

# Lecture 2: Decision Trees

## DD2421

Atsuto Maki

Autumn, 2021

- Lecture 1: Nearest Neighbour Classifier (Memory-based)
- Lecture 2: **Decision Trees** (Logical inference, Rule-based)
- Lecture 3: Challenges in Machine Learning

## 1 Decision Trees

- The representation
- Training

## 2 Unpredictability

how to automatically  
build the tree?

- Entropy
- Information gain
- Gini impurity

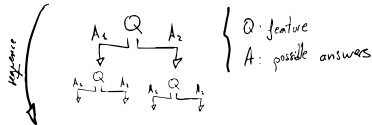
## 3 Overfitting

build a tree able  
to generalize not  
adapt to new data!

- Overfitting
- Occam's principle
- Training and validation set approach
- Extensions

- 1 Decision Trees
  - The representation
  - Training
- 2 Unpredictability
  - Entropy
  - Information gain
  - Gini impurity
- 3 Overfitting
  - Overfitting
  - Occam's principle
  - Training and validation set approach
  - Extensions

Basic Idea: Test the attributes (features) sequentially  
= Ask questions about the target/status sequentially



Q & A about the target  
↓  
until you can reach a conclusion  
about the target and make a prediction

Basic Idea: Test the attributes (features) **sequentially**  
= Ask questions about the target/status **sequentially**

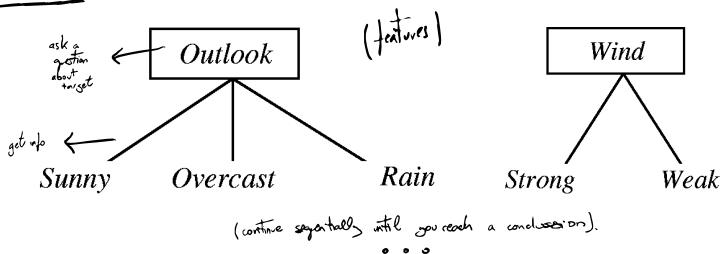
Example: building a concept of whether someone would like to play tennis.

e.g:

$$\begin{aligned} \text{target: } & \text{play tennis} \in \{Y, N\} \\ \text{features: } & Q \begin{cases} \text{weather} \leftarrow A \in \{\text{sunny}, \text{cloudy}\} \\ \text{wind} \leftarrow A \in \{Y, N\} \end{cases} \end{aligned}$$

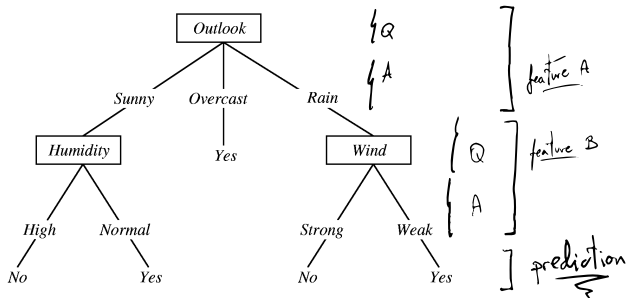
Basic Idea: Test the attributes (features) **sequentially**  
= Ask questions about the target/status **sequentially**

Example: building a concept of whether someone would like to play tennis.



Useful also (but not limited to) when nominal data are involved, e.g. in medical diagnosis, credit risk analysis etc.

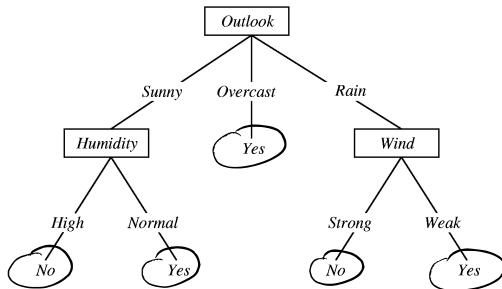
The whole analysis strategy can be seen as a tree.



(T. Mitchell, Machine Learning)

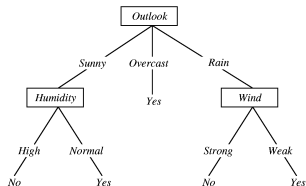


The whole analysis strategy can be seen as a tree.

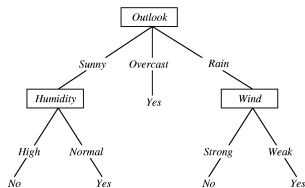


(T. Mitchell, Machine Learning)

Each **leaf node** bears a category label, and the test pattern is assigned the category of the leaf node reached.

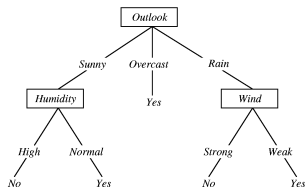


What does the tree encode?



What does the **tree encode**?  $\Rightarrow$  logical inference  $\rightarrow$  [OR and AND statements]

$(\text{Sunny} \wedge \text{Normal Humidity}) \vee (\text{Cloudy}) \vee (\text{Rainy} \wedge \text{Weak Wind})$



What does the tree encode?

$(\text{Sunny} \wedge \text{Normal Humidity}) \vee (\text{Cloudy}) \vee (\text{Rainy} \wedge \text{Weak Wind})$

Logical expressions of the conjunction of decisions along the path.

Arbitrary boolean functions can be represented!

**Training**: we need to grow a tree from scratch given a set of labeled training data.

How to grow/construct the tree automatically?

Training: we need to grow a tree from scratch given a set of labeled training data.

How to grow/construct the tree automatically?

- 1 Choose the best question (according to the information gain), and split the input data into subsets

↳ the question must give information about the dataset  
Split should be done s.t. data is essentially more readily classifiable.

Training: we need to grow a tree from scratch given a set of labeled training data.

How to grow/construct the tree automatically?

- 1 Choose the **best question** (according to the **information gain**), and split the input data into subsets
- 2 **Terminate**: call branches with a unique class labels **leaves** (no need for further questions)

any path must lead to a final leave where a label can be finally assigned.

Training: we need to grow a tree from scratch given a set of labeled training data.

How to grow/construct the tree automatically?

- 1 Choose the **best question** (according to the **information gain**), and split the input data into subsets
- 2 **Terminate**: call branches with a unique class labels **leaves** (no need for further questions)
- 3 **Grow**: recursively extend other branches (with subsets bearing mixtures of labels)



## 1 Decision Trees

- The representation
- Training

## 2 Unpredictability

- Entropy
- Information gain
- Gini impurity

## 3 Overfitting

- Overfitting
- Occam's principle
- Training and validation set approach
- Extensions

Quiz time – Game of “sixty-three”

$x$  drawn from  $\{0, 1, 2, 3, 4, \dots, 63\}$

- I pick a number  $x$  from the set.
- You ask me yes/no questions.

How many (and what) questions will you ask me to get the number  $x$  as rapidly as possible?

# Entropy

How to measure **information gain**?

$$x = \{0-63\}$$

Q&A:

$x \geq 32$
$x \geq 16$
$x \geq 8$
$x \geq 4$
$x \geq 2$
$x \geq 1$

"how many bits are needed?"

↳ feature space is sequentially divided  
→ this example assumes all integers  
0-63 are equally likely

# Entropy

How to **measure information gain**?

The Shannon information content of an outcome is:

$$\log_2 \frac{1}{p_i}$$

(information measurement  
of an event)

( $p_i$  : probability for event  $i$ )

# Entropy

How to measure **information gain**?

The Shannon information content of an outcome is:

$$\log_2 \frac{1}{p_i}$$

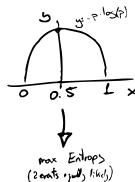
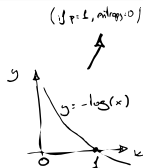
( $p_i$  : probability for event  $i$ )

The **Entropy** — measure of uncertainty (unpredictability)

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

← *average*

is a sensible measure of expected information content.



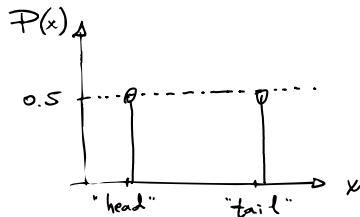
# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$

(probability is  
equally distributed)

Entropy is maximal!



# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$



$$\underline{\text{Entropy}} = \sum_i -p_i \log_2 p_i \quad \rightarrow \text{smaller when biased}$$

i.e. some events are more likely!

# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= \underbrace{-0.5}_{p_1} \log_2 \underbrace{0.5}_{p_1} - \underbrace{0.5}_{p_2} \log_2 \underbrace{0.5}_{p_2} \quad \left. \begin{array}{l} \text{(sum of the two} \\ \text{possible events)} \end{array} \right\} : \Sigma \end{aligned}$$



# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= -0.5 \underbrace{\log_2 0.5}_{-1} - 0.5 \underbrace{\log_2 0.5}_{-1} \end{aligned}$$

# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$



$$\begin{aligned}\text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= -0.5 \underbrace{\log_2 0.5}_{-1} - 0.5 \underbrace{\log_2 0.5}_{-1} = \\ &= 1\end{aligned}$$

→ max Entropy because both events have equal probability

# Entropy

Example: tossing a coin

$$p_{\text{head}} = 0.5; \quad p_{\text{tail}} = 0.5$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= -0.5 \underbrace{\log_2 0.5}_{-1} - 0.5 \underbrace{\log_2 0.5}_{-1} = \\ &= 1 \end{aligned}$$

(1 bit =  $\log_2 2 = \log_2 \{\text{head, tails}\}$ ).

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

# Entropy

Example: rolling a die

$$p_1 = \frac{1}{6}; \quad p_2 = \frac{1}{6}; \dots \quad p_6 = \frac{1}{6}$$

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



(six possible outcomes  
with equal probability).

# Entropy

Example: rolling a die

$$p_1 = \frac{1}{6}; \quad p_2 = \frac{1}{6}; \dots \quad p_6 = \frac{1}{6}$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= 6 \times \left(-\frac{1}{6} \log_2 \frac{1}{6}\right) \end{aligned}$$

# Entropy

Example: rolling a die

$$p_1 = \frac{1}{6}; \quad p_2 = \frac{1}{6}; \dots \quad p_6 = \frac{1}{6}$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= 6 \times \left(-\frac{1}{6} \log_2 \frac{1}{6}\right) = \\ &= -\log_2 \frac{1}{6} = \log_2 6 \approx 2.58 \end{aligned}$$

# Entropy

Example: rolling a die

$$p_1 = \frac{1}{6}; \quad p_2 = \frac{1}{6}; \dots \quad p_6 = \frac{1}{6}$$



$$\begin{aligned} \text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= 6 \times \left(-\frac{1}{6} \log_2 \frac{1}{6}\right) = \\ &= -\log_2 \frac{1}{6} = \log_2 6 \approx 2.58 \end{aligned}$$

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

↗ (3 bits are needed)  
 $\begin{pmatrix} 1^1 & 1^2 \\ 2^1 & 2^2 \\ 3^1 & 3^2 \end{pmatrix}$

# Entropy

Example: rolling a fake die

$p_1 = 0.1; \dots p_5 = 0.1; p_6 = 0.5$

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



↓  
(8x possible outcomes  
with different probabilities)



# Entropy

Example: rolling a **fake die**

$$p_1 = 0.1; \dots \quad p_5 = 0.1; \quad p_6 = 0.5$$



$$\begin{aligned}\text{Entropy} &= \sum_i -p_i \log_2 p_i = \\ &= -5 \cdot 0.1 \log_2 0.1 - 0.5 \log_2 0.5\end{aligned}$$

# Entropy

Example: rolling a **fake die**

$$p_1 = 0.1; \dots \quad p_5 = 0.1; \quad p_6 = 0.5$$



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

# Entropy

Example: rolling a **fake die**

$$p_1 = 0.1; \dots \quad p_5 = 0.1; \quad p_6 = 0.5$$



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

these numbers are more likely, so it becomes more predictable, i.e. less info is needed to guess.

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

# Entropy

## Unpredictability of a dataset

- 100 examples, 42 positive

} more equally distributed  
 $\{ \frac{42}{100}, \frac{58}{100} \}$

↑ entropy

- 100 examples, 3 positive

} very unequally distributed  
 $\Rightarrow$  one clear winner/underdog  
 $\frac{97}{100}$   $\frac{3}{100}$

↓ entropy

# Entropy

Unpredictability of a **dataset**

- 100 examples, 42 positive

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

(almost  
1 bit)  
↑ (almost  
50:50)

- 100 examples, 3 positive

# 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

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

almost zero!  
almost just one  
possible outcome

# Entropy

Unpredictability of a **dataset** (think of a subset at a node)

- 100 examples, 42 positive

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

easy to split into 2 or are there two decisions? / very "tie"!

- 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.



search the attribute that  
minimizes entropy  $\nabla$

↳ (ask a question whose answer  
leads to a very clear split of data)



Back to the decision trees

Smart idea:

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

Information gain

Ask about attribute A for a data set S that has Entropy  $\text{Ent}(S)$ ,

$$\text{Gain} = \underbrace{\text{Ent}(S)}_{\text{before}} -$$

## Back to the decision trees

## Smart idea:

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

## Information gain

Ask about attribute  $A$  for a data set  $S$  that has Entropy  $\text{Ent}(S)$ , and get subsets  $S_v$  according to the value of  $A$

$$\text{Gain} = \underbrace{\text{Ent}(S)}_{\substack{\text{before} \\ \text{entropy of the whole dataset}}} - \underbrace{\sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Ent}(S_v)}_{\substack{\text{weighted sum} \\ \text{after}}} \quad \begin{array}{l} \text{entropy of all} \\ \text{sub sets} \\ \text{(after split)} \end{array}$$

↳ is entropy reduced after split?  $\Rightarrow$  take attribute leading to highest gain.

What is the entropy of this binary dataset (attributes= $\{A, B, C, D\}$ ,  $n = 25$ )?

for black dots

$A: 3/6$   
 $B: 9/11$   
 $C: 6/12$   
 $D: 3/5$

eg.  $\left( A: \frac{3 \text{ positives}}{6 \text{ events}} \right)$

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

Q: is  $\{A, B, C, D\}$  feature "+"?

A: Black/White

target: "+", non "+".

↳ which feature is the best question?

What is the entropy of this binary dataset (attributes= $\{A, B, C, D\}$ ,  $n = 25$ )?

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

$\underbrace{\quad}_{\text{"+" entropy}} \quad \underbrace{\quad}_{\text{"-+" entropy}}$

↳ which of the four features  $\{A, B, C, D\}$   
is the best split?

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

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

$A = \bullet: \frac{3}{6}$  positive  $\rightarrow 1.0$  (entirely + & for 50.50 ans.)

$$A = 0: \frac{9}{19} \text{ positive} \rightarrow 0.9980 \text{ (entropy for white dots).}$$

Expected:  $\frac{6}{25} \cdot 1.0 + \frac{19}{25} \cdot 0.9980 \approx \mathbf{0.9985}$

weighted entropy for  
black class

weighted entropy  
for white class.

average entropy for  
binary classification problem.

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

What is the entropy of this binary dataset (attributes= $\{A, B, C, D\}$ ,  $n = 25$ )?

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

$$A = \bullet: \frac{3}{6} \text{ positive} \rightarrow 1.0$$

$$A = \circ: \frac{9}{19} \text{ positive} \rightarrow 0.9980$$

$$\text{Expected: } \frac{6}{25} \cdot 1.0 + \frac{19}{25} \cdot 0.9980 \approx \mathbf{0.9985}$$

$$B = \bullet: \frac{9}{11} \text{ positive} \rightarrow 0.684$$

$$B = \circ: \frac{3}{14} \text{ positive} \rightarrow 0.750$$

$$\text{Expected: } \mathbf{0.721}$$

*entropy is really reduced when feature B is used for splitting.*

*almost equal at top*

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

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

$A = \bullet: \frac{3}{6} \text{ positive} \rightarrow 1.0$

$$A = \circ: \frac{9}{19} \text{ positive} \rightarrow 0.9980$$

Expected:  $\frac{6}{25} \cdot 1.0 + \frac{19}{25} \cdot 0.9980 \approx \mathbf{0.9985}$

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

Expected: **0.721**

$$C = \bullet: \frac{6}{12} \text{ positive} \rightarrow 1.0$$
$$C = \circ: \frac{6}{13} \text{ positive} \rightarrow 0.9957$$

Expected: **0.9977**

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

What is the entropy of this binary dataset (attributes= $\{A, B, C, D\}$ ,  $n = 25$ )?

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

$A = \bullet: \frac{3}{6} \text{ positive} \rightarrow 1.0$

$$A = \circ: \frac{9}{19} \text{ positive} \rightarrow 0.9980$$

Expected:  $\frac{6}{25} \cdot 1.0 + \frac{19}{25} \cdot 0.9980 \approx \mathbf{0.9985}$

$$B = \bullet: \frac{9}{11} \text{ positive} \rightarrow 0.684$$

$$B = \circ: \frac{3}{14} \text{ positive} \rightarrow 0.750$$

Expected: **0.721**

$$C = \bullet: \frac{6}{12} \text{ positive} \rightarrow 1.0$$

$$C = \circ: \frac{6}{13} \text{ positive} \rightarrow 0.9957$$

Expected: **0.9977**

$$D = \bullet: \frac{3}{5} \text{ positive} \rightarrow 0.9710$$

$$D = \circ: \frac{9}{20} \text{ positive} \rightarrow 0.9928$$

Expected: **0.9884**

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





$$\text{Gain}(A) = 0.9988 - 0.9985 = \mathbf{0.0003}$$

$$\text{Gain}(B) = 0.9988 - 0.7210 = \mathbf{0.2778}$$

$$\text{Gain}(C) = 0.9988 - 0.9977 = \mathbf{0.0011}$$

$$\text{Gain}(D) = 0.9988 - 0.9884 = \mathbf{0.0104}$$

 total entropy

 with each  
feature being  
used to split

$$\text{Gain}(A) = 0.9988 - 0.9985 = \mathbf{0.0003}$$

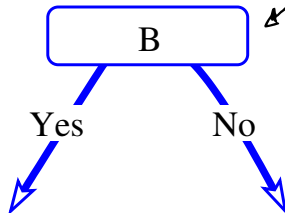
$$\text{Gain}(B) = 0.9988 - 0.7210 = \mathbf{0.2778}$$

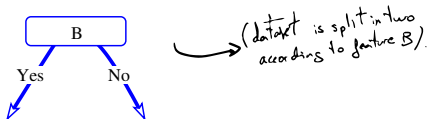
$$\text{Gain}(C) = 0.9988 - 0.9977 = \mathbf{0.0011}$$

$$\text{Gain}(D) = 0.9988 - 0.9884 = \mathbf{0.0104}$$

Attribute B gives most information gain

*feature B is the  
chosen for creating  
a "splitting" node*



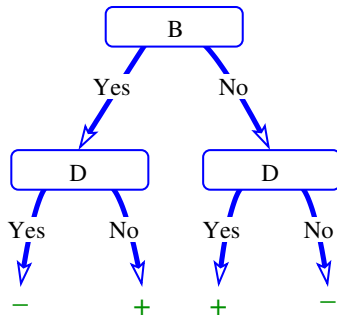


Examples where  
 $B = \bullet$

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

Examples where  
 $B = \circ$

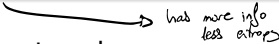
A	B	C	D	
○	○	○	○	
○	○	●	●	+
●	○	●	○	
○	○	○	○	
●	○	○	○	
○	○	○	●	+
○	○	●	○	
○	○	○	○	
○	○	●		
○	○	○	○	
●	○	○	○	
○	○	●	○	
○	○	●	●	+
○	○	○	○	
○	○	●	○	



Greedy approach to choose a question:

Choose the attribute which tells us most about the answer

In sum, we need to find good questions to ask.  
(more than one attribute could be involved in one question)



**Gini impurity**: Another definition of predictability (impurity).

$$\sum_i p_i(1 - p_i) = 1 - \sum_i p_i^2$$

( $p_i$  : probability for event  $i$ )

binary problem!

Gini impurity: Another definition of predictability (impurity).

$$\sum_i p_i(1 - p_i) = 1 - \sum_i p_i^2$$

( $p_i$  : probability for event  $i$ )

The expected error rate at a node,  $N$ , if the category label is randomly selected from the class distribution present at  $N$ .

Gini impurity: Another definition of predictability (impurity).

$$\sum_i p_i(1 - p_i) = 1 - \sum_i p_i^2$$

( $p_i$  : probability for event  $i$ )

The expected error rate at a node,  $N$ , if the category label is randomly selected from the class distribution present at  $N$ .

Similar to the entropy but more strongly peaked at equal probabilities.





## 1 Decision Trees

- The representation
- Training

## 2 Unpredictability

- Entropy
- Information gain
- Gini impurity

## 3 **Overfitting**

- Overfitting
- Occam's principle
- Training and validation set approach
- Extensions

## Overfitting

When the learned models are overly specialized for the training  
samples.

## Overfitting

When the learned models are overly specialized for the training samples.

Good results on training data, but generalizes poorly.

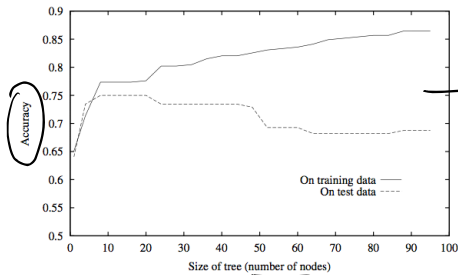
## Overfitting

When the learned models are overly specialized for the training samples.

Good results on training data, but generalizes poorly.

When does this occur?

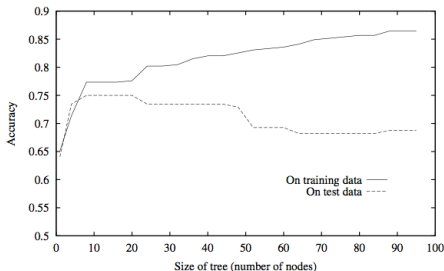
- Non-representative sample
- Noisy examples
- Too complex model



increasing nodes  
might lead to overfitting.

(T. Mitchell, Machine Learning)

What can be done about it?



(T. Mitchell, Machine Learning)

What can be done about it?

Choose a simpler model and accept some errors for the training examples

→ that adapts better to new samples (test data).

Which hypothesis should be preferred when several are compatible with the data?

Which hypothesis should be preferred when several are compatible with the data?

Occam's principle (Occam's razor)

William from Ockham, Theologian and Philosopher  
(1288–1348)

"Entities should not be multiplied beyond necessity"

the simplest explanation  
is the best.



Which hypothesis should be preferred when several are compatible with the data?

**Occam's principle** (Occam's razor)

William from Ockham, Theologian and Philosopher  
(1288–1348)

"Entities should not be multiplied beyond necessity"

The simplest explanation compatible with data  
tends to be the right one

Separate the available data into two sets of examples

- Training set  $T$ : to form the learned model
- Validation set  $V$ : to evaluate the accuracy of this model

Separate the available data into two sets of examples

- *Training set*  $T$ : to form the learned model
- *Validation set*  $V$ : to evaluate the accuracy of this model

The motivations:

- The training may be misled by random errors, but the validation set is unlikely to exhibit the same random fluctuations
- The validation set to provide a safety check against overfitting the spurious characteristics of the training set

Separate the available data into two sets of examples

- *Training set*  $T$ : to form the learned model
- *Validation set*  $V$ : to evaluate the accuracy of this model

The motivations:

- The training may be misled by random errors, but the validation set is unlikely to exhibit the same random fluctuations
- The validation set to provide a safety check against overfitting the spurious characteristics of the training set

( $V$  need be large enough to provide statistically meaningful instances)

# Reduced-Error Pruning

Split data into training and validation set

Do until further pruning is harmful:

- Evaluate impact on validation set of pruning each possible node (plus those below it)
- Greedily remove the one that most improves validation set accuracy

(make tree smaller).

Produces smallest version of most accurate subtree

## Possible ways of improving/extending the decision trees

- Avoid overfitting
  - Stop growing when data split not statistically significant
  - Grow full tree, then post-prune (e.g. Reduced error pruning)

## Possible ways of improving/extending the decision trees

- Avoid overfitting
  - Stop growing when data split not statistically significant
  - Grow full tree, then post-prune (e.g. Reduced error pruning)

A collection of trees (Ensemble learning: in Lecture 10)

- Bootstrap aggregating (bagging)
- Decision Forests