

# Intro. to machine learning (CSI 5325)

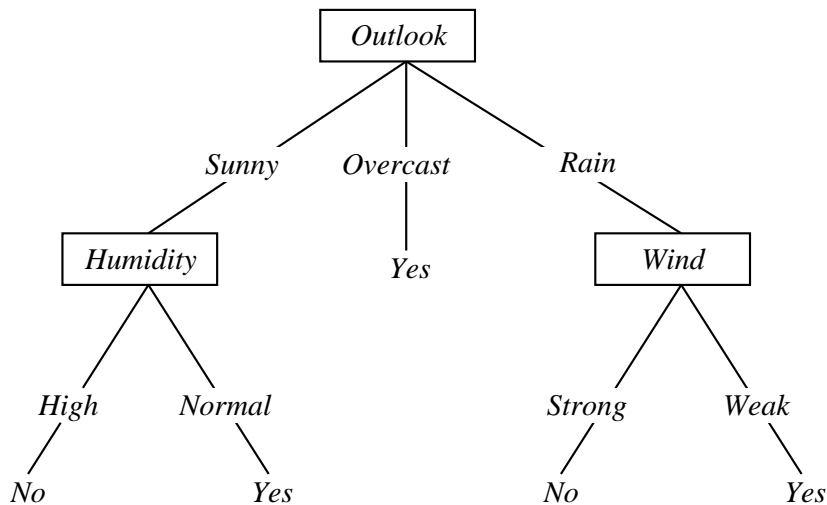
## Lecture 4: decision trees

Greg Hamerly

Fall 2008

Some content from Tom Mitchell.

- 1 Decision tree representation
- 2 ID3 learning algorithm
- 3 Entropy, Information gain
- 4 ID3 algorithm behavior

Decision Tree for *PlayTennis*

# A Tree to Predict C-Section Risk

Learned from medical records of 1000 women

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-
```

# Decision Trees

Decision tree representation:

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent:

- $\wedge, \vee, \text{XOR}$
- $(A \wedge B) \vee (C \wedge \neg D \wedge E)$
- $M$  of  $N$

# When to Consider Decision Trees

- Instances describable by attribute–value pairs
- Target function is discrete valued
- Disjunctive hypothesis may be required
- Possibly noisy training data

## Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling influences on EU legislatures

# Top-Down Induction of Decision Trees

Main loop:

- 1  $A \leftarrow$  the “best” decision attribute for next *node*
- 2 Assign  $A$  as decision attribute for *node*
- 3 For each value of  $A$ , create new descendant of *node*
- 4 Sort training examples to leaf nodes
- 5 If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

# Reading decision trees

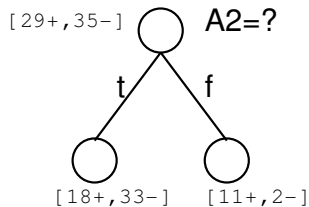
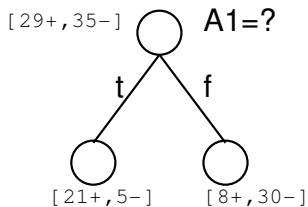
Typically, most influential attribute is chosen first.

Less-influential attributes are chosen later.

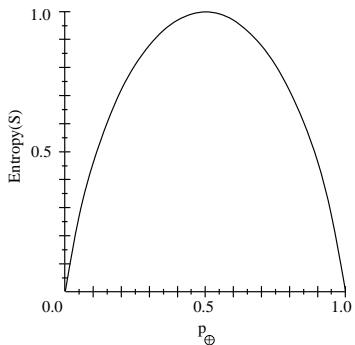
Trees are easy to read and interpret.

# Choosing attributes

Which attribute is best?



# 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  $S$

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

# Entropy

*Entropy*( $S$ ) = expected number of bits needed to encode class ( $\oplus$  or  $\ominus$ ) of randomly drawn member of  $S$  (under the optimal, shortest-length code).

Why?

Information theory: optimal length code assigns  $-\log_2 p$  bits to message having probability  $p$ .

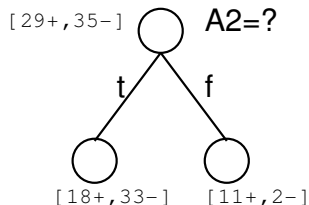
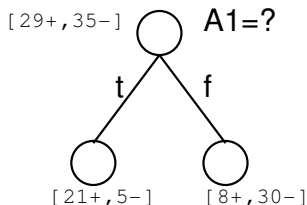
So, expected number of bits to encode  $\oplus$  or  $\ominus$  of random member of  $S$ :

$$p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus})$$

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

# Information Gain

$$\begin{aligned}
 \text{Gain}(S, A) &= \text{expected reduction in entropy due to sorting on } A \\
 &\equiv \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)
 \end{aligned}$$



# Information Gain = Lowering entropy

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

The information gain of different attributes for the same node (the same  $S$ ) all have the same  $Entropy(S)$ .

The important part is the second term, the average entropy of the split nodes:  $\sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$

We want  $Gain(S, A)$  to be high, so we want this part to be as low as possible – we want to lower the entropy of the predicted variable by selecting on some attribute.

## Entropy for multiple values

Entropy is well-defined for any discrete random variable... even one having more than two values.

$N$  is the random variable giving the number of classes a particular student shows up for this week.

- $0 \leq N \leq 2$
- $P(N = 0) + P(N = 1) + P(N = 2) = 1$

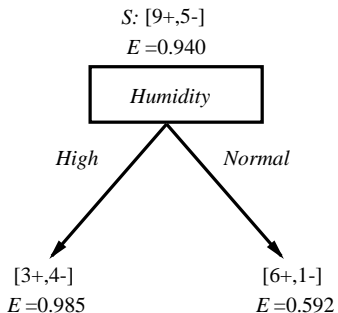
$$\begin{aligned} \text{Entropy}(N) &= -P(N = 0) \log_2(P(N = 0)) \\ &\quad -P(N = 1) \log_2(P(N = 1)) \\ &\quad -P(N = 2) \log_2(P(N = 2)) \end{aligned}$$

# Training Examples

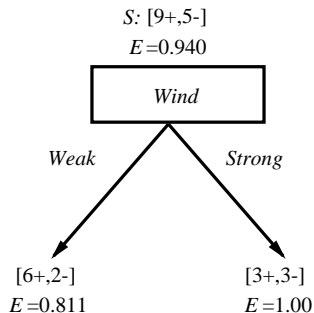
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal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

# Selecting the Next Attribute

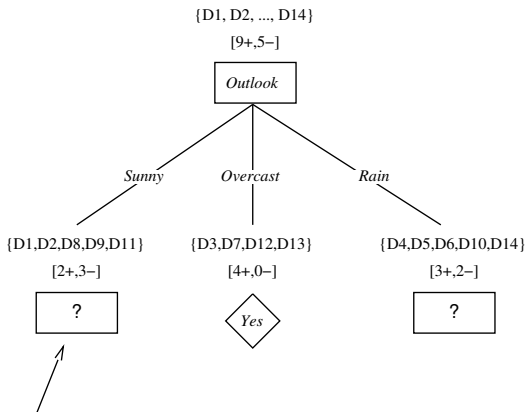
Which attribute is the best classifier?



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$



$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$



Which attribute should be tested here?

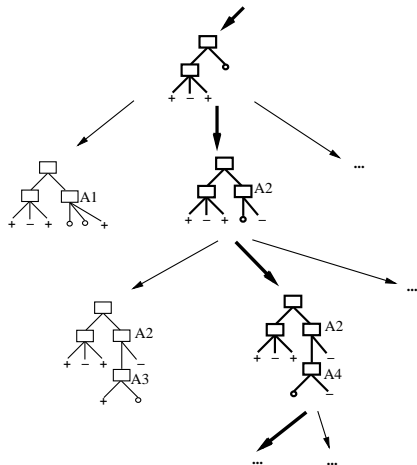
$$S_{\text{sunny}} = \{D1, D2, D8, D9, D11\}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

# Hypothesis Space Search by ID3



# Hypothesis Space Search by ID3

- Hypothesis space is complete!
  - Target function surely in there...
- Outputs a single hypothesis (which one?)
  - Can't play 20 questions...
- No back tracking
  - Local minima...
- Statically-based search choices
  - Robust to noisy data...
- Inductive bias: approx “prefer shortest tree”