Decision Trees

CMSC 422

Marine Carpuat
marine@cs.umd.edu

Credit: some examples & figures by Tom Mitchell

Last week: introducing machine learning

What does “learning by example” mean?

- Classification tasks

- Learning requires examples + inductive bias
- Generalization vs. memorization

- Formalizing the learning problem
  - Function approximation
  - Learning as minimizing expected loss

Machine Learning as Function Approximation

Problem setting

- Set of possible instances X
- Unknown target function $f: X \rightarrow Y$
- Set of function hypotheses $H = \{h \mid h: X \rightarrow Y\}$

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$ of unknown target function f

Output

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

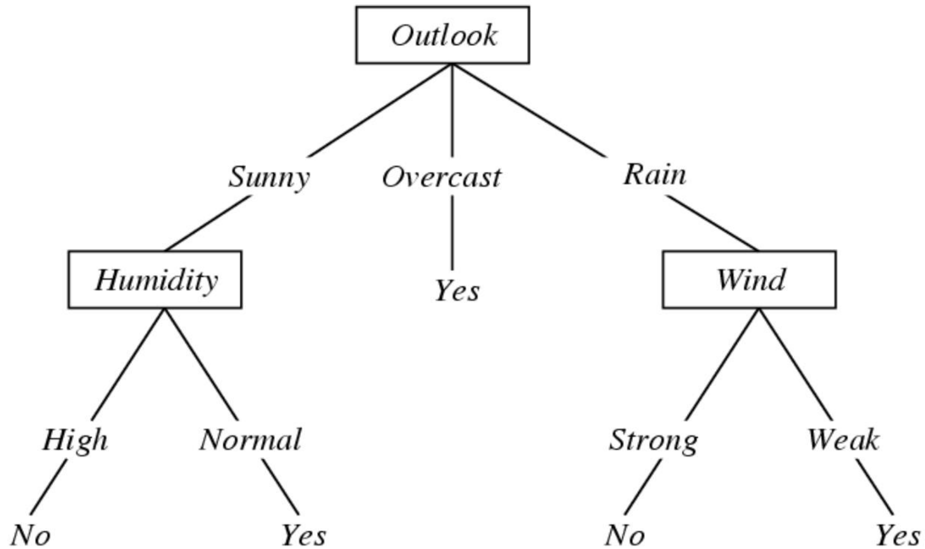
Today: Decision Trees

- **What is a decision tree?**
- How to learn a decision tree from data?
- What is the inductive bias?
- Generalization?

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

A decision tree to decide whether to play tennis



Decision Trees

- Representation
  - Each internal node tests a feature
  - Each branch corresponds to a feature value
  - Each leaf node assigns a classification
    - or a probability distribution over classifications

- Decision trees represent functions that map examples in X to classes in Y

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?

Exercise

- How would you represent the following Boolean functions with decision trees?
  - AND
  - OR
  - XOR
  - $(A \cap B) \cup (C \cap \neg D)$

Today: Decision Trees

- What is a decision tree?
- **How to learn a decision tree from data?**
- What is the inductive bias?
- Generalization?

Function Approximation with Decision Trees

Problem setting

- Set of possible instances X
  - Each instance $x \in X$ is a feature vector $x = [x_1, \dots, x_D]$
- Unknown target function $f: X \rightarrow Y$
  - Y is discrete valued
- Set of function hypotheses $H = \{h \mid h: X \rightarrow Y\}$
  - Each hypothesis h is a decision tree

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \dots (x^{(N)}, y^{(N)})\}$ of unknown target function f

Output

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

Decision Trees Learning

- Finding the hypothesis $h \in H$
  - That minimizes training error
  - Or maximizes training accuracy

- How?
  - H is too large for exhaustive search!
  - We will use a **heuristic search** algorithm
    - Pick questions to ask, in order
    - Such that classification accuracy is maximized

Top-down Induction of Decision Trees

```
CurrentNode = Root

DTtrain(examples for CurrentNode,features at CurrentNode):
    1. Find F, the “best” decision feature for next node
    2. For each value of F, create new descendant of node
    3. Sort training examples to leaf nodes
    4. If training examples perfectly classified
         Stop
       Else
         Recursively apply DTtrain over new leaf nodes
```

How to select the "best" feature?

- A good feature is a feature that lets us make correct classification decision

- Criteria to measure how good a feature is
  - classification accuracy
  - entropy
  - …

- Let's try it on the PlayTennis dataset

Let's build a decision tree using features W, H, T

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Partitioning examples according to Humidity feature

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Partitioning examples: H = Normal

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

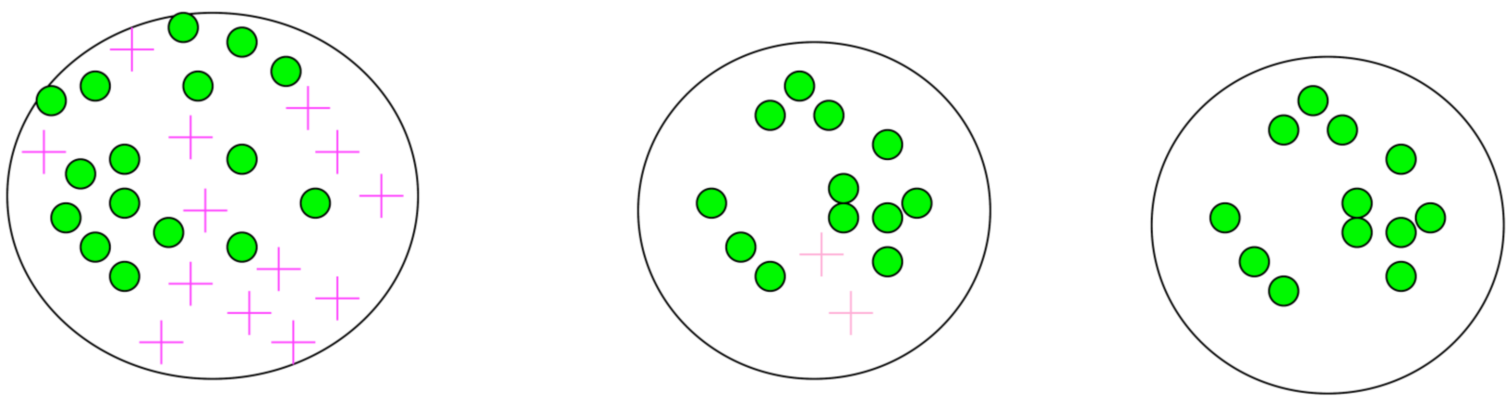
Partitioning examples: H = Normal and W = Strong

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

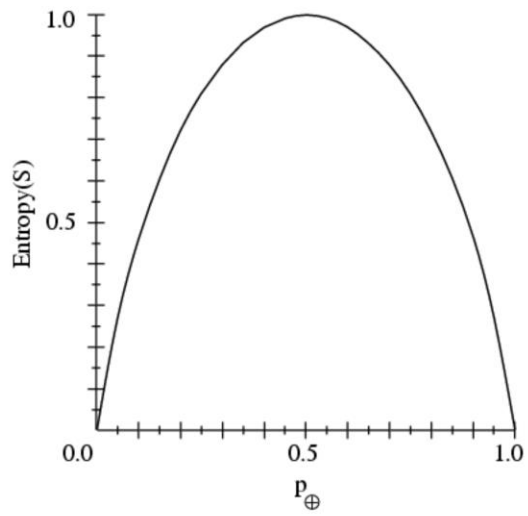
Another feature selection criterion: Entropy

- Used in the ID3 algorithm [Quinlan, 1963]
  - pick feature with smallest entropy to split the examples at current iteration

- Entropy measures impurity of a sample of examples



Sample Entropy



- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S
- $p_{\ominus}$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy

Entropy $H(X)$ of a random variable X

\# of possible values for X

$$H(X) = -\sum_{i=1}^{n} P(X=i)\log_2 P(X=i)$$

$H(X)$ is the expected number of bits needed to encode a randomly drawn value of X (under most efficient code)

Why? Information theory:

- Most efficient possible code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random X is:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

A decision tree to predict C-sections

Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05-
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

A decision tree to distinguish homes in New York from homes in San Francisco

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Today: Decision Trees

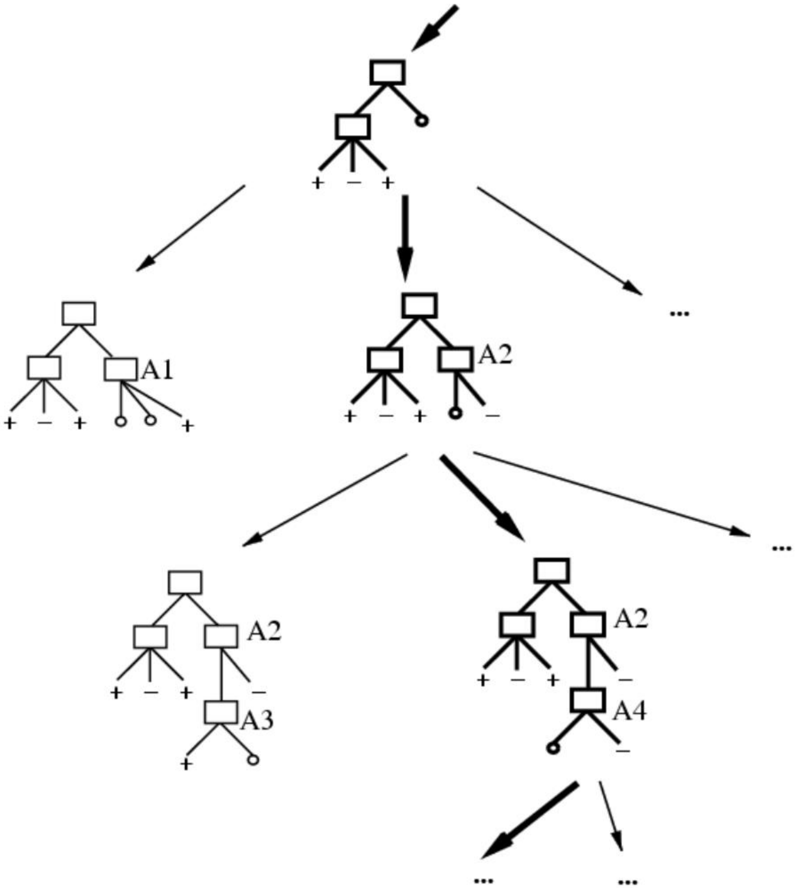
- What is a decision tree?
- How to learn a decision tree from data?
- **What is the inductive bias?**
- Generalization?

Inductive bias in decision tree learning

```
CurrentNode = Root

DTtrain(examples for CurrentNode,features at CurrentNode):
    1. Find F, the “best” decision feature for next node
    2. For each value of F, create new descendant of node
    3. Sort training examples to leaf nodes
    4. If training examples perfectly classified
         Stop
       Else
         Recursively apply DTtrain over new leaf nodes
```

Inductive bias in decision tree learning



- Our learning algorithm performs heuristic search through space of decision trees
- It stops at smallest acceptable tree
- Occam's razor: prefer the simplest hypothesis that fits the data

Why prefer short hypotheses?

- Pros
  - Fewer short hypotheses than long ones
    - A short hypothesis that fits the data is less likely to be a statistical coincidence

- Cons
  - What's so special about short hypotheses?

Today: Decision Trees

- What is a decision tree?
- How to learn a decision tree from data?
  - Top-down induction to minimize classification error
- What is the inductive bias?
  - Occam's razor: preference for short trees