DECISION TREES

Today

- Reading
  - AIMA 18.3, 18.7
- Goals
  - Introduce decision tree classifier
  - ID-3 algorithm for learning a decision tree classifier
  - Using entropy to choose attributes

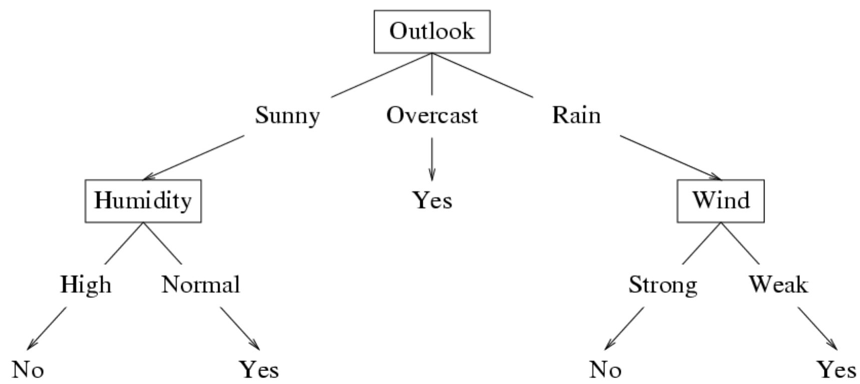
Wednesday's Class

- Guest lecture at HMC on NLP!
- Same time as our class 1:15-2:30pm
- Location is Big Beckman B126
  - Big Beckman is in basement of Olin Science Center
  - Head down middle stairs in Olin

Decision Tree Classifier

	Day	Outlook	Temp.	Humidity	Wind	PlayTennis	
$\mathbf{x}_1 \rightarrow$	D1	Sunny	Hot	High	Weak	No	$\leftarrow y_1$
$\mathbf{x}_2 \rightarrow$	D2	Sunny	Hot	High	Strong	No	$\leftarrow y_2$
$\mathbf{x}_3 \rightarrow$	D3	Overcast	Hot	High	Weak	Yes	$\leftarrow y_3$
	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	

Decision Tree Classifier

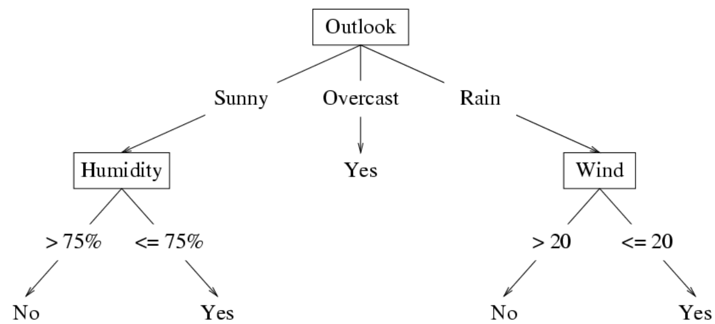


Decision Tree Classifier

- Decision trees are best suited to problems where
  - Each attribute is discrete
  - The label y is discrete
  - The labels may contain errors
  - The training data may contain missing attribute values
  - The hypothesis can be expressed using disjunctions (OR) of conjunctions (AND)

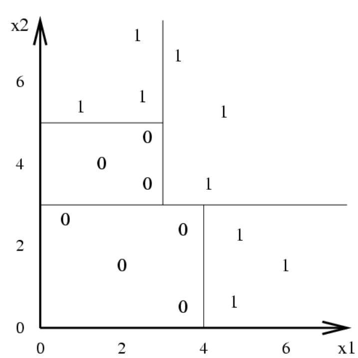
Decision Tree Classifier

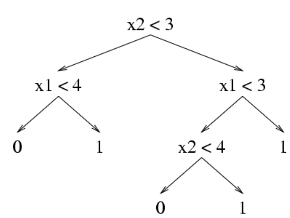
- If the features are continuous, internal nodes may test the value of a feature against a threshold



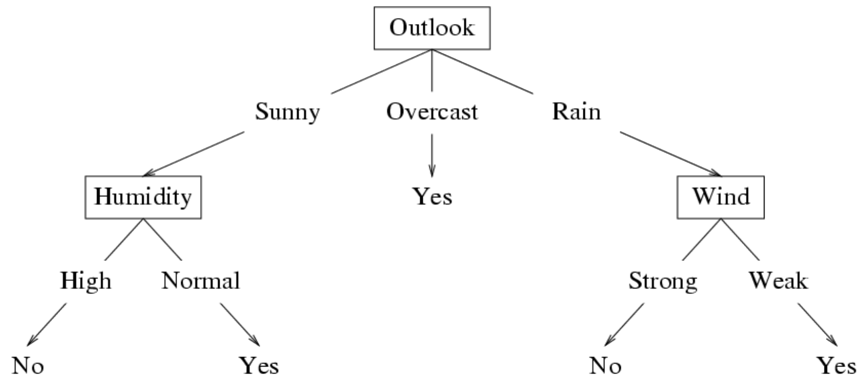
Decision Tree Classifier

- Learns axis-parallel decision boundaries, i.e. divides feature space into hyper-rectangles





Learning a Decision Tree



Learning a Decision Tree

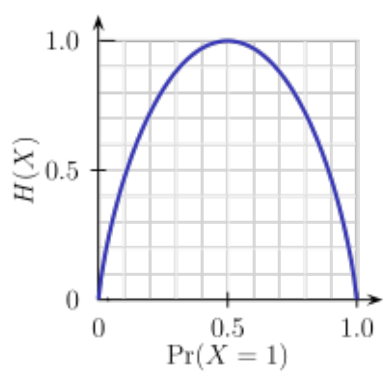
```
function DECISION-TREE-LEARNING (examples, attributes, parents) returns a tree
    if examples is empty return MAJORITY_VOTE(parents)
    else if all examples have same label return label
    else if attributes is empty return MAJORITY_VOTE(examples)
    else
        A     ←——  CHOOSE-BEST-ATTRIBUTE (examples)
        tree  ←——  a new decision tree with root A
        for each value v_k of A
            S_k      ←——  examples with value v_k for attribute A
            subtree  ←——  DECISION-TREE-LEARNING(S_k, attributes-A, examples)
            add branch to tree with label (A=v_k) and subtree
        return tree
```

Choosing the best attribute

- Splitting on a good attribute
  - After the split, the examples at each branch have the same label
- Splitting on a bad attribute
  - After the split, the examples at each branch have the same proportion of positive and negative labels
- We will use entropy and information gain to formalize what we mean by *good* and *bad* attributes

Entropy

- Entropy measures the uncertainty of a random variable
  - How many bits are needed to efficiently encode the possible values (outcomes) of a random variable?
- Introduced by Shannon in 1948 paper
- Example: flipping a coin
  - A completely biased coin requires 0 bits of entropy
  - A fair coin requires 1 bit of entropy
  - How many bits are need to encode the outcome of flipping a fair coin twice?



Entropy and Information Gain

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

Day	Outlook	Temp.	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