Aprendizagem Computacional
Machine Learning

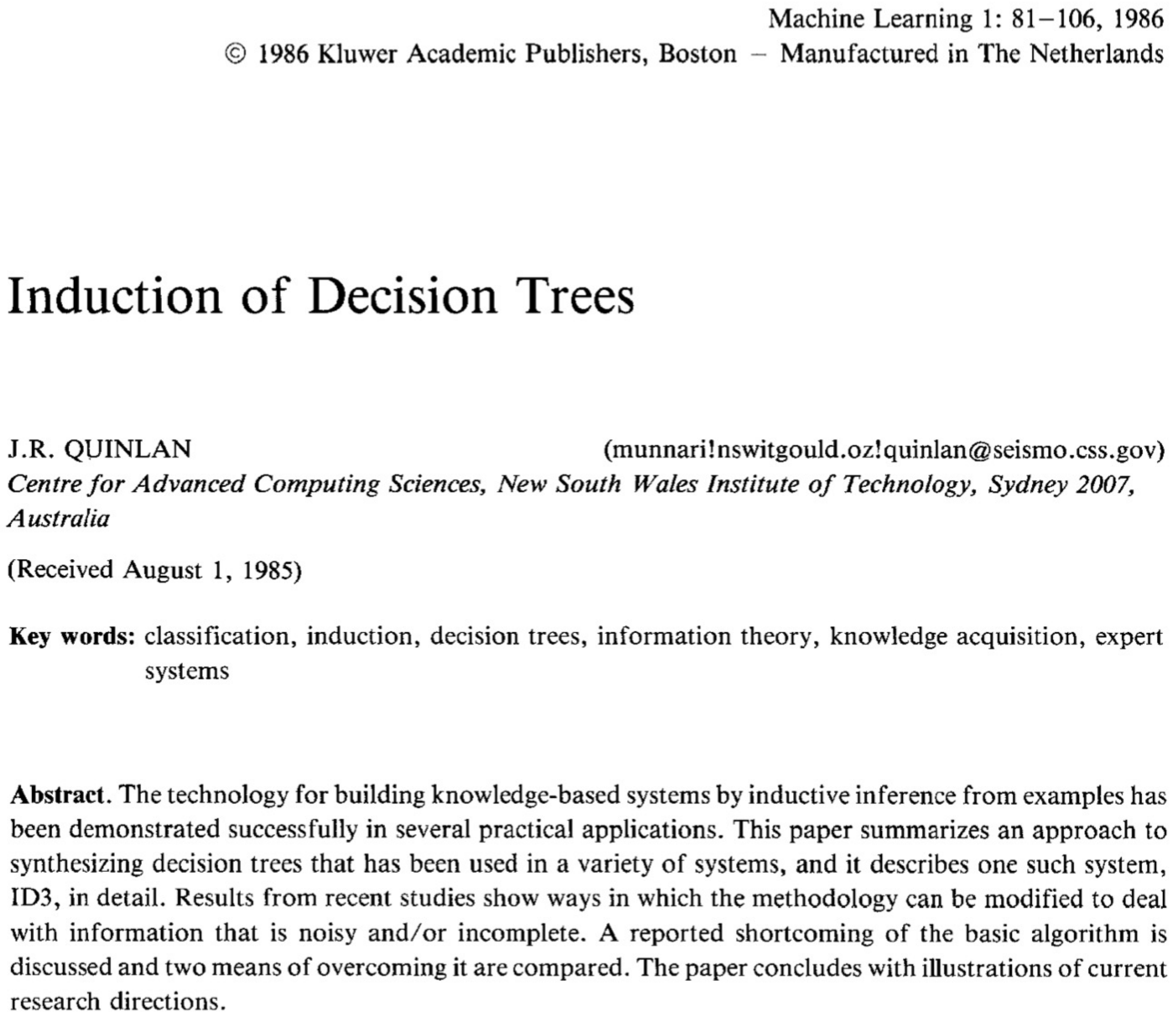
Decision trees

Departamento de Engenharia Informática,
Universidade de Coimbra

Catarina Silva, 2024

Contents

Quinlan



Machine Learning 1: 81–106, 1986
© 1986 Kluwer Academic Publishers, Boston – Manufactured in The Netherlands

Induction of Decision Trees

J.R. QUINLAN (munnari!nswitgould.oz!quinlan@seismo.css.gov)
Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007, Australia

(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

Abstract. The technology for building knowledge-based systems by inductive inference from examples has been demonstrated successfully in several practical applications. This paper summarizes an approach to synthesizing decision trees that has been used in a variety of systems, and it describes one such system, ID3, in detail. Results from recent studies show ways in which the methodology can be modified to deal with information that is noisy and/or incomplete. A reported shortcoming of the basic algorithm is discussed and two means of overcoming it are compared. The paper concludes with illustrations of current research directions.

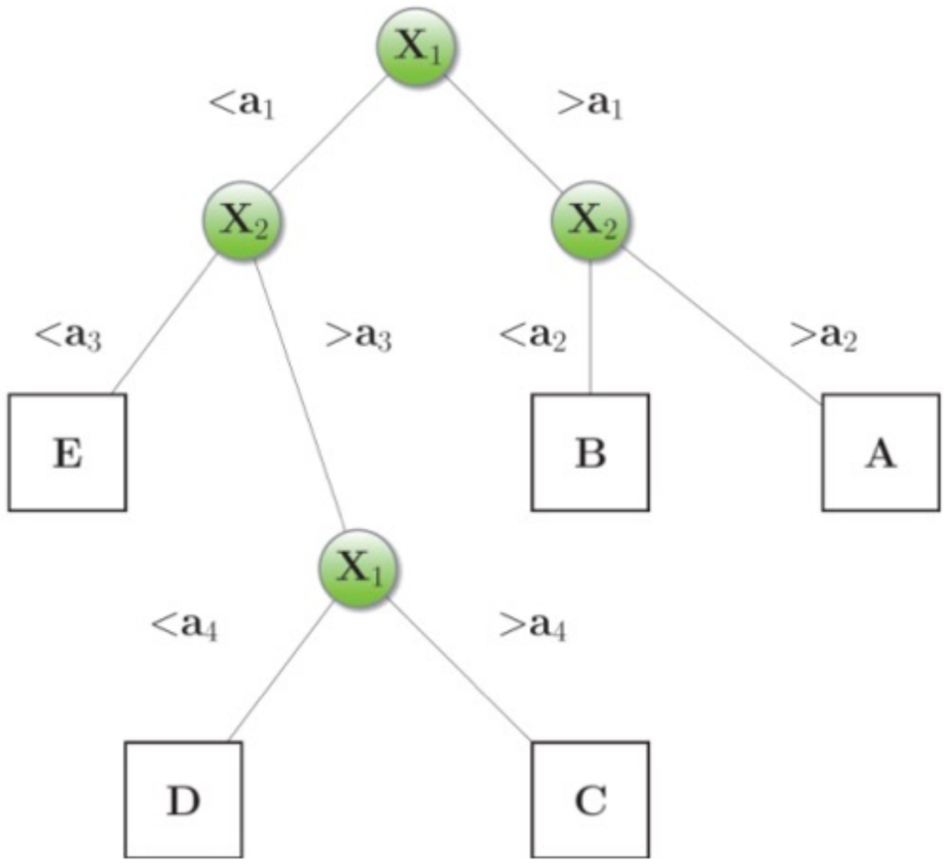
Introduction (1)

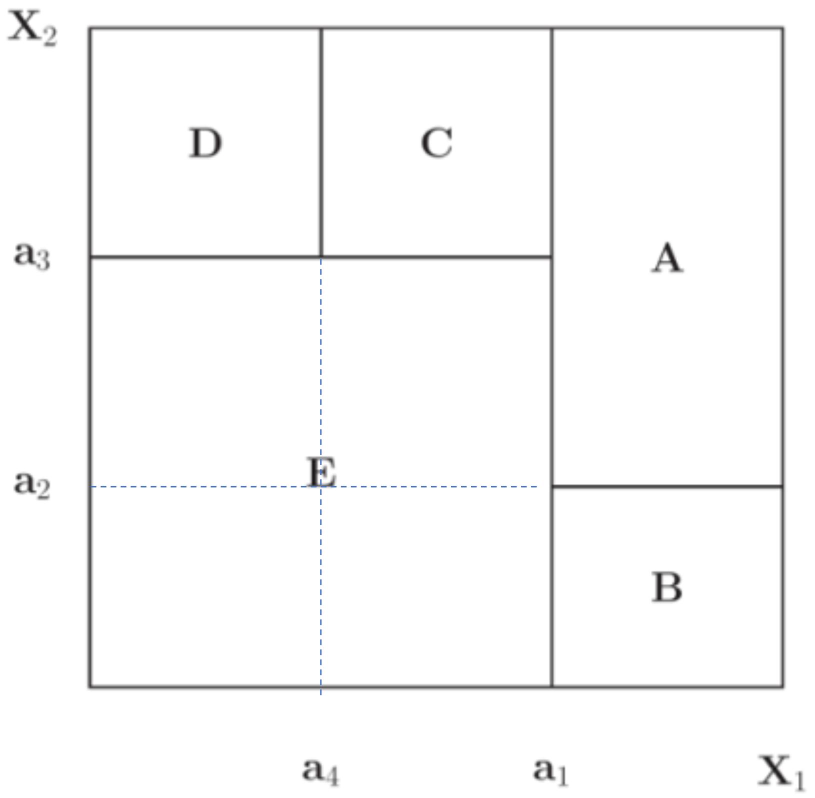
- The basic idea is to solve simpler sub-problems starting from a complex problem.
- As decision trees are very simple to understand, they can be transformed into a set of if-then rules.
- A decision tree uses the divide-and-rule strategy, which consists of decomposing a complex problem into simpler sub-problems.
- This strategy is applied recursively to each subproblem.

Introduction (2)

- The ability to discriminate comes from dividing the space defined by the attributes into subspaces.
- A class is associated with each subspace.
- Example: Deciding whether to take an umbrella with us when we leave the house.
  - Considering the information about whether it is currently raining, we can divide the problem into two simpler ones:
    1. decide whether to wear the hat knowing that it is raining;
    2. decide whether to take the hat if it's not raining.

Attribute space partition example

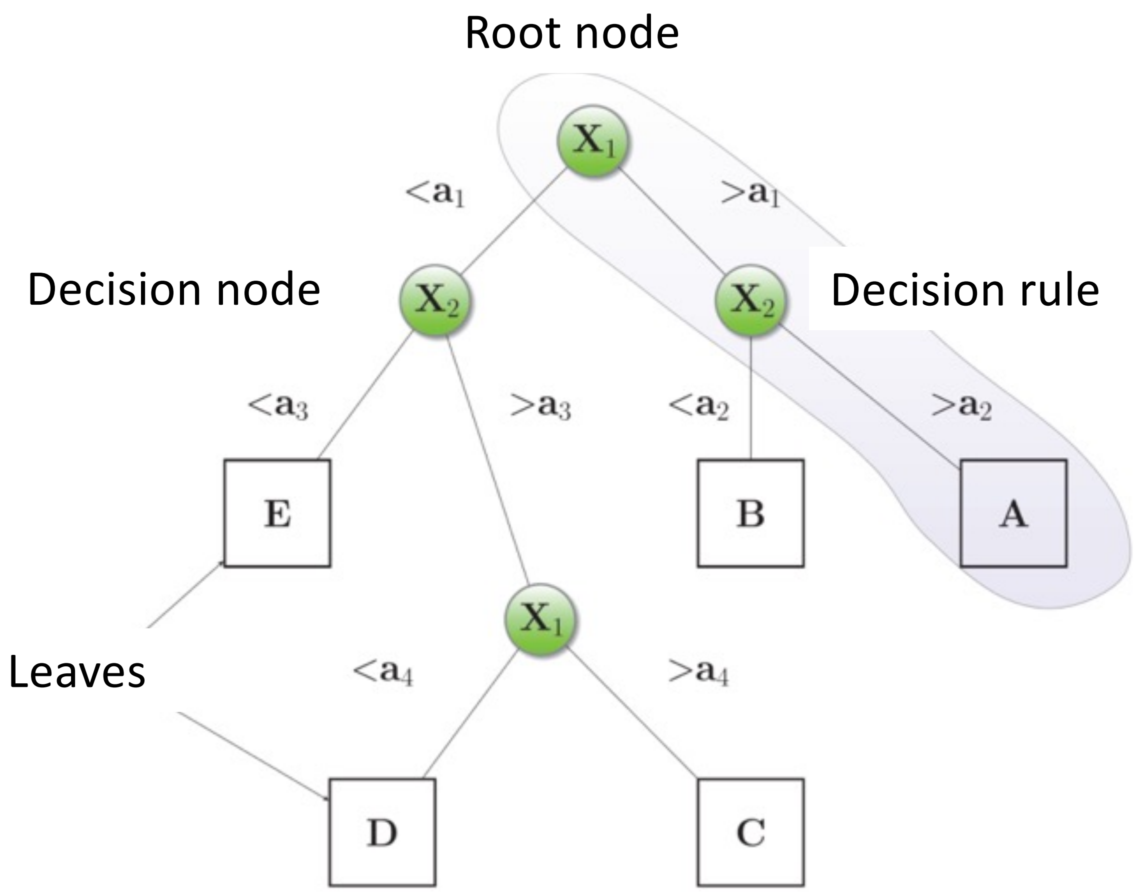




Representation (1)

- For the representation of a decision tree
  - We use **decision nodes** that contain a test relative to a certain attribute
  - each **descending branch** corresponds to a possible value (or range of values) of that attribute.
  - each **leaf** is associated with a class, and each path in the tree (from root to leaf) corresponds to a classification rule

Representation (2)

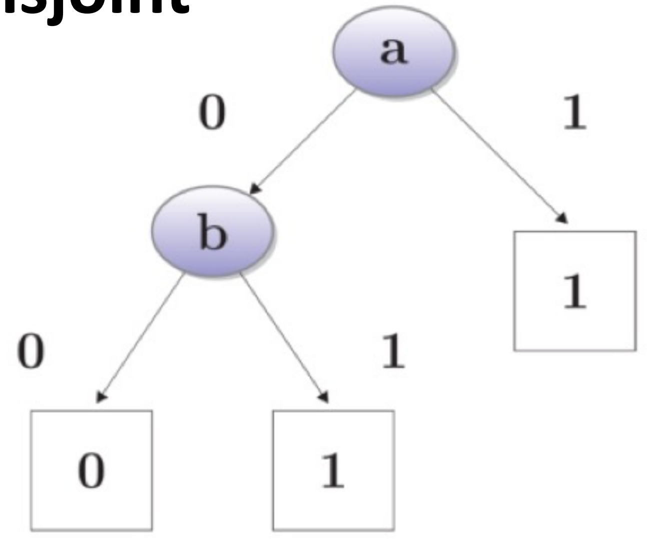


Representation (3)

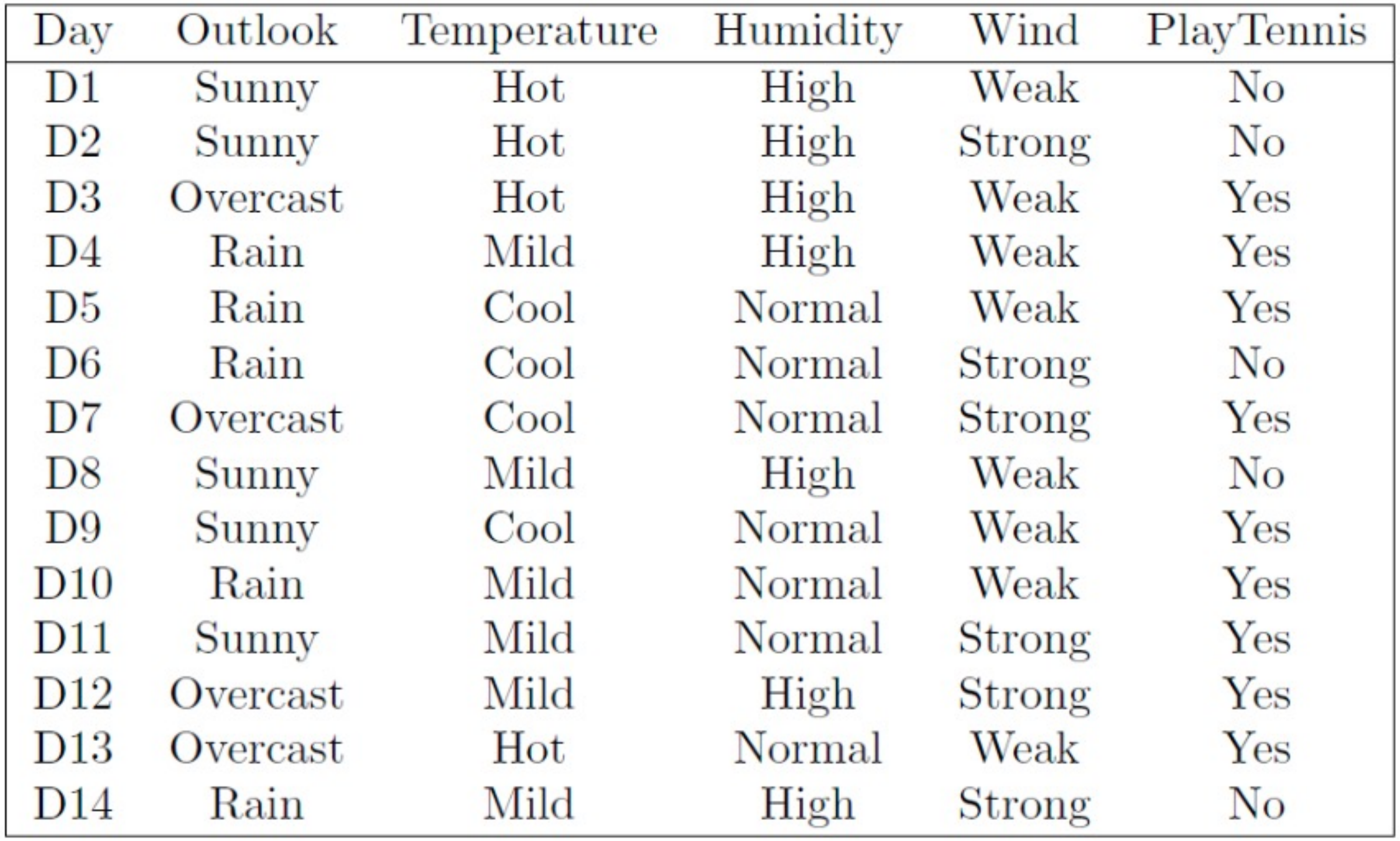
- A decision tree represents the **disjunction of conjunctions** of constraints on attribute values.
- Each path in the branch in the tree is a **conjunction** of conditions
- The set of branches in the tree is **disjoint**
- Any logical function can be

represented by a decision tree,

as the **OR function**



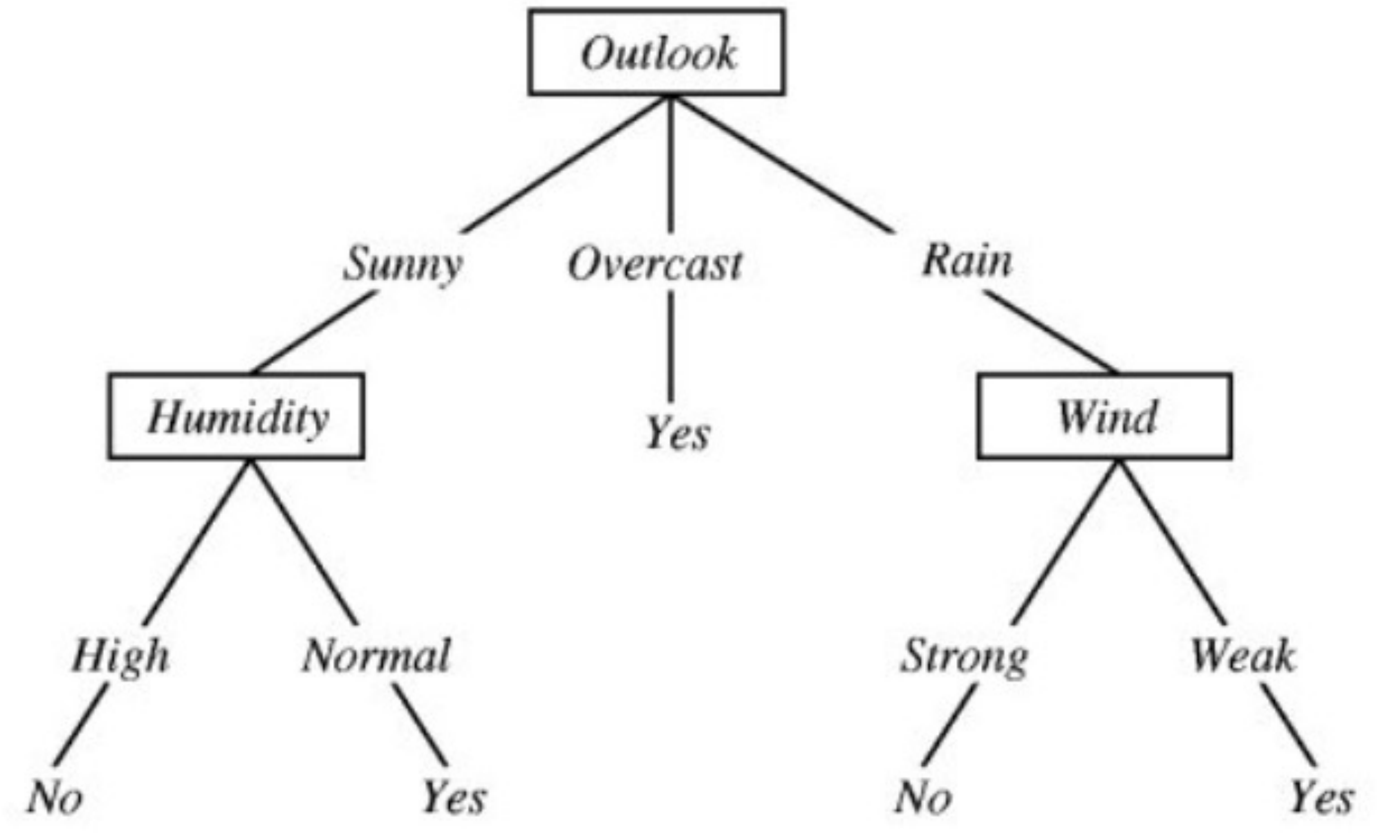
Canonical example: Play tennis?



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

Example from Mitchell, T (1997). Machine Learning, McGraw Hill

Canonical example tree



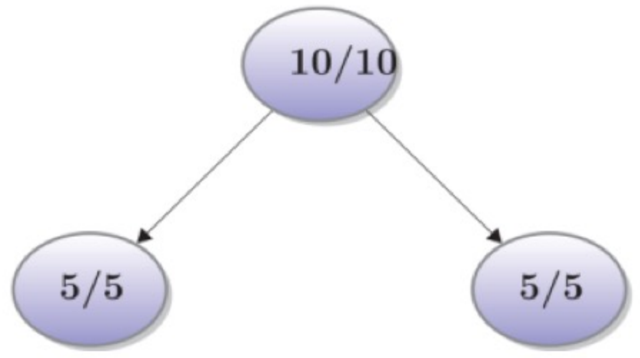
Example from Mitchell, T (1997). Machine Learning, McGraw Hill

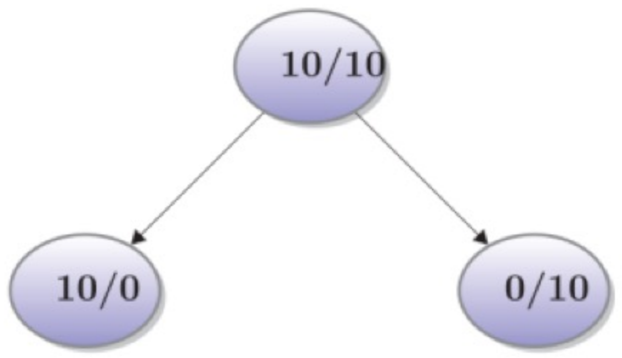
Construction – ID3 algorithm

- While training examples are not perfectly classified, do
  1. choose the "most informative" attribute θ (that has not already been used) as the decision attribute for the next node N (greedy selection)
  2. for each value (discrete θ) / range (continuous θ), create a new descendant of N
  3. sort the training examples to the descendants of N

How to measure the ability of a given attribute to discriminate between classes? (1)

- There are many measures and the choice of an attribute normally involves the use of heuristics that look one step ahead and usually do not reconsider the options already taken.
- Heuristics agree in two aspects:
  1. A partition that maintains the proportions of classes in all partitions becomes useless
  2. A partition where in each partition all the examples are of the same class has maximum utility





How to measure the ability of a given attribute to discriminate between classes? (2)

- Partition measures can be characterized in 2 ways:
  1. By measuring the **difference given by the proportions** of the classes between the current node and the descending nodes. It has the advantage of valuing the purity of the partitions.

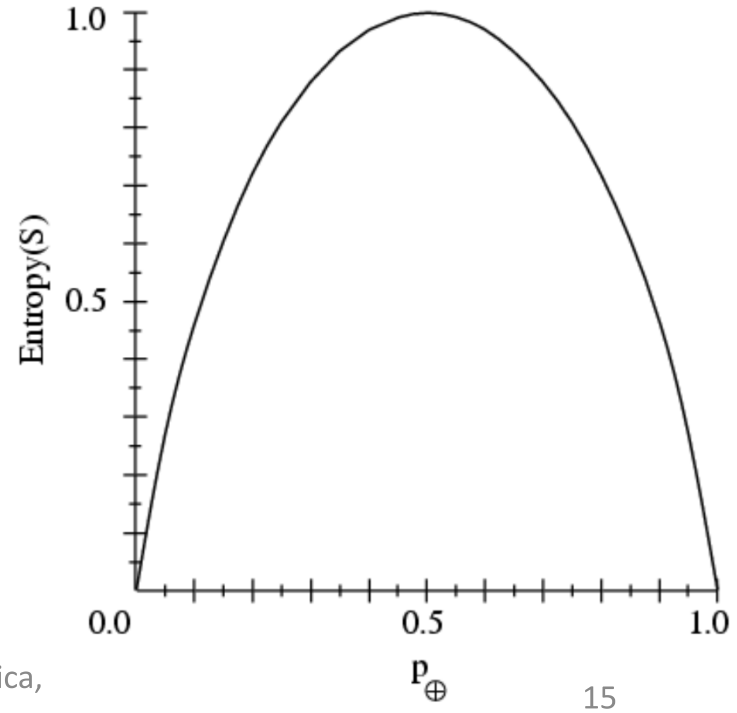
  2. By measuring the **difference given by a function based on the proportions** of the classes between the descendant nodes. In this case, the disparity between the partitions is valued, and being a measure of independence, it is also a measure of the degree of association between the attributes and the class.

Entropy (1)

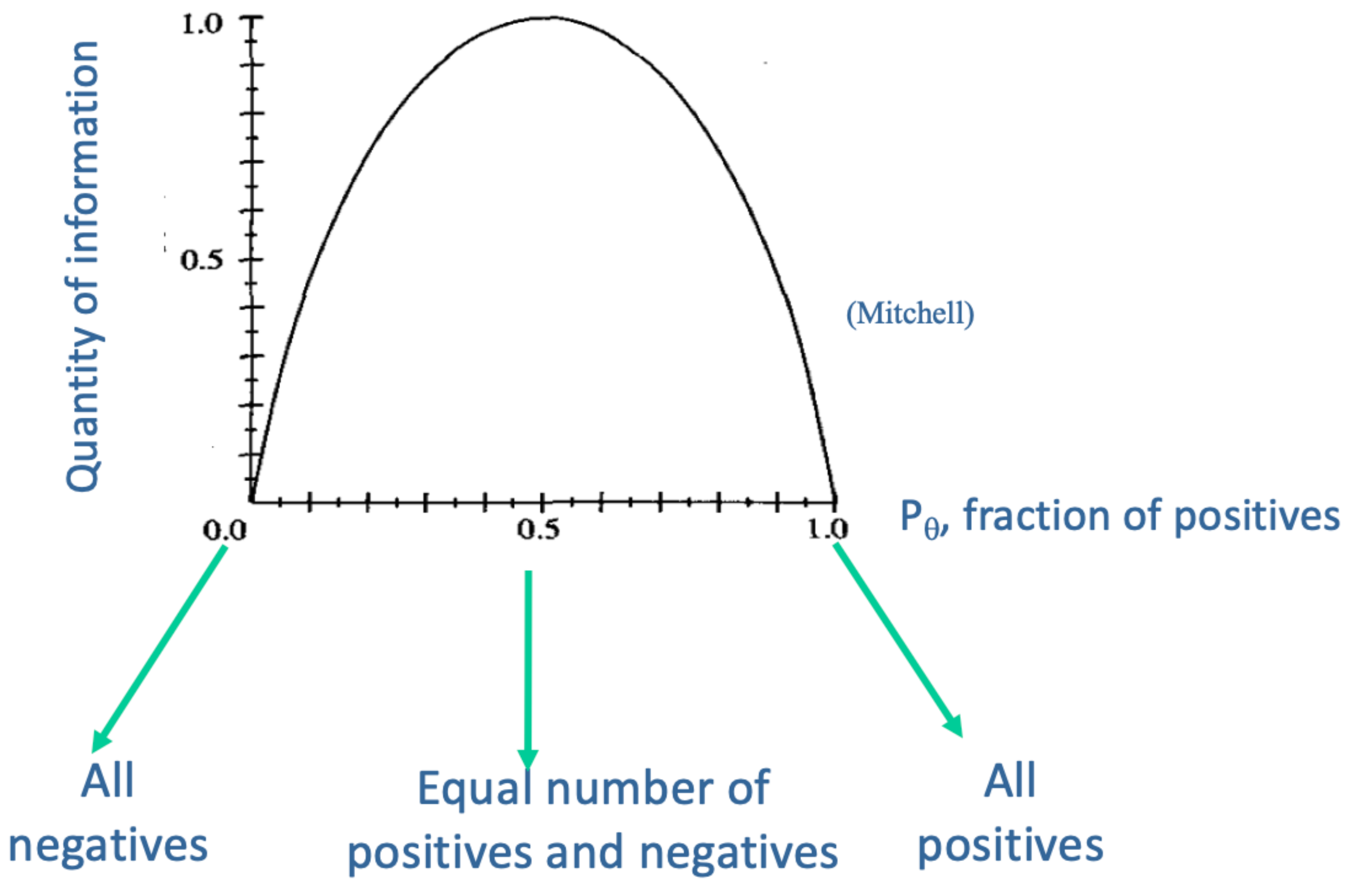
- Entropy is a measure of the randomness of a variable:

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

- More uncertainty, more entropy!
- Information Theory interpretation: H(Y) is the expected number of bits needed to encode a randomly drawn value of Y (under most efficient code)



Entropy (2)



Example from @ADC/DEI

Entropy example

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

$P(Y=t) = 5/6$

$P(Y=f) = 1/6$

$H(Y) = -\ 5/6 \log_2 5/6 - 1/6 \log_2 1/6$

$= 0.65$

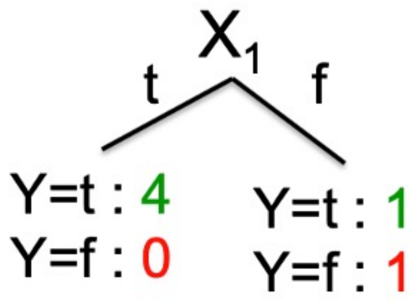
Conditional entropy

Conditional Entropy $H(Y|X)$ of a random variable Y conditioned on a random variable X

$$H(Y \mid X) = -\sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X = x_j) \log_2 P(Y = y_i \mid X = x_j)$$

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$H(Y|X_1) = -\,4/6\,(1 \log_2 1 + 0 \log_2 0)$$
$$-\,2/6\,(1/2 \log_2 1/2 + 1/2 \log_2 1/2)$$
$$= 2/6$$

Entropy example (2)

- Given a collection S , containing positive and negative examples of a target, the Entropy of S relative to that Boolean classification is

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

where

$p_{\oplus}$ is the proportion of positive examples in S

$p_{\odot}$ is the proportion of negative examples in S

In all calculations it is considered that $0 \log 0 = 0$

Example from @ADC/DEI

Entropy example (3)

- For a collection S with 14 examples of a boolean concept, being 9 positives and 5 negatives (notation 9+ and 5-), the entropy S relative to the boolean classification will be :

$$Entropy[9+, 5-] =$$

$$= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0,940$$

$$Entropy[14+, 0-] =$$

$$= -(14/14)\log_2(14/14) - (0/14)\log_2(0/14) =$$

$$= -1 \times 0 - 0\log_2 0 = 0$$

$$Entropy[7+, 7-] = -(7/14)\log_2(7/14) - (7/14)\log_2(7/14) =$$

$$= -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) =$$

$$= -\log_2(1/2) = -(-1) = 1$$

Example from @ADC/DEI

Entropy considerations

- The entropy measures the level of “impurity” (high entropy, high impurity) or of homogeneity (low entropy, high homogeneity) of a training data set.
- Descending the tree, from the root to the leaves, the level of entropy must diminish, such that the homogeneity will increase until one well defined class is reached (entropy equal to zero).
- This objective allows to define a measure of how much appropriate is an attribute to classify the training data set.
- This measure is the information gain , that is exactly the expected reduction of the entropy obtained by the partition of that data with that attribute.

From @ADC/DEI

Information gain

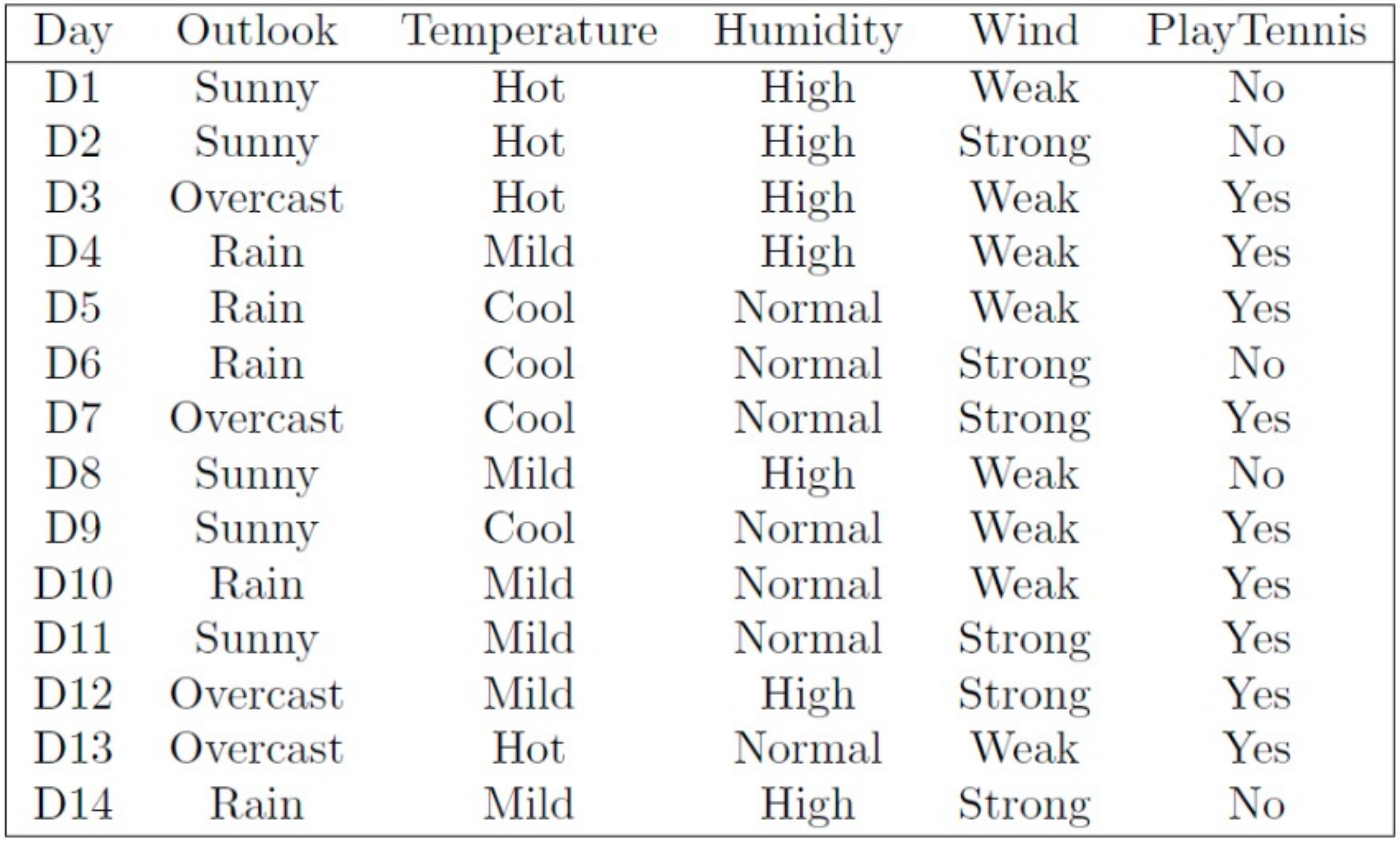
- The information gain Gain (S,A) of an attribute A, in relation to a collection S of examples, is

$$Gain(S,A) = \underbrace{Entropy(S)}_{\text{Initial entropy of } S} - \underbrace{\sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)}_{\text{Expected values of the entropy after partitioning } S \text{ using the attribute A}}$$

- Where *Values(A)* is the set of all possible values of the attribute *A*, and S_v is the subset of *S* for which the attribute *A* has the value *v*, that is $S_v = \{ s \in S \mid A(s) = v \}$

From @ADC/DEI

Decision tree for Play tennis?



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

Example from Mitchell, T (1997). Machine Learning, McGraw Hill

Play tennis analysis

- Outlook: Sunny, Overcast, Rain
- Temperature:Hot, Mild, Cold
- Humidity:High, Normal
- Wind: Weak, Strong

- Combinations: 3x3x2x2=36 (at least)
- However, the example only has 14 examples!!!

Computing the gain: Wind

- In the set S with the 14 training examples , we have [9+,5-].

- *Wind= Weak* in 6 positives and 2 negatives , S_{Wweak}: [6+, 2-]
- *Wind=Strong* in 3 positives e 3 negatives, $S_{Wstrong}$: [3+,3-]

$$Entropy(S) = Entropy[9+,5-] = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0,940$$

$$Entropy(S_{Wstrong}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = (1/2)\log_2 2 + (1/2)\log_2 2 = 1/2 + 1/2 = 1,00$$

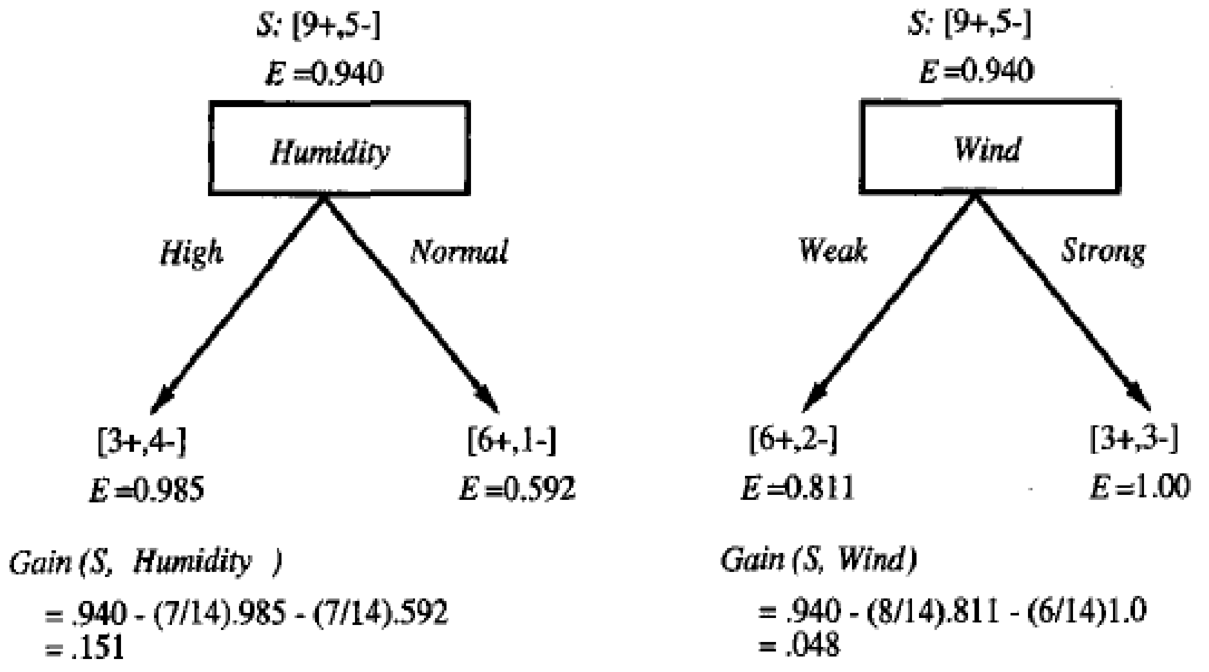
$$Entropy(S_{Wweak}) = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0,811$$

$$Gain(S,Wind) = Entropy(S) - \frac{|S_{Wweak}|}{|S|} Entropy(S_{Wweak}) - \frac{|S_{Wstrong}|}{|S|} Entropy(S_{Wstrong}) =$$

$$=0.940-(8/14)0.811-(6/14)1.00=0.048$$

Computing the gain: Humidity

- It is *High* in 3 positives and 4 negatives S_{Hhigh} : [3+, 4-]
- It is normal *Normal* in 6 positives 1 negative $S_{Hnormal}$: [6+,1-]



Example from Mitchell, T (1997). Machine Learning, McGraw Hill

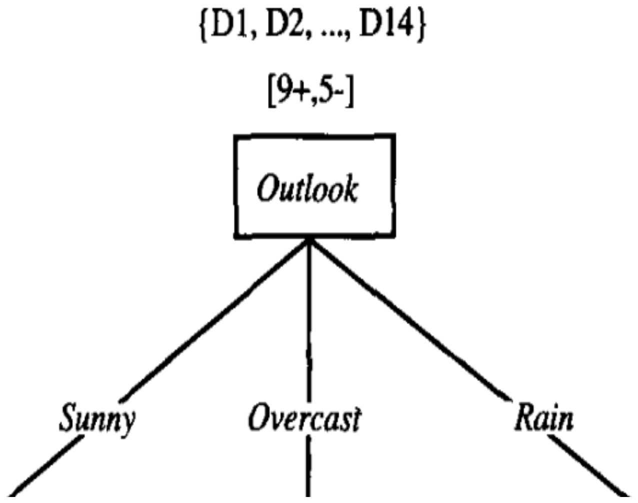
Computing the gain

$$\boxed{Gain(S, Outlook) = 0.246}$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

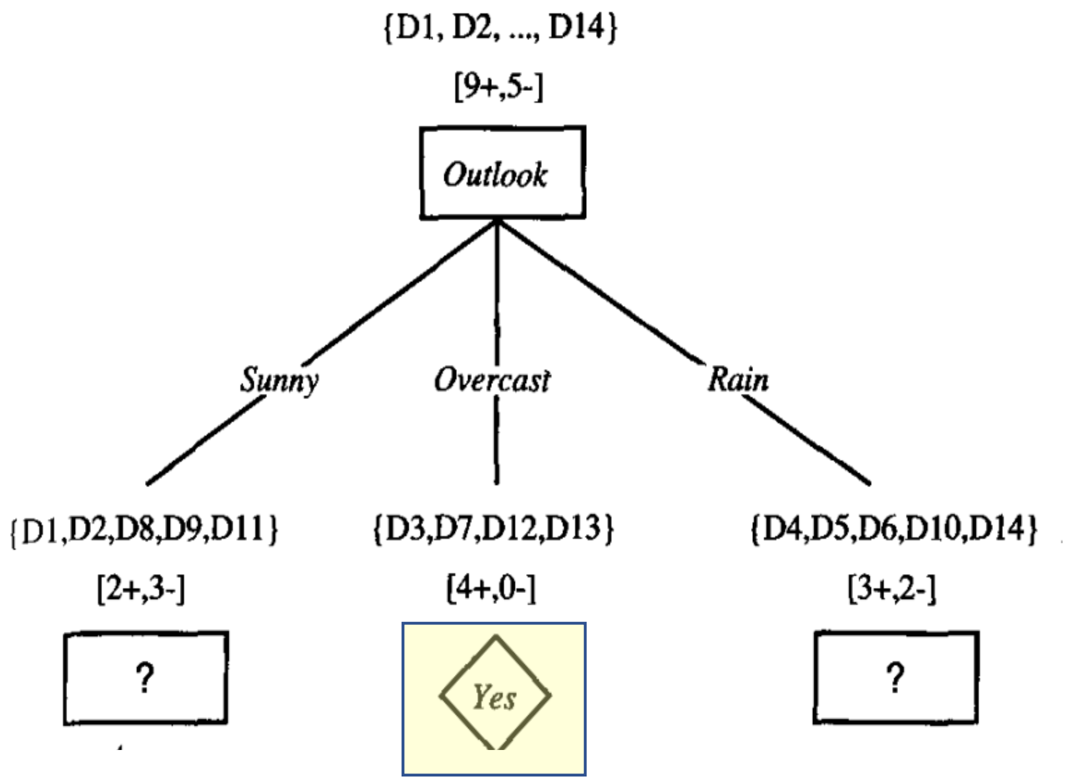
$$Gain(S, Temperature) = 0.029$$



Example from Mitchell, T (1997). Machine Learning, McGraw Hill

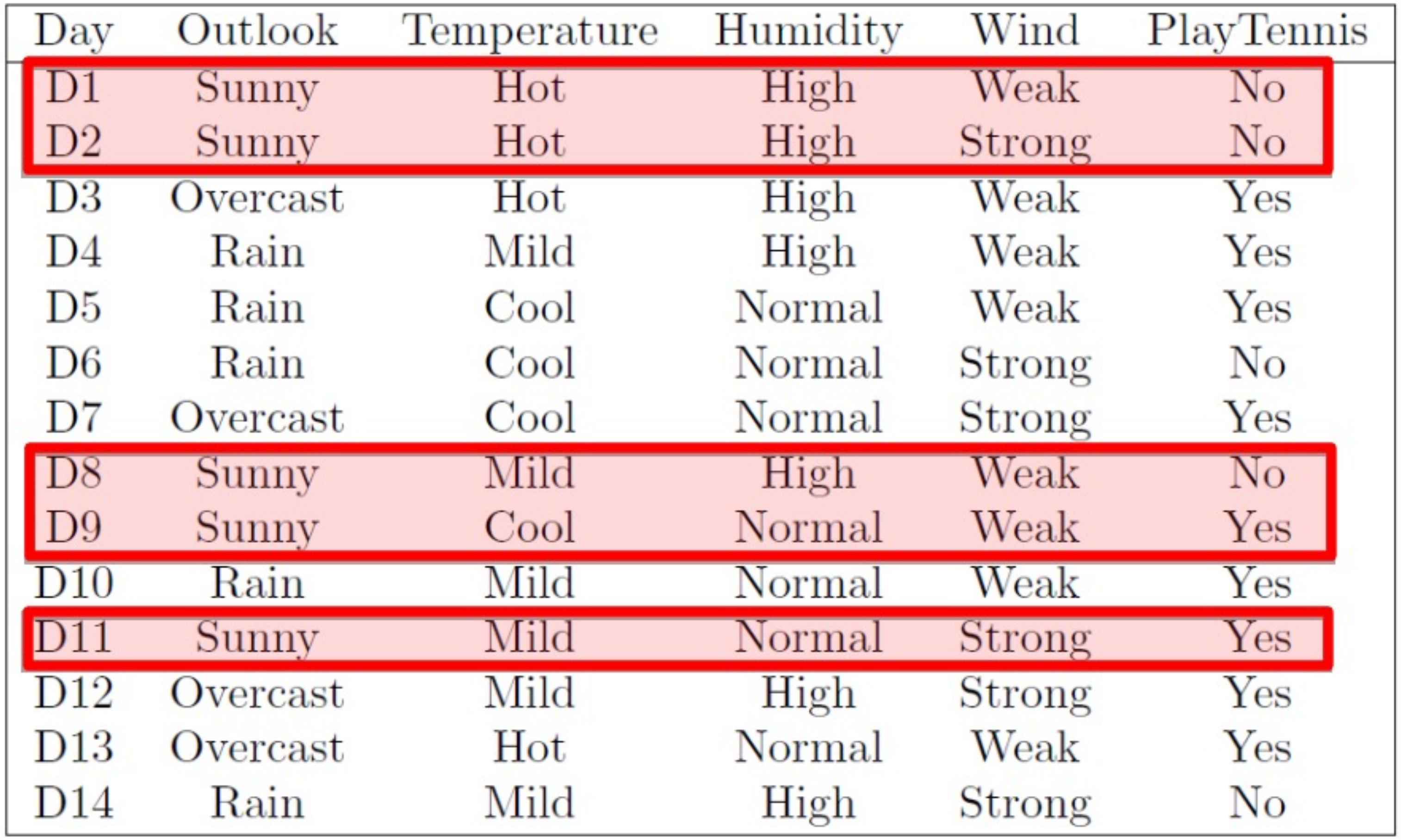
Second level (1)

- For each branch repeat the process



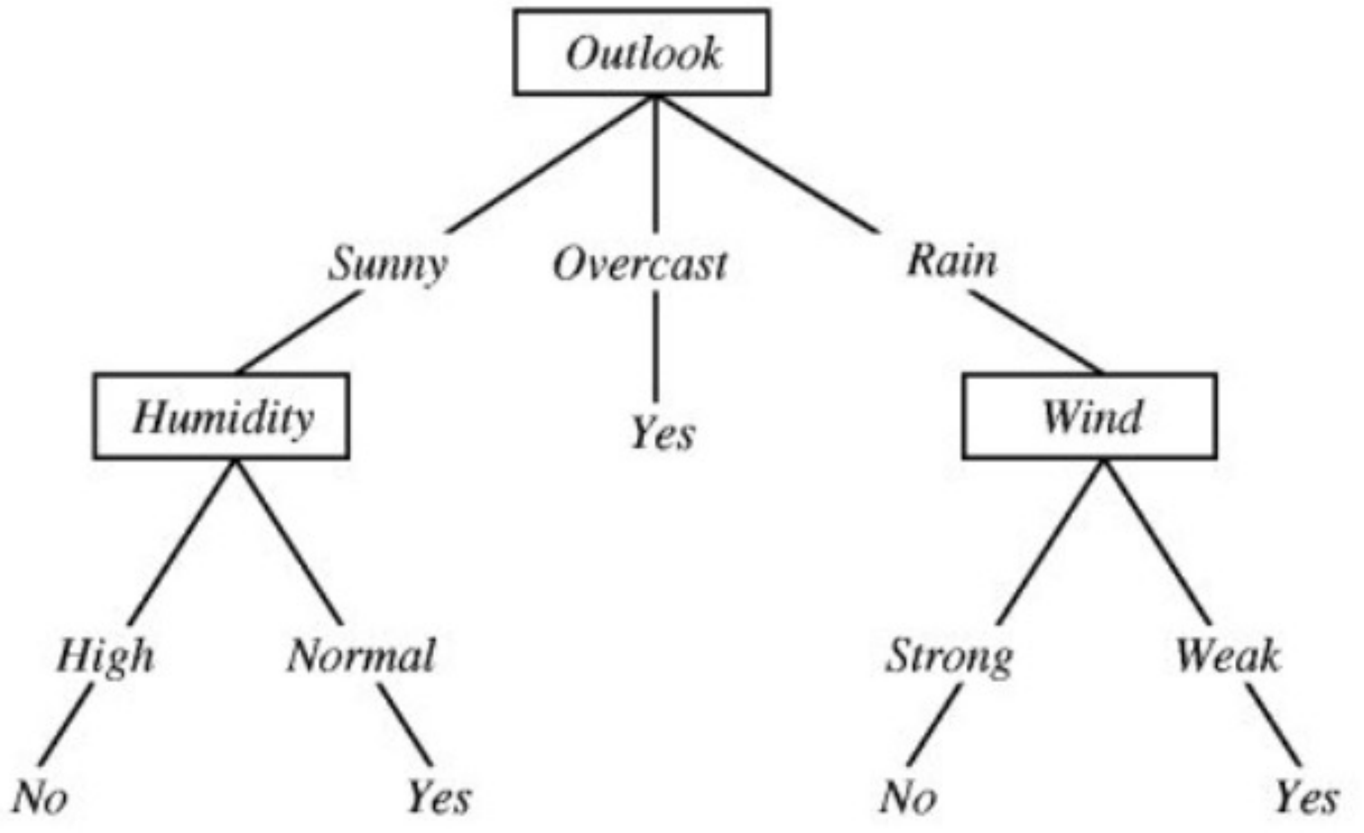
Example from Mitchell, T (1997). Machine Learning, McGraw Hill

Second level (2)



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

Final tree – Play tennis?



Example from Mitchell, T (1997). Machine Learning, McGraw Hill

ID3 – final considerations

- No warranty that the smaller tree is achieved:
  - shortest trees are preferred (Occam’s razor)
  - attributes with larger information gain are put closer to the root

C4.5 – alternative to ID3

- Uses gain ratio to get the most informative attribute
- Split information

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- Gain ratio:

$$GaiRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

where Gain (S,A) is the previous information gain

Overfitting

- Post-Prunning using the validation set

Continous values

- Binning: partitioning the continous attributes values into a set of intervals, defining a set of thresholds

Missing data

- Different alternatives:
  - Remove data
  - Substitute with a constant value
  - Substitute with most common value (mode)
  - Substitute with the most common for the desired class
  - Etc.

Different costs

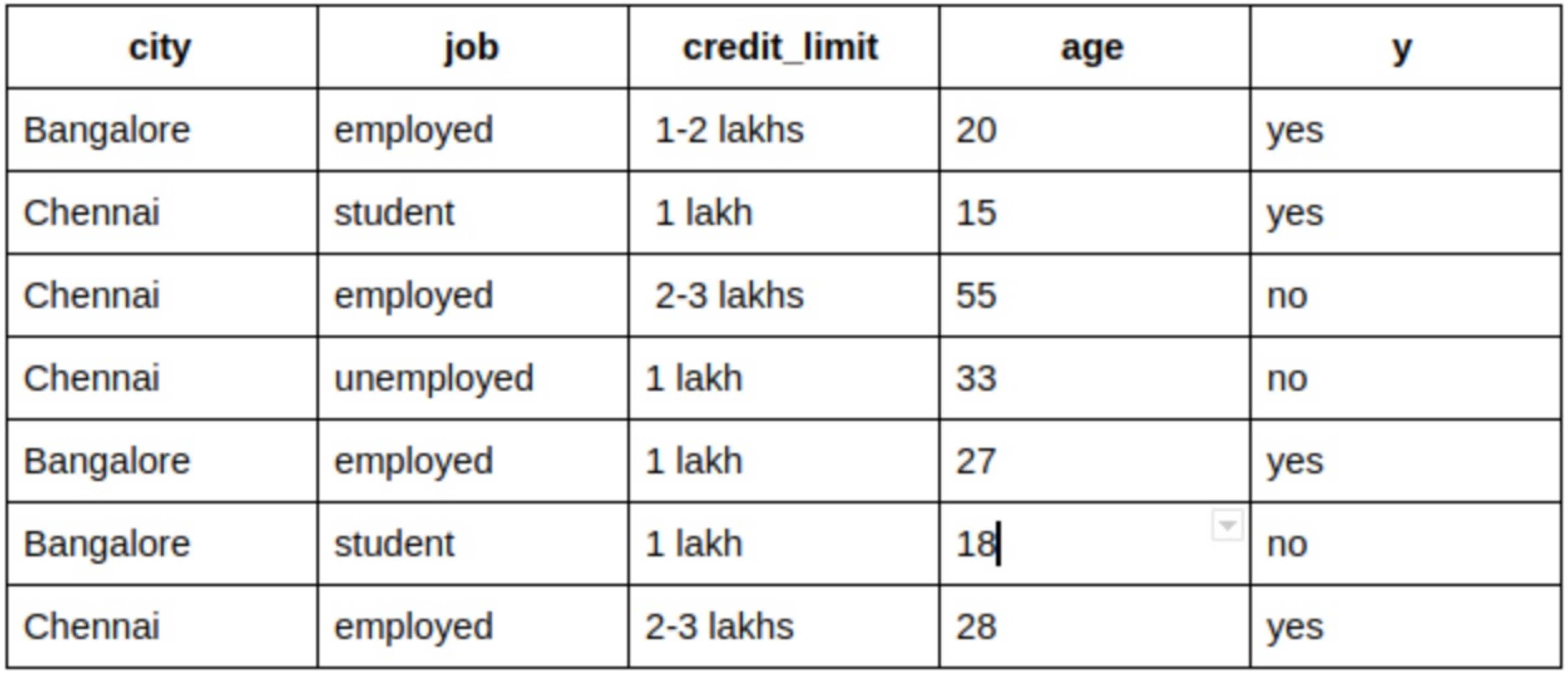
- Often attributes have different costs of
  - Obtaining
  - Impact
  - Anonymity
  - Etc
- To cope different costs can be used when computing the gain, e.g., by reducing the gain for costly attributes

DT – Final considerations

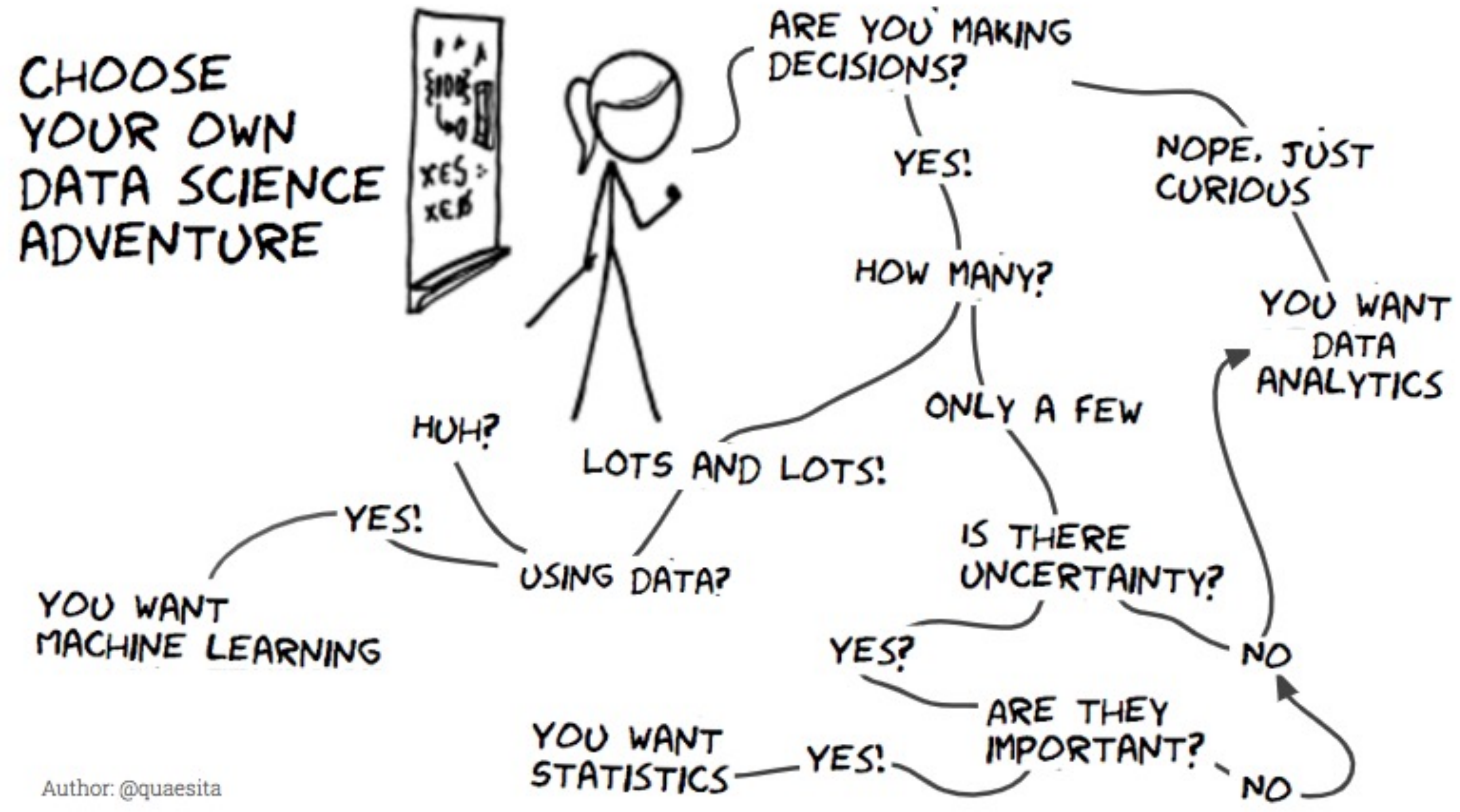
- Flexible and robust approach
- Efficient
- Interpretable

- Simple decision boundaries
- Problems with missing data, continuous attributes
- Unstable (small variations -> big changes)
- Prone to overfitting

Example (homework)



city	job	credit_limit	age	y
Bangalore	employed	1-2 lakhs	20	yes
Chennai	student	1 lakh	15	yes
Chennai	employed	2-3 lakhs	55	no
Chennai	unemployed	1 lakh	33	no
Bangalore	employed	1 lakh	27	yes
Bangalore	student	1 lakh	18	no
Chennai	employed	2-3 lakhs	28	yes


CHOOSE YOUR OWN DATA SCIENCE ADVENTURE
ARE YOU MAKING DECISIONS?
YES!
NOPE, JUST CURIOUS
YOU WANT DATA ANALYTICS
HOW MANY?
LOTS AND LOTS!
ONLY A FEW
HUH?
USING DATA?
YES!
YOU WANT MACHINE LEARNING
IS THERE UNCERTAINTY?
YES?
NO
ARE THEY IMPORTANT?
YES!
NO
YOU WANT STATISTICS
Author: @quaesita