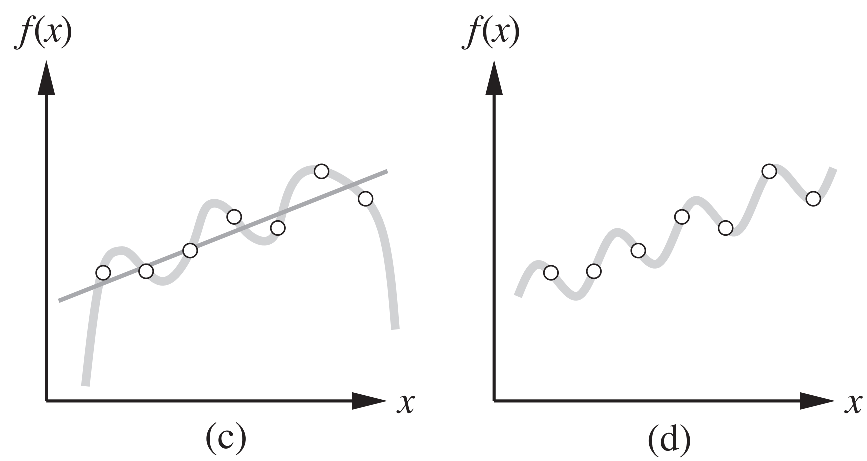
Supervised learning

- All available data is usually broken into:
  - Training set: exclusively used for study & training
  - Test set: exclusively used for testing
- Ensures that the learning generalizes from training data to test data
  - Want to avoid overfitting data



Ockham's razor

- GIven multiple possible hypotheses that explain the data, choose the simplest one
  - 1st degree polynomial is probably better than a 3rd degree polynomial
- Decision isn't always clear

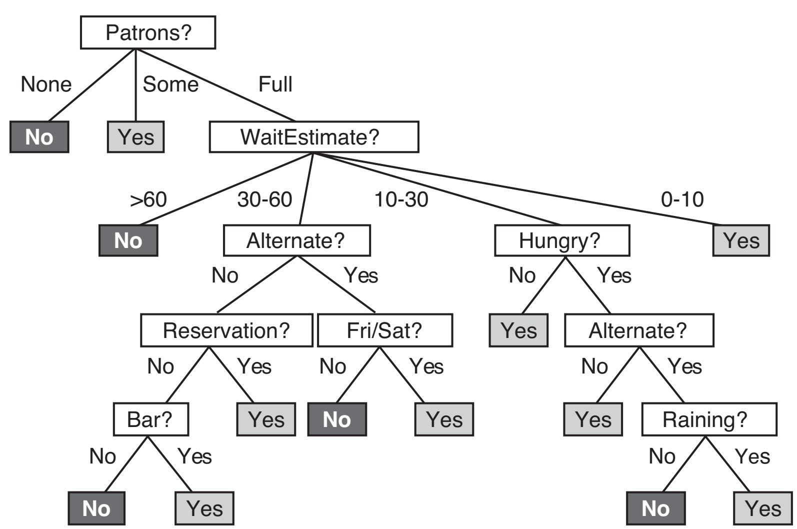


Decision Trees

- A decision tree is a simple classifier
- Training input:
  - Data points with a set of attributes
- Classifier output:
  - Can be boolean or have multiple outputs
  - Each leaf stores an “answer”

Example

- Should we wait for a table at a restaurant?
- Possible attributes:
  - Alternate restaurant nearby?
  - Is there a bar to wait in?
  - Is it Friday or Saturday?
  - How hungry are we?
  - How busy is the restaurant?
  - How many people in the restaurant?



Representation

- The states which reach each outcome can be represented by written as the disjunction (or) of each possible path of decisions
- What about a decision tree for N boolean inputs:
  - Are more than N/2 inputs true?

General Approach

- Greedy approaches work well
  - Choose the category that divides into the best sub-problems

Example

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

Figure 18.3 Examples for the restaurant domain.

Recursive splitting

- Choosing and assigning to a node in the decision tree to an attribute produces a smaller decision tree problem
  - When all examples have the same outcome; done.
  - If examples are split, choose another attribute
  - If there are no examples, set default value
  - If there are no attributes left, there are conflicting examples (use the best classification)

Measuring the best splitting

- The choice for splitting is defined in terms of entropy
  - Entropy measures uncertainty
    - A fair coin has 1-bit of entropy
    - A 4-sided die has 2 bits of entropy
- Entropy of a random variable V with values v_k and probabilities $P(v_k)$ is:

$$-\sum_k P(v_k) log_2 P(v_k)$$

Entropy examples

$$-\sum_k P(v_k) log_2 P(v_k)$$

- Entropy of a fair coin:
  - $-(0.5 \log_2(0.5) + 0.5 \log_2(0.5)) = 1$
- Entropy of a coin which is heads 99% of the time:
  - $-(0.99 \log_2(0.99) + 0.01 \log_2(0.01)) \approx 0.08$

Entropy & Decision tree learning

- Let B(q) be the entropy of a boolean variable with probability q of being true
- Assume the training set has p positive and n negative examples
  - H(Goal) = B(p / p+n)
  - This is the entropy of the problem being decided

Entropy & Decision tree learning

- Measure the change in entropy after splitting on a variable A

$$Remainder(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B(\frac{p_k}{p_k + n_k})$$

- The gain of splitting on A is:

$$Gain(a) = B(\frac{p}{p+n}) - Remainder(A)$$

- Gain(Patrons) = 0.541 bits
- Gain(Type) = 0 bits

Class Example

- Everyone provide an example for what we should do tonight.
- Choices:
  - Go out with friends
  - Stay in with friends
  - Stay in and work/sleep
- Features
  - HW: high, medium, low
  - Tired: high, medium, low
  - (other features?)