Topic 2: Information-based learning

Part 1: Introduction

Learning objectives

After completing this topic successfully, you will be able to …

1. Explain what supervised learning is?
2. Distinguish it from unsupervised learning and reinforcement learning
3. Describe in detail an algorithm for decision tree induction
4. Apply decision tree induction to a data set
5. List related algorithms
6. Discuss high-level concepts such as choice of hypothesis language, overfitting, underfitting and noise

Reading: Russell & Norvig 3rd Ed, Chapter 18.18.4; Kelleher et al. Chapter 4

Overview of topic

This week:

1. Introduction, learning objectives and overview
2. Supervised learning principles
3. Decision trees
4. Entropy
5. Information gain

Next week:

6. The ID3 algorithm
7. Issues in decision tree learning
8. ID3 extensions and related algorithms
9. Supervised learning considerations
10. Review of topic

Topic 2: Information-based learning

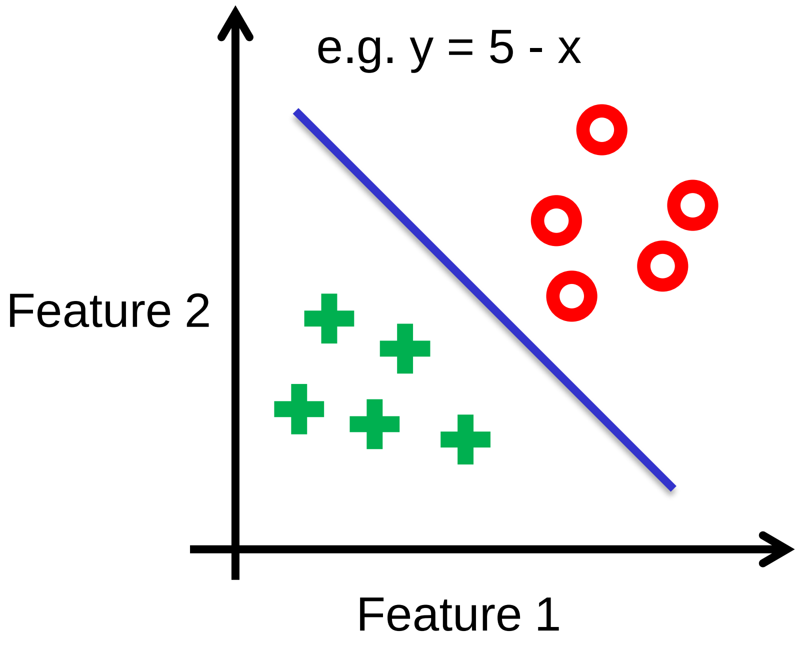
Part 2: Supervised learning principles

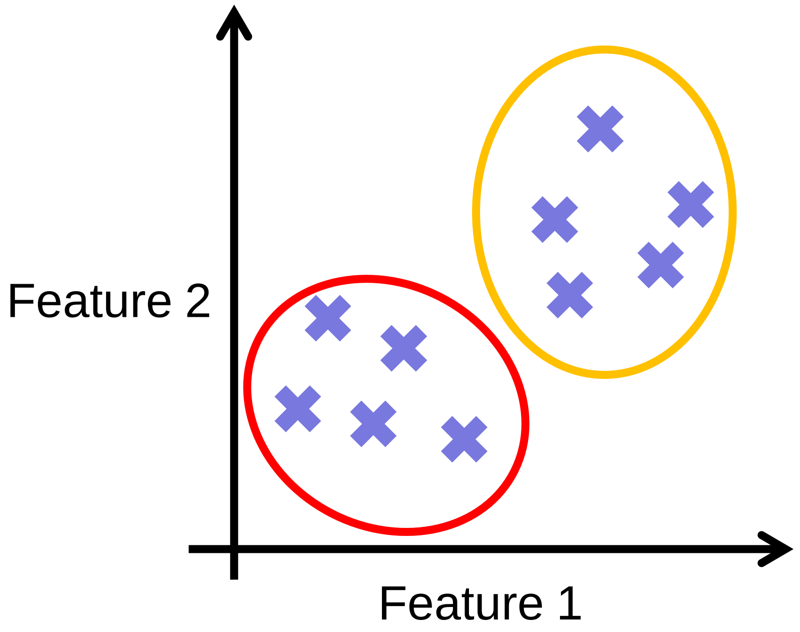
Supervised learning: motivating examples

1. Estimate sale price of a house, given past data of house sizes, locations and their prices
2. Before unlocking a tablet, determine whether a known user or somebody else is looking at the webcam
3. Decide whether a chemical spectrum of a mixture has evidence of containing cocaine, based on other spectra with & without cocaine
4. Predict concentration of cocaine in mixture
5. Determine whether objects of interest are present in a scene – if so, what are they? (relevant for autonomous vehicles and robotics, among other domains)

Key feature:
given "right answers" / "ground truth" as start point.
Which tasks are classification, and which are regression?

Supervised vs. unsupervised learning





Supervised learning: task definition

- Given examples, return function *h* (*hypothesis*) that approximates some 'true' function *f* that (hypothetically) generated the labels for the examples
  - Have set of examples, the **training data**:
    each has a **label** and a set of **attributes** that have known **values**
  - Consider *labels* (classes) to be *outputs* of some function *f*; the observed *attributes* are its *inputs*
  - Denote the attribute value inputs ***x*,** labels are their corresponding outputs ***f*(*x*)**
  - An example is a pair **(*x*, *f*(*x*))**
  - Function ***f*** is *unknown*; want to discover an approximation of it, ***h***
  - Can use ***h*** to **predict** labels of new data: generalisation

Also known as Pure Inductive Learning – why?

Training data example

Task Description

Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
	overcast	cool	normal	true	yes

Attributes/ Dimensions

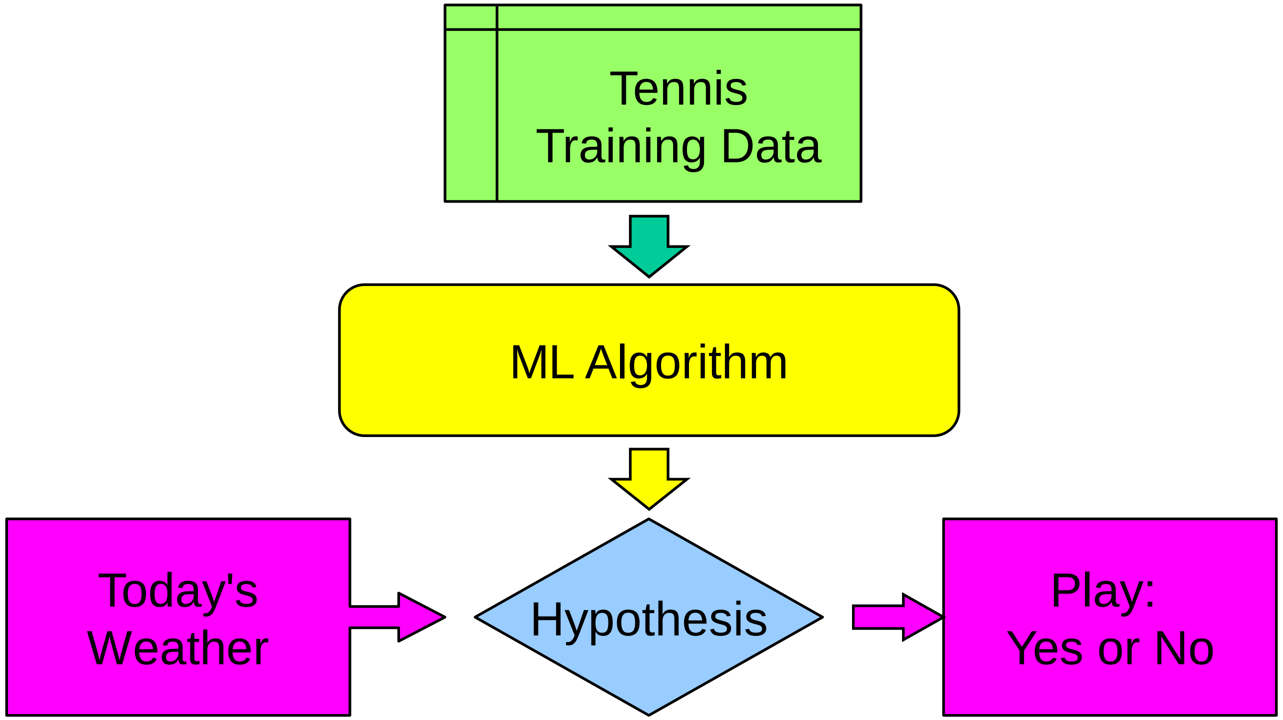
One Training Case

Identifier (not used in learning)

Attribute values = Independent variables

Labels/Classes = Dependent variables

Overview of the supervised learning process

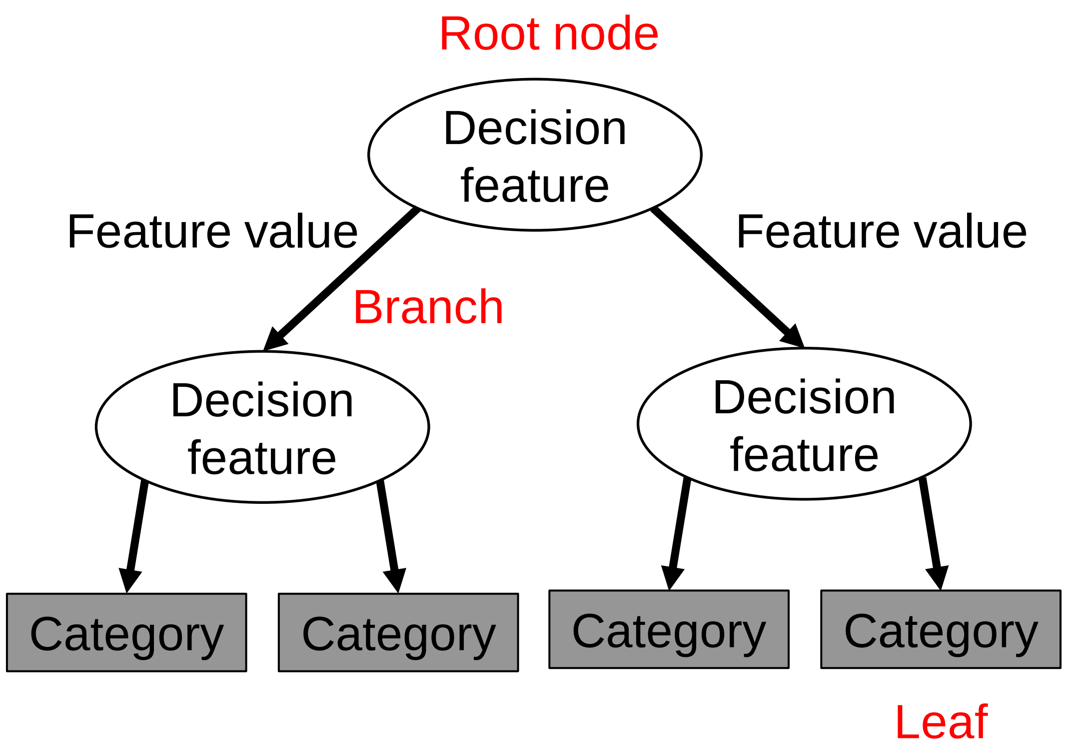


Topic 2: Information-based learning

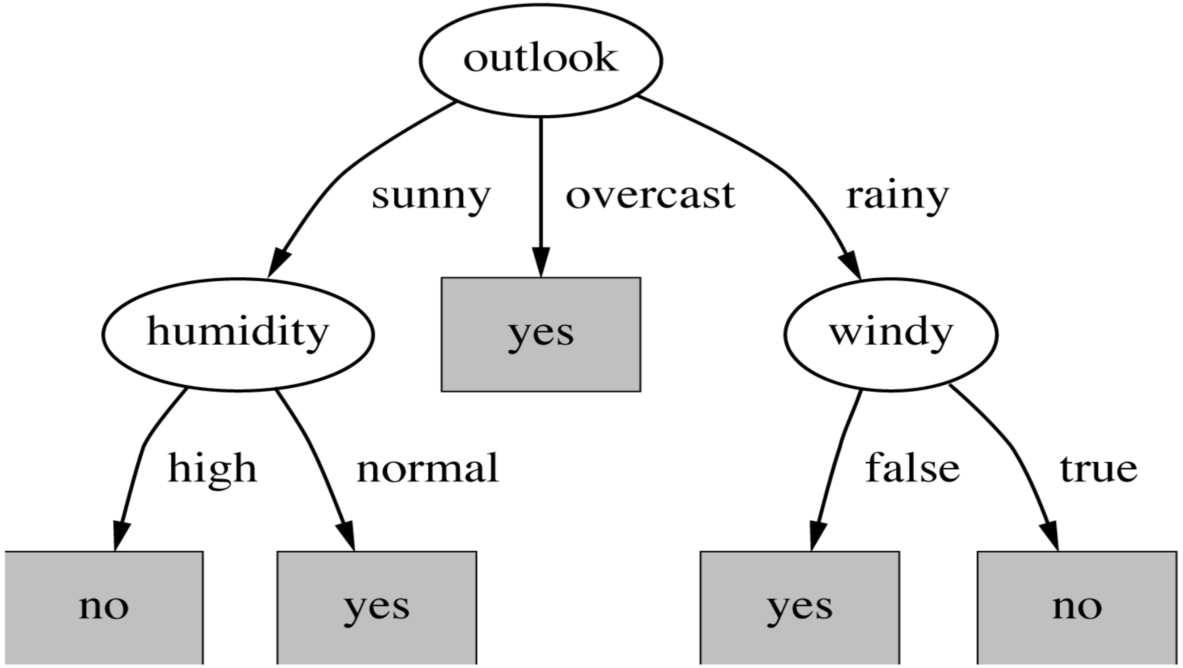
Part 3: Decision trees

Decision trees

- Decision trees are a fundamental structure used in information-based machine learning
- Main idea: use a decision tree as a predictive model, to decide what category/label/class an item belongs to based on the values of its features
- So-called due to their tree-like structure:
  - A node (where two branches intersect) is a decision point. Nodes partition the data.
  - observations about an item (values of features) are represented using branches
  - The terminal nodes are called leaves; these specify the target label for an item



Decision tree for a sample dataset



Example dataset for induction (1)

- Weather dataset
  - Four attributes:
    outlook: sunny / overcast / rainy
    temperature: hot / mild / cool
    humidity: high / normal
    windy: true / false
  - Used to decide whether or not to *play tennis*
  - 14 examples in dataset
  - See **weather.xls** (spreadsheet) or **weathertxt.csv** (comma separated values format)

- Objective:
  - Find hypothesis that *describes the cases* given
    and can be used to *make decisions* in other cases
  - Express the hypothesis as a decision tree.

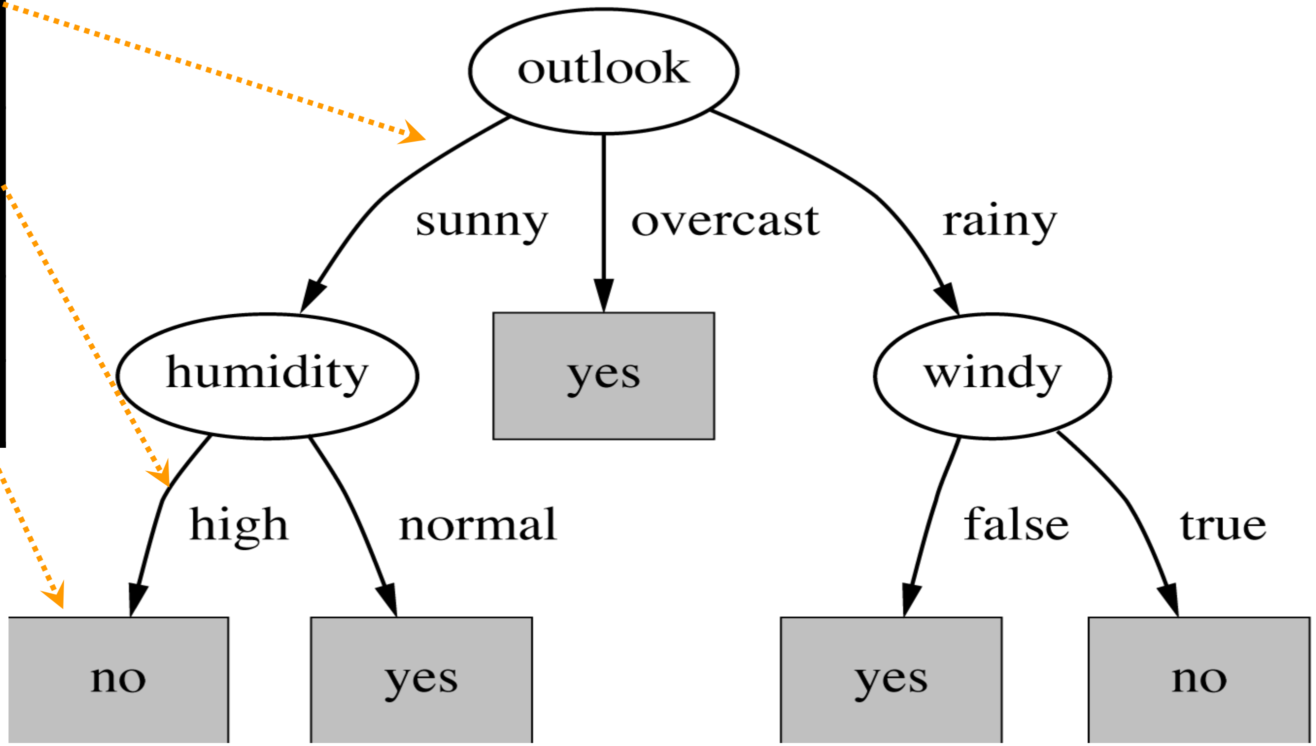
Example dataset for induction (2)

Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

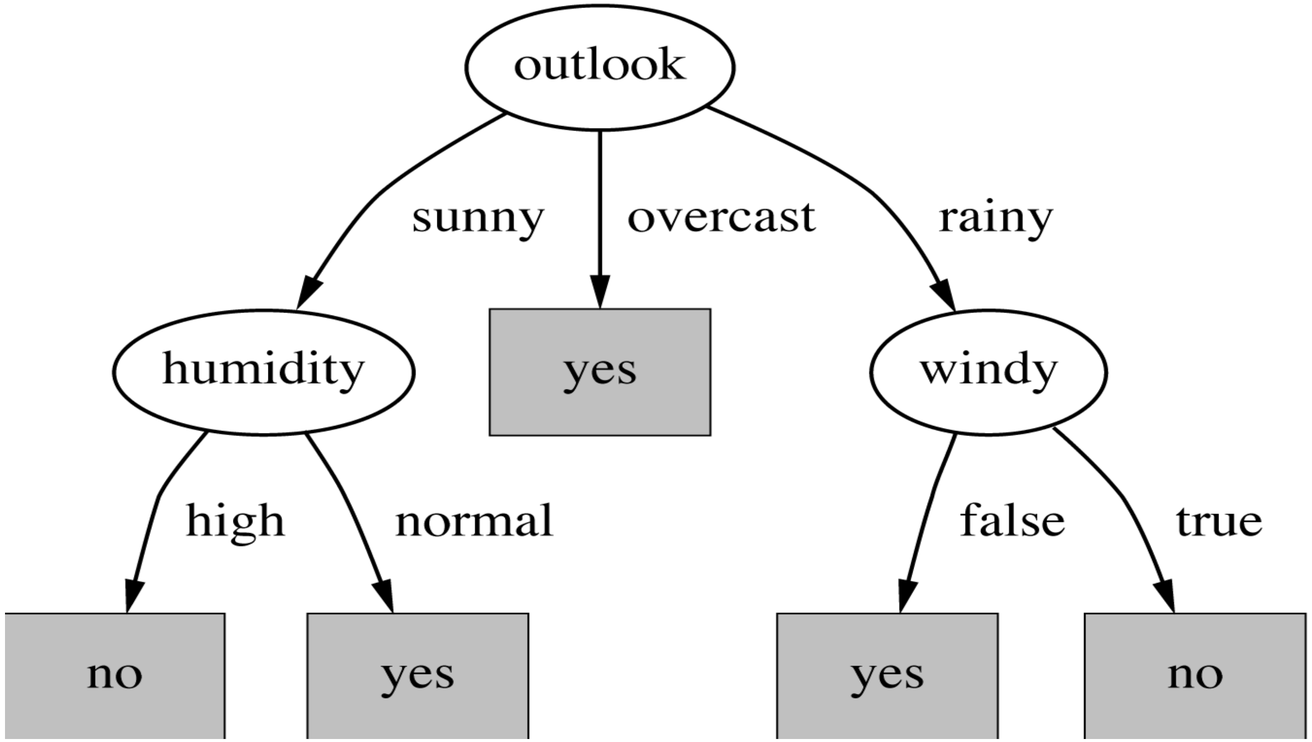
Decision tree for this data (1)

Outlook	sunny
Temp	hot
Humidity	high
Windy	false
Play?	no



Decision tree for this data (2)

Anyone for Tennis?					
ID	**Outlook**	**Temp**	**Humidity**	**Windy**	**Play?**
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no



Inductive learning of a decision tree

Step 1
- For all attributes that have not yet been used in the tree, calculate their **entropy** and **information gain** values for the training samples

Step 2
- Select the attribute that has the highest information gain

Step 3
- Make a tree node containing that attribute

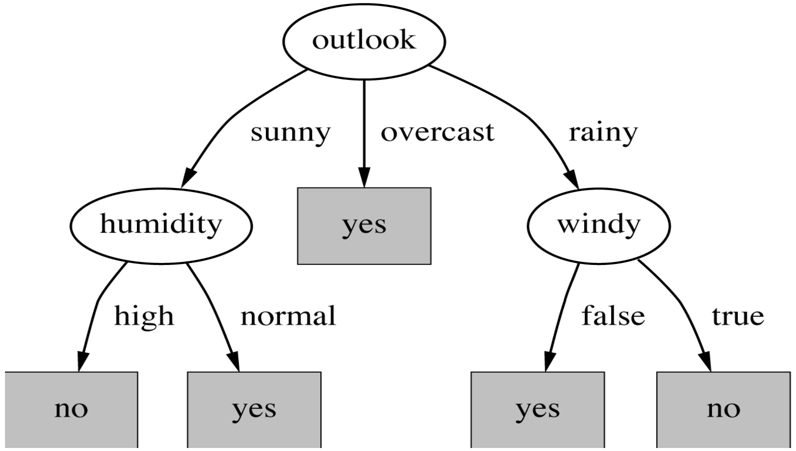
Repeat
- This node **partitions** the data:
  apply the algorithm **recursively** to each partition

Topic 2: Information-based learning

Part 4: Entropy

Motivation

- We already saw how some descriptive features can more effectively discriminate between (or predict) classes which are present in the dataset
- Decision trees partition the data at each node, so it makes sense to use features which have higher discriminatory power “higher up” in a decision tree.
- Therefore we need to develop a formal measure of the discriminatory power of a given attribute
- Information gain – this can be calculated using entropy

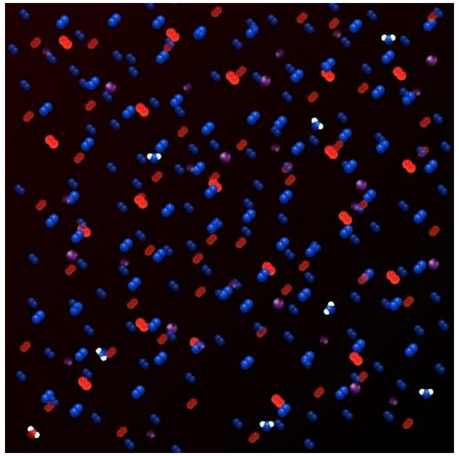


Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Entropy

- Claude Shannon (often referred to as “the father of information theory”) proposed a measure to of the impurity of the elements in a set, referred to as entropy
- Entropy may be used to measure of the uncertainty of a random variable
- The term entropy generally refers to disorder or uncertainty, so the use of this term in the context of information theory is analogous to the other well-known use of the term in statistical thermodynamics
- Acquisition of information (information gain) corresponds to a reduction in entropy
- “Information is the resolution of uncertainty” (Shannon)
- 1948 article “A Mathematical Theory of Communication”



Calculating entropy

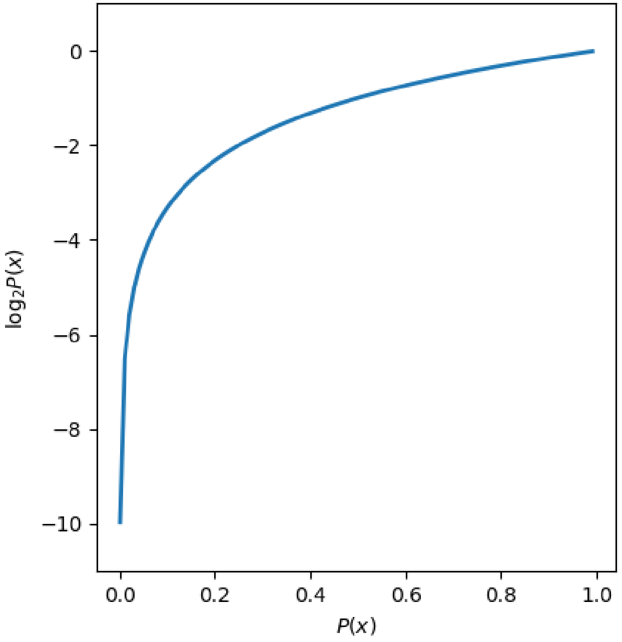
- The entropy of a dataset S with n different classes may be calculated as:

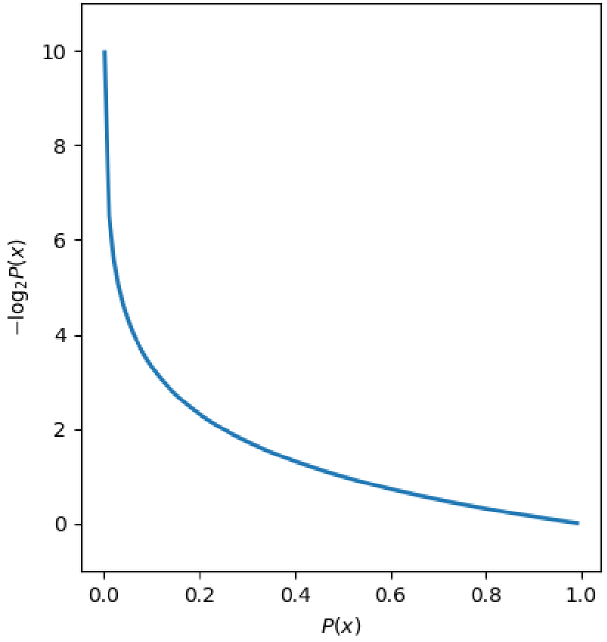
$$\text{Ent}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

- Here p_i is the proportion of class *i* in the dataset.
- This is an example of a probability mass function
- Entropy is typically measured in bits (note $\log_2$ in the equation above)
- The lowest possible entropy output from this function is 0 ($\log_2 1$ =0)
- The highest possible entropy is $\log_2 n$ (=1 when there are only 2 classes)

Why use the binary logarithm?

- A useful measure of uncertainty should:
  - Assign high uncertainty values to outcomes with a low probability
  - Assign low uncertainty values to outcomes with a high probability
- Consider the plot to the right
  - log_2 returns large negative values when P is close to 0
  - log_2 returns small negative values when P is close to 1
- Using $-log_2$ is more convenient, as this will give positive entropy values, with 0 as the lowest entropy





Entropy worked example 1

$$\mathrm{Ent}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

$$\mathrm{Ent(S)} = \mathrm{Ent([9+,5\text{-}])}$$

$$\mathrm{Ent(S)} = -9/14 \log_2(9/14) - 5/14 \log_2(5/14)$$

$$\mathrm{Ent(S)} = 0.9403$$

If calculating this in a spreadsheet application such as Excel, make sure that you are using $\log_2$ (e.g. LOG(9/14, **2**))

Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Entropy worked example 2

Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

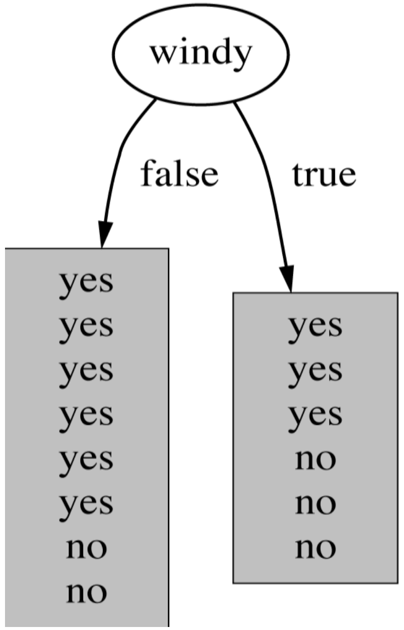
Entropy worked example 2

$$\mathrm{Ent}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

$\mathrm{Ent}(S_{windy=false}) = \mathrm{Ent}([6+,2-])$

$= -6/8 \log_2(6/8) - 2/8 \log_2(2/8)$

$= 0.3112+0.5 \ = 0.8112$



Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

$\mathrm{Ent}(S_{windy=true}) = \mathrm{Ent}([3+,3-] =$

$= -3/6 \log_2(3/6) -3/6 \log_2(3/6)$

$= 0.5+0.5 \ = \mathbf{1.0}$

Topic 2: Information-based learning

Part 5: Information gain

Information gain

- The **information gain** of an attribute is the reduction in entropy from partitioning the data according to that attribute

$$\mathrm{Gain}(S, A) = \mathrm{Ent}(S) - \sum_{v \in \mathrm{Values}(A)} \frac{|S_v|}{|S|} \mathrm{Ent}(S_v)$$

- Here S is the entire set of data being considered, and S_v refers to each partition of the data according to each possible value v for the attribute
- $|S|$ and $|S_v|$ refer to the cardinality or size of the overall dataset, and the cardinality or size of a partition respectively
- When selecting an attribute for a node in a decision tree, use whichever attribute A gives the greatest information gain

Information gain worked example

$$\text{Gain}(S, A) = \text{Ent}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Ent}(S_v)$$

$|S|=14$

$|S_{windy=true}|=6$

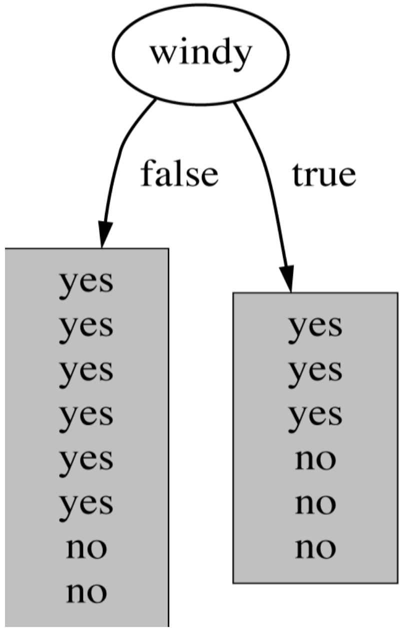
$|S_{windy=false}|=8$

Gain(S, Windy)

$= \text{Ent}(S) - |S_{windy=true}|/|S| \text{ Ent}(S_{windy=true}) - |S_{windy=false}|/|S| \text{ Ent}(S_{windy=false})$

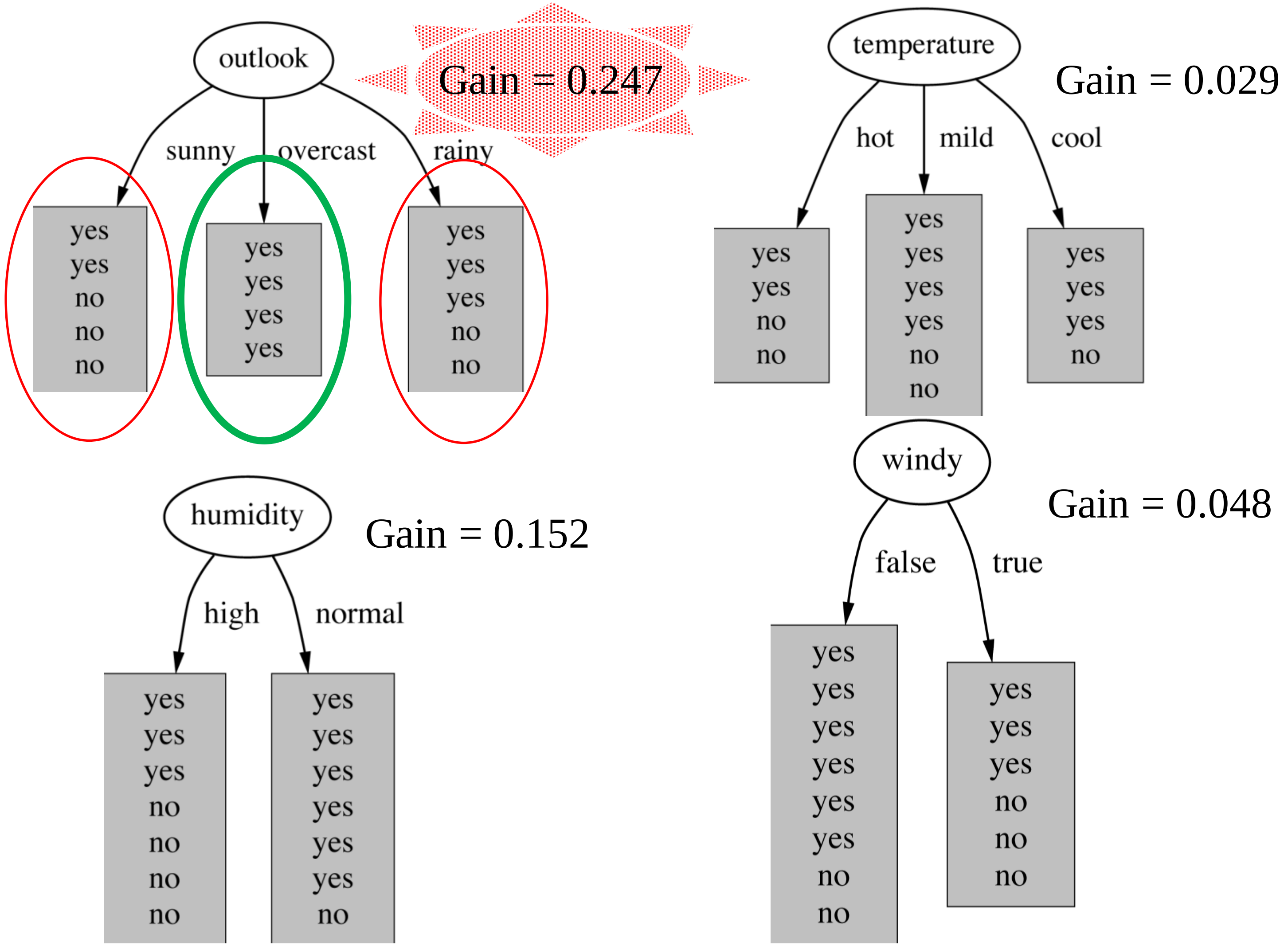
$= \text{Ent}(S) - (6/14) \text{ Ent}([3+,3-]) - (8/14) \text{ Ent}([6+,2-])$

$= 0.940 - (6/14)\ 1.00 - (8/14)\ 0.811$



Gain(S, Windy) = 0.048

Best partitioning = highest information gain

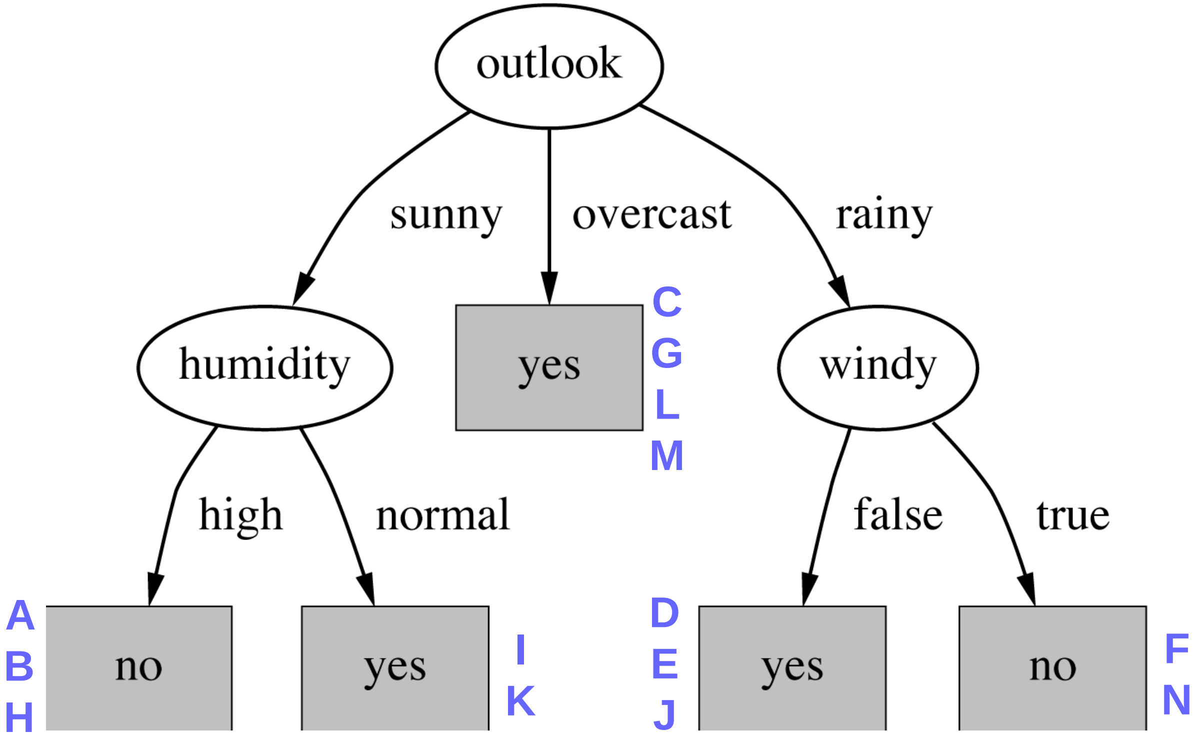


Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

Having found the best split for the root node, repeat the whole procedure with each subset of examples …
S will now refer to the subset in the partition being considered, instead of the entire dataset

Example: complete decision tree



Anyone for Tennis?

ID	Outlook	Temp	Humidity	Windy	Play?
A	sunny	hot	high	false	no
B	sunny	hot	high	true	no
C	overcast	hot	high	false	yes
D	rainy	mild	high	false	yes
E	rainy	cool	normal	false	yes
F	rainy	cool	normal	true	no
G	overcast	cool	normal	true	yes
H	sunny	mild	high	false	no
I	sunny	cool	normal	false	yes
J	rainy	mild	normal	false	yes
K	sunny	mild	normal	true	yes
L	overcast	mild	high	true	yes
M	overcast	hot	normal	false	yes
N	rainy	mild	high	true	no

What about Temp = {Hot, Mild, Cool}?