Concept Learning

Decision Trees

Concept Learning

Concept Learning

Learning of a boolean function from examples

Categories

- "Nice weather"
- "Dog"
- "Motor vehicle"
- "Criminal offence"

Subsets of a superset X

Terminology

c The concept to learn

$$c(x) \rightarrow \text{True/False}, \quad x \in X$$

h Hypothesis, Result of the learning ("guessed *c*")

$$h(x) \rightarrow \text{True/False}, \quad x \in X$$

H Hypotheses space , All conceivable hypotheses (before data arrives)

$$h \in H$$

D Set of available training data

$$D \subseteq X$$

Terminology

Two kinds of training examples

Positive example:

$$x : c(x) = \text{True}, \quad x \in D$$

Negative example:

$$x : c(x) = \text{False}, \quad x \in D$$

Example of a *concept*

"*Nice Weather*"

Let each "weather instance" x_i be composed of four attributes:

x_1 = <Sunny, Warm, Windy, Dry>
x_2 = <Cloudy, Warm, Calm, Dry>
x_3 = ...

Generally: *Sky* × *Temperature* × *Wind* × *Humidity*

Assume that the attributes can only take on certain discrete values:

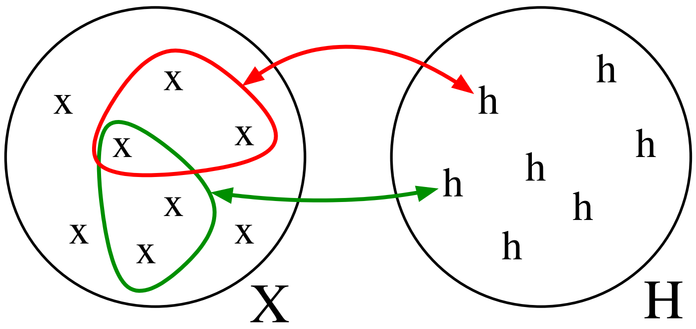
Sky ∈ { Sunny, Cloudy, Rainy }
Temp ∈ { Warm, Cold }
Wind ∈ { Windy, Calm }
Humid ∈ { Humid, Dry }

Number of possible weathers: $|X| = 3 \cdot 2 \cdot 2 \cdot 2 = 24$

Typical training samples

$x_1 =$	<Sunny, Warm, Windy, Dry>	$\rightarrow$ Nice
$x_2 =$	<Sunny, Warm, Windy, Humid>	$\rightarrow$ Nice
$x_3 =$	<Rainy, Cold, Windy, Humid>	$\rightarrow$ Bad
$x_4 =$	<Sunny, Warm, Calm, Humid>	$\rightarrow$ Nice

What does the hypotheses space H look like?



Each hypothesis h corresponds to one subset of X

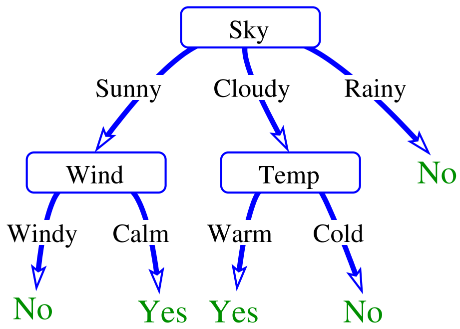
How many hypotheses can we choose from?
How many subsets does X have?

$$|H| = 2^{|X|}$$

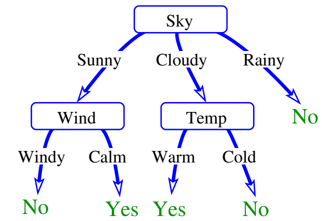
$$|H| = 2^{24} = 16777216$$

Decision Tree

- Test the attributes sequentially
- Choose attributes to test depending on earlier attribute values



The results (classifications) are coded by the *leaves*



What does the tree encode?

$$(\text{Sunny} \wedge \text{Calm}) \vee (\text{Cloudy} \wedge \text{Warm})$$

Works as a *disjunction of conjunctions*

Normal Form for boolean functions
Arbitrary boolean functions can be represented!

How can a decision tree be constructed automatically?

1. Choose an attribute to test
2. Branches with a unique class become leaves
3. Other branches are extended recursively

Remaining question: how do we choose attributes?

Greedy approach:

Choose the attribute which *tells us most* about the answer

Entropy

Entropy — measure of unpredictability

$$\text{Entropy} = \sum_i -p_i \log_2 p_i$$

p_i probability for outcome i

Entropy

Example: tossing a coin
$p_{\text{head}} = 0.5$; $p_{\text{tail}} = 0.5$

$$\text{Entropy} = \sum_i -p_i \log_2 p_i =$$
$$= -0.5 \log_2 0.5 + -0.5 \log_2 0.5 = -0.5 \underbrace{\log_2 0.5}_{-1} + -0.5 \underbrace{\log_2 0.5}_{-1} =$$
$$= 1$$

The result of a coin-toss has 1 bit of information

Entropy

Example: rolling a dice
$p_1 = \frac{1}{6}$; $p_2 = \frac{1}{6}; \ldots$ $p_6 = \frac{1}{6}$

$$\text{Entropy} = \sum_i -p_i \log_2 p_i =$$
$$= 6 \times -\frac{1}{6} \log_2 \frac{1}{6} =$$
$$= -\log_2 \frac{1}{6} = \log_2 6 \approx 2.58$$

The result of a dice-roll has 2.58 bit of information

Entropi

Example: rolling a fake dice
$p_1 = 0.1; \ldots$ $p_5 = 0.1$; $p_6 = 0.5$

$$\text{Entropy} = \sum_i -p_i \log_2 p_i =$$
$$= -5 \cdot 0.1 \log_2 0.1 - 0.5 \log_2 0.5 =$$
$$\approx 2.16$$

A real dice is more unpredictable (2.58 bit) than a fake (2.16 bit)

Entropy

Unpredictability of a dataset

- 100 examples, 42 positive

$$-\frac{58}{100}\log_2\frac{58}{100}-\frac{42}{100}\log_2\frac{42}{100}=0.981$$

- 100 examples, 3 positive

$$-\frac{97}{100}\log_2\frac{97}{100}-\frac{3}{100}\log_2\frac{3}{100}=0.194$$

Back to the decision trees

Smart idea:

Ask about the attribute which maximizes the expected reduction of the entropy.

Information Gain

Assume that we ask about attribute A for a dataset S

$$\text{Gain} = \text{Ent}(S)\underbrace{\text{Ent}(S)}_{\text{before}} - \sum_{v\in\text{Values}(A)}\frac{|S_v|}{|S|}\underbrace{\sum_{v\in\text{Values}(A)}\frac{|S_v|}{|S|}}_{\substack{\text{weighted}\\\text{average}}}\text{Ent}(S_v)\underbrace{\text{Ent}(S_v)}_{\text{after}}$$

What is the entropy for this dataset?

$\text{Ent} = -\frac{12}{25}\log_2\frac{12}{25}-\frac{13}{25}\log_2\frac{13}{25}\approx\mathbf{0.9988}$

$A=\bullet$: $\frac{6}{12}$ positive $\rightarrow 1.0$
$A=\circ$: $\frac{6}{13}$ positive $\rightarrow 0.9957$
Expected: $\frac{12}{25}\cdot 1.0+\frac{13}{25}\cdot 0.9957\approx\mathbf{0.9977}$

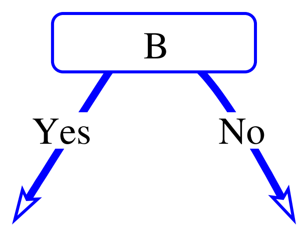
$B=\bullet$: $\frac{9}{11}$ positive $\rightarrow 0.684$
$B=\circ$: $\frac{3}{14}$ positive $\rightarrow 0.750$
Expected: **0.721**

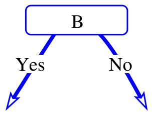
$C=\bullet$: $\frac{3}{6}$ positive $\rightarrow 1.0$
$C=\circ$: $\frac{9}{19}$ positive $\rightarrow 0.9980$
Expected: **0.9985**

$D=\bullet$: $\frac{3}{5}$ positive $\rightarrow 0.9710$
$D=\circ$: $\frac{9}{20}$ positive $\rightarrow 0.9928$
Expected: **0.9884**

A	B	C	D	
●	●	○	○	+
○	●	●	○	+
○	○	○	○	
●	○	○	●	+
○	●	○	○	+
●	○	●	○	
●	●	○	○	+
○	○	○	○	
○	○	●	○	
●	●	○	○	+
○	○	○	●	+
●	○	○	○	
●	●	●	○	+
○	●	○	●	
○	○	○	○	
●	○	○	○	
●	●	○	●	
○	●	○	○	+
○	○	●	○	
●	○	○	○	
○	●	○	○	+
●	○	○	●	+
○	●	●	○	+
○	○	○	○	
●	○	○	○	

$\text{Gain}(A) = 0.9988 - 0.9977 = \mathbf{0.0011}$
$\text{Gain}(B) = 0.9988 - 0.7210 = \mathbf{0.2778}$
$\text{Gain}(C) = 0.9988 - 0.9985 = \mathbf{0.0003}$
$\text{Gain}(D) = 0.9988 - 0.9884 = \mathbf{0.0104}$
Attribute B gives most information



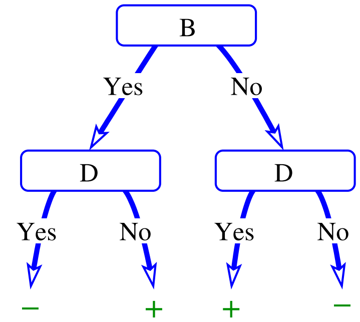


Examples where
$B = \bullet$

A	B	C	D	
●	●	○	○	+
○	●	●	○	+
○	●	○	○	+
●	●	○	○	+
●	●	○	○	+
●	●	●	○	+
○	●	○	●	
●	●	○	●	
○	●	○	○	+
○	●	○	○	+
○	●	●	○	+

Examples where
$B = \circ$

A	B	C	D	
○	○	○	○	
●	○	○	●	+
●	○	●	○	
○	○	○	○	
○	○	●	○	
○	○	○	●	+
●	○	○	○	
○	○	○	○	
●	○	○	○	
○	○	●	○	
●	○	○	○	
●	○	○	●	+
○	○	○	○	
●	○	○	○	



Possible ways of improving the decision trees

- Avoid overfitting
  - Limit the tree's height
  - Pruning
- Attributes with graded values
- Missing attribute values
- Variable cost for different attributes