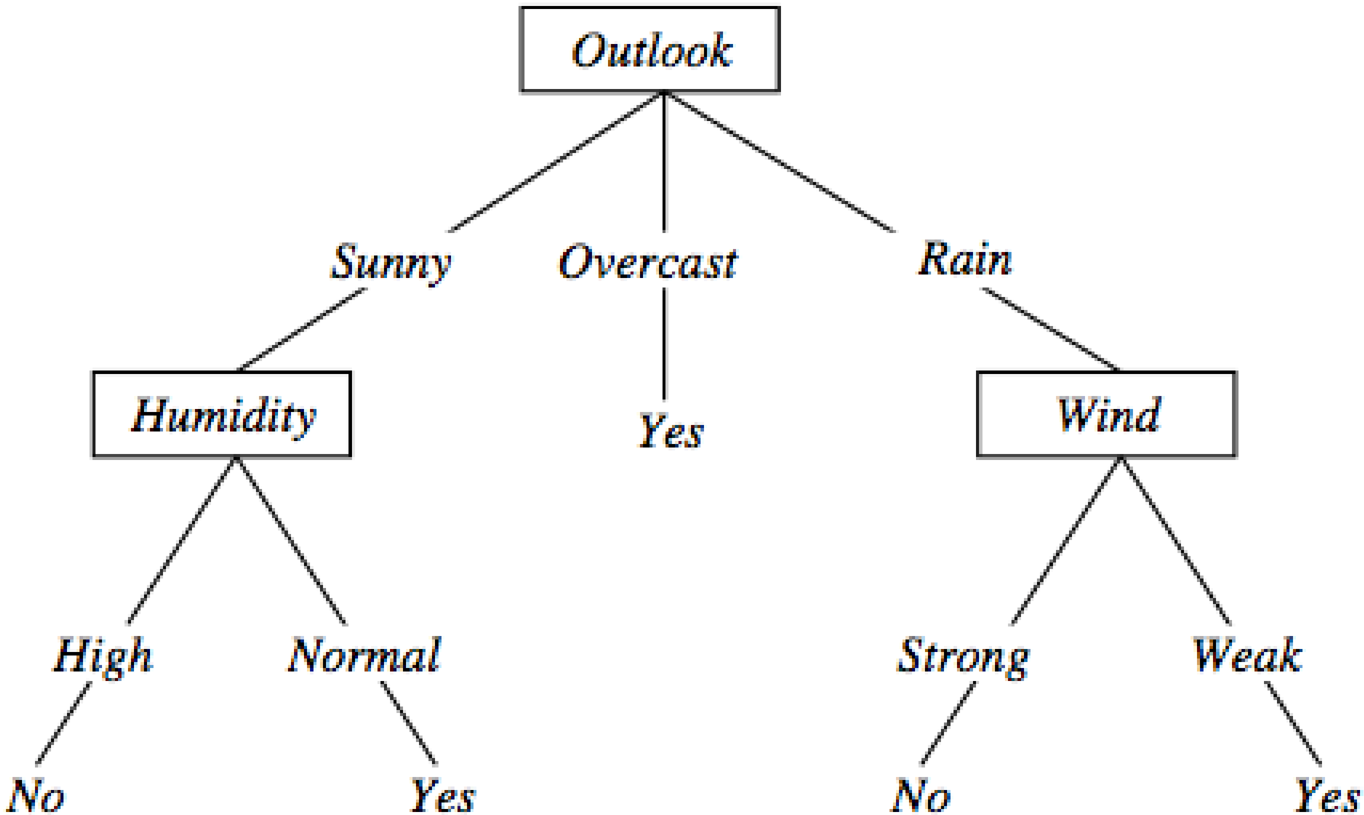
Decision Tree based Learning

Example

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

Decision tree representation (PlayTennis)

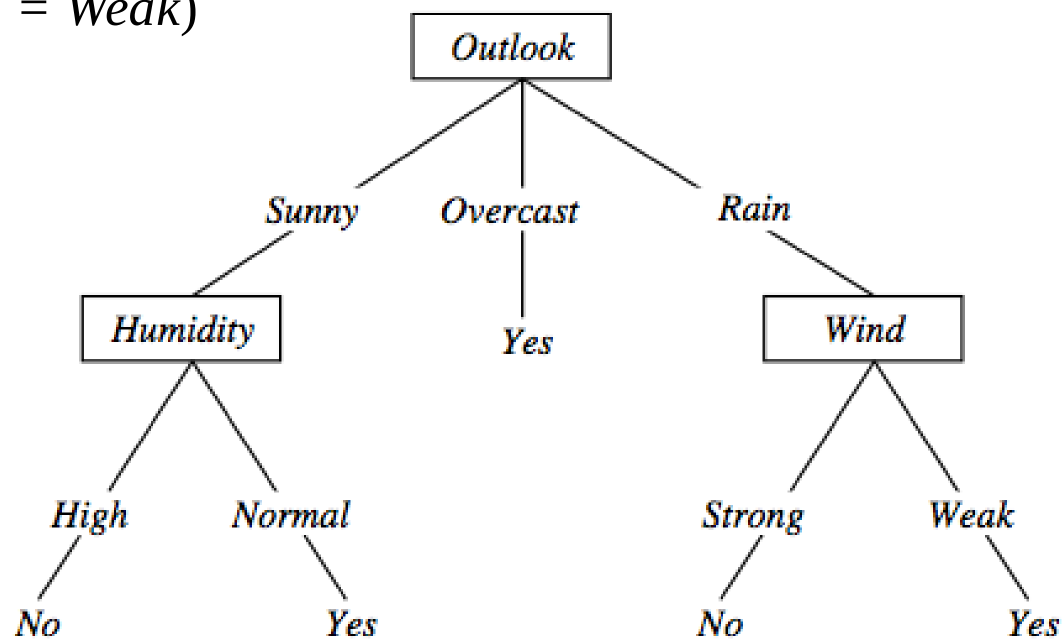


$\langle Outlook{=}Sunny, Temp{=}Hot, Humidity{=}High, Wind{=}Strong \rangle$ No

Decision trees expressivity

- Decision trees represent a disjunction of conjunctions on constraints on the value of attributes:

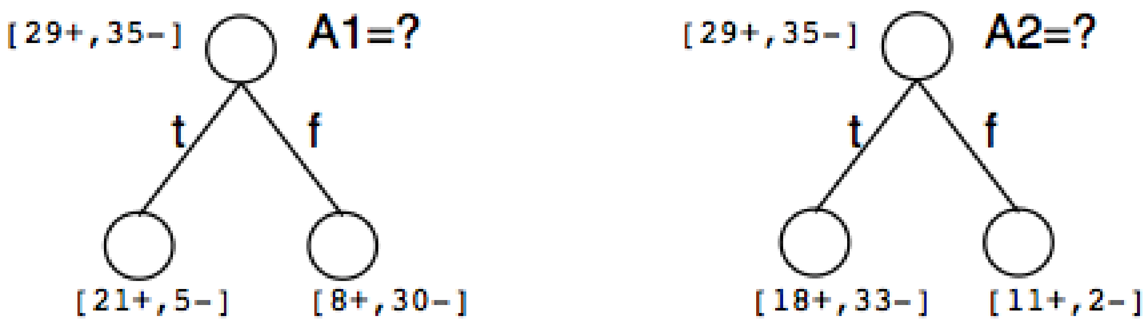
$$(Outlook = Sunny \wedge Humidity = Normal) \vee$$
$$(Outlook = Overcast) \vee$$
$$(Outlook = Rain \wedge Wind = Weak)$$



Top-down induction of Decision Trees

- ID3 (Quinlan, 1986) is a basic algorithm used to create DT's
- Given a training set of examples, the algorithms for building DT performs search in the space of decision trees.
- The construction of the tree is top-down. The algorithm is greedy.
- The fundamental question is “which attribute should be tested next? Which attribute gives us more information?”
- Select the *best* attribute
- A descendent node is then created for each possible value of this attribute and data set is partitioned according to this value.
- The process is repeated for each successor node until all the examples are classified correctly or there are no attributes left

Which attribute is the best classifier?



- A statistical property called *information gain*, measures how well a given attribute separates the training examples
- Information gain uses the notion of *entropy*, commonly used in information theory
- *Information gain = expected reduction of entropy*


prince
one
pierre
now
natasha
andrew
princess
rostov
face
eyes
thought
know
chapter
time
room
man
count
went
nicholas
moscow
french
emperor
without
something
looked
seemed
came
well
asked
like
still
first
just
though
go
two
yes
look
sonya
kutuzov
voice
house
nothing
general
right
napoleon
day
hand
people
countess
looking
began
army
suddenly
russian
knew
another
young
away
come
see
love
head
old
long
men
mary
left
whole
everything
back
little
even
felt
saw
officer
round
heard
way
made


Everyone

Entropy in binary classification

- Entropy measures the *impurity* of a collection of examples. It depends from the distribution of the random variable *p*.
  - S is a collection of training examples
  - p_+ the proportion of positive examples in S
  - p_- the proportion of negative examples in S

$$Entropy\,(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_- \quad [0 \log_2 0 = 0]$$

$$Entropy\,([14+, 0-]) = -14/14 \log_2 (14/14) - 0 \log_2 (0) = 0$$

$$Entropy\,([9+, 5-]) = -9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0.94$$

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

$$= 1/2 + 1/2 = 1 \qquad [\log_2 1/2 = -1]$$

Note: the log of a number < 1 is negative, $0 \leq p \leq 1$, $0 \leq entropy \leq 1$

Entropy in general

- Entropy measures the amount of information in a random variable

  $$H(X) = -p_+ \log_2 p_+ - p_- \log_2 p_- \qquad X = \{+, -\}$$

  for binary classification [two-valued random variable]

  $$H(X) = -\sum_{i=1}^{c} p_i \log_2 p_i = \sum_{i=1}^{c} p_i \log_2 1/p_i \qquad X = \{i, \ldots, c\}$$

  for classification in c classes

  Example: rolling a die with 8, equally probable, sides

  $$H(X) = -\sum_{i=1}^{8} 1/8 \log_2 1/8 = -\log_2 1/8 = \log_2 8 = 3$$

Information gain as entropy reduction

- *Information gain* is the *expected* reduction in entropy caused by partitioning the examples on an attribute.
- The higher the information gain the more effective the attribute in classifying training data.
- Expected reduction in entropy knowing *A*

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

$Values(A)$ possible values for A

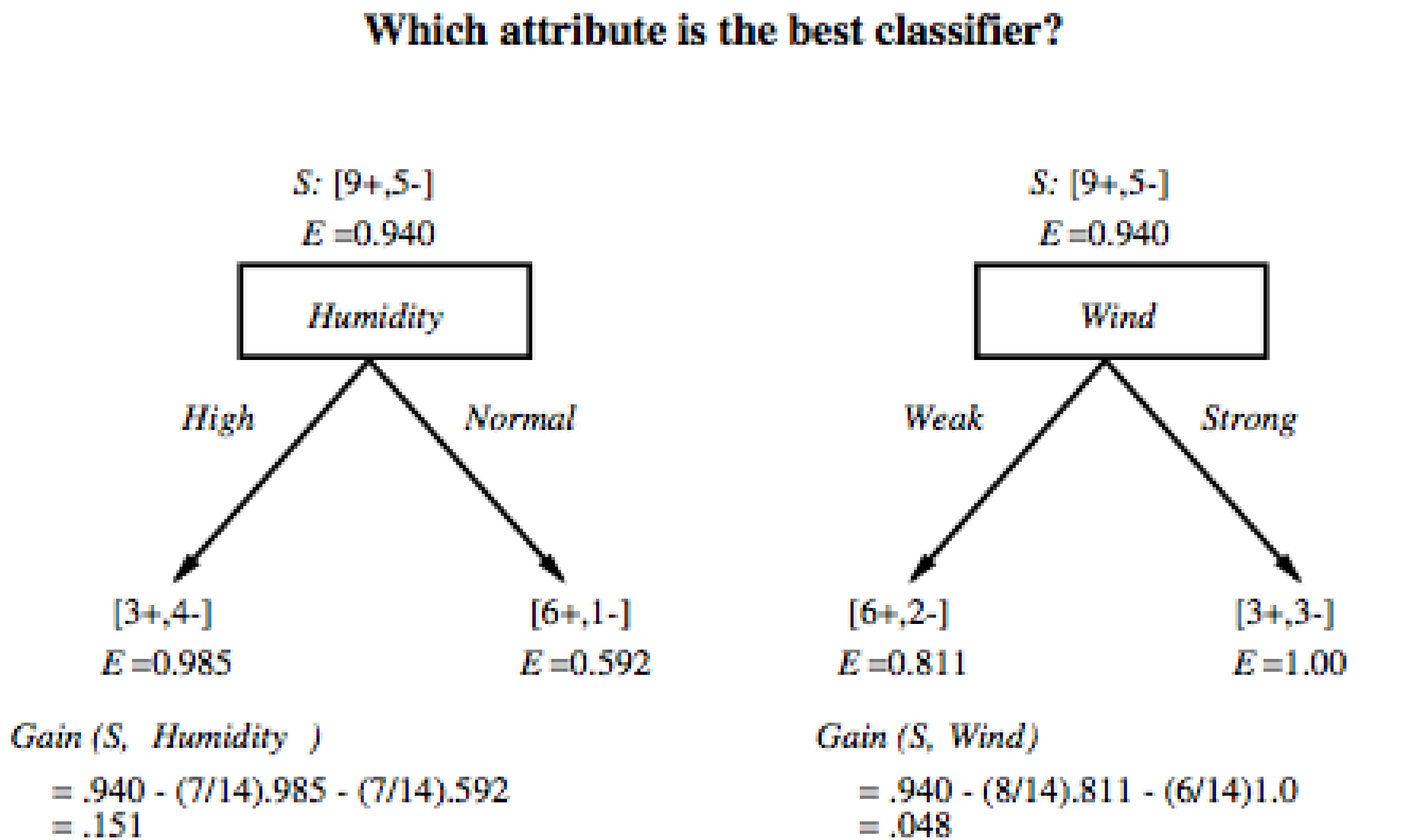
Sv subset of S for which A has value v

Example: expected information gain

- Let
  - $Values(Wind) = \{Weak, Strong\}$
  - $S = [9+, 5-]$
  - $S_{Weak} = [6+, 2-]$
  - $S_{Strong} = [3+, 3-]$
- Information gain due to knowing *Wind*:

$$
\begin{aligned}
Gain(S, Wind) &= Entropy(S) - 8/14\ Entropy(S_{Weak}) - 6/14\ Entropy(S_{Strong}) \\
&= 0.94 - 8/14 \times 0.811 - 6/14 \times 1.00 \\
&= 0.048
\end{aligned}
$$

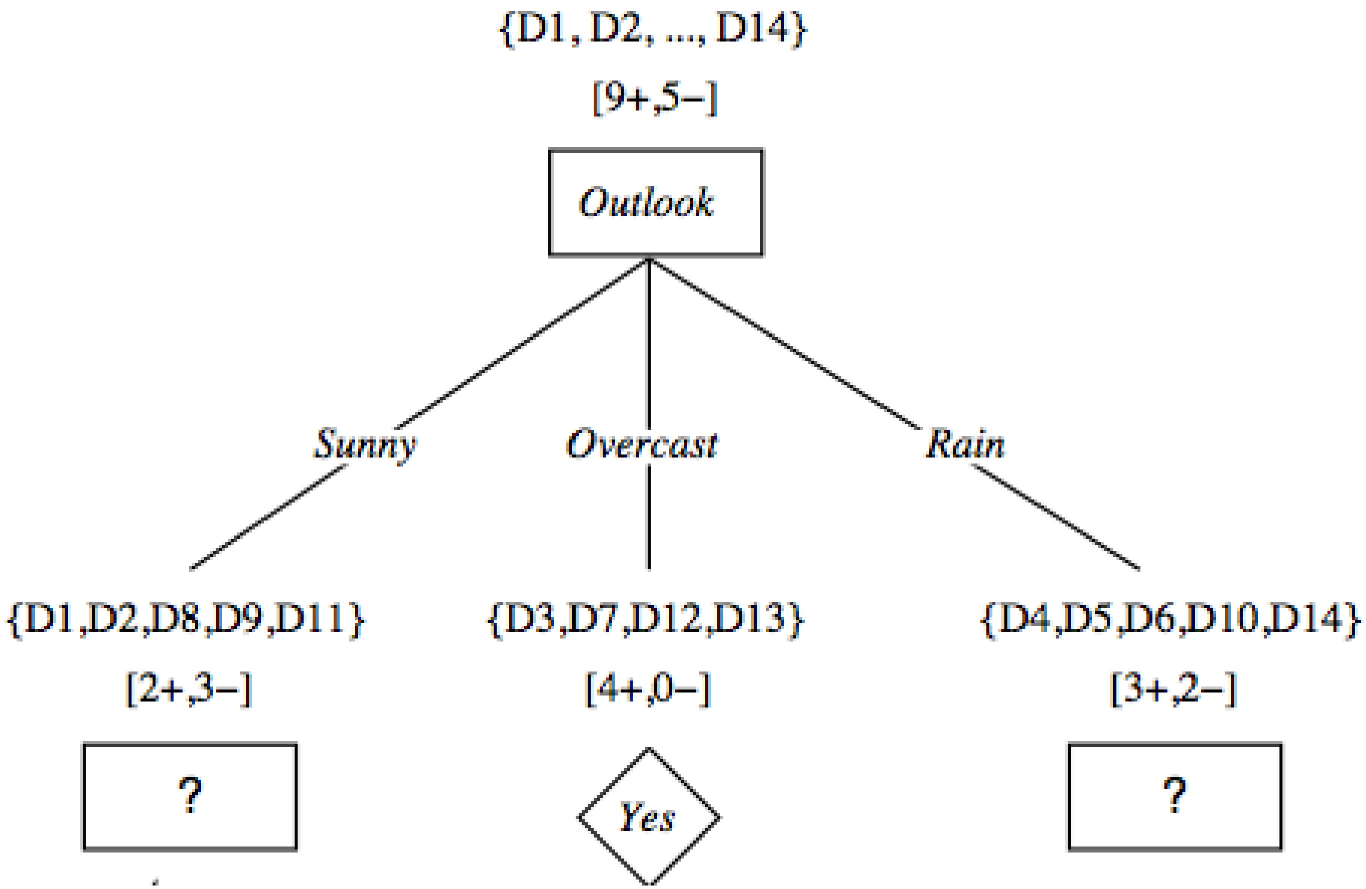
Which attribute is the best classifier?



First step: which attribute to test at the root?

- Which attribute should be tested at the root?
  - $Gain(S, Outlook) = 0.246$
  - $Gain(S, Humidity) = 0.151$
  - $Gain(S, Wind) = 0.048$
  - $Gain(S, Temperature) = 0.029$
- $Outlook$ provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of $Outlook$
  - partition the training samples according to the value of $Outlook$

After first step



Second step

- Working on *Outlook=Sunny* node:

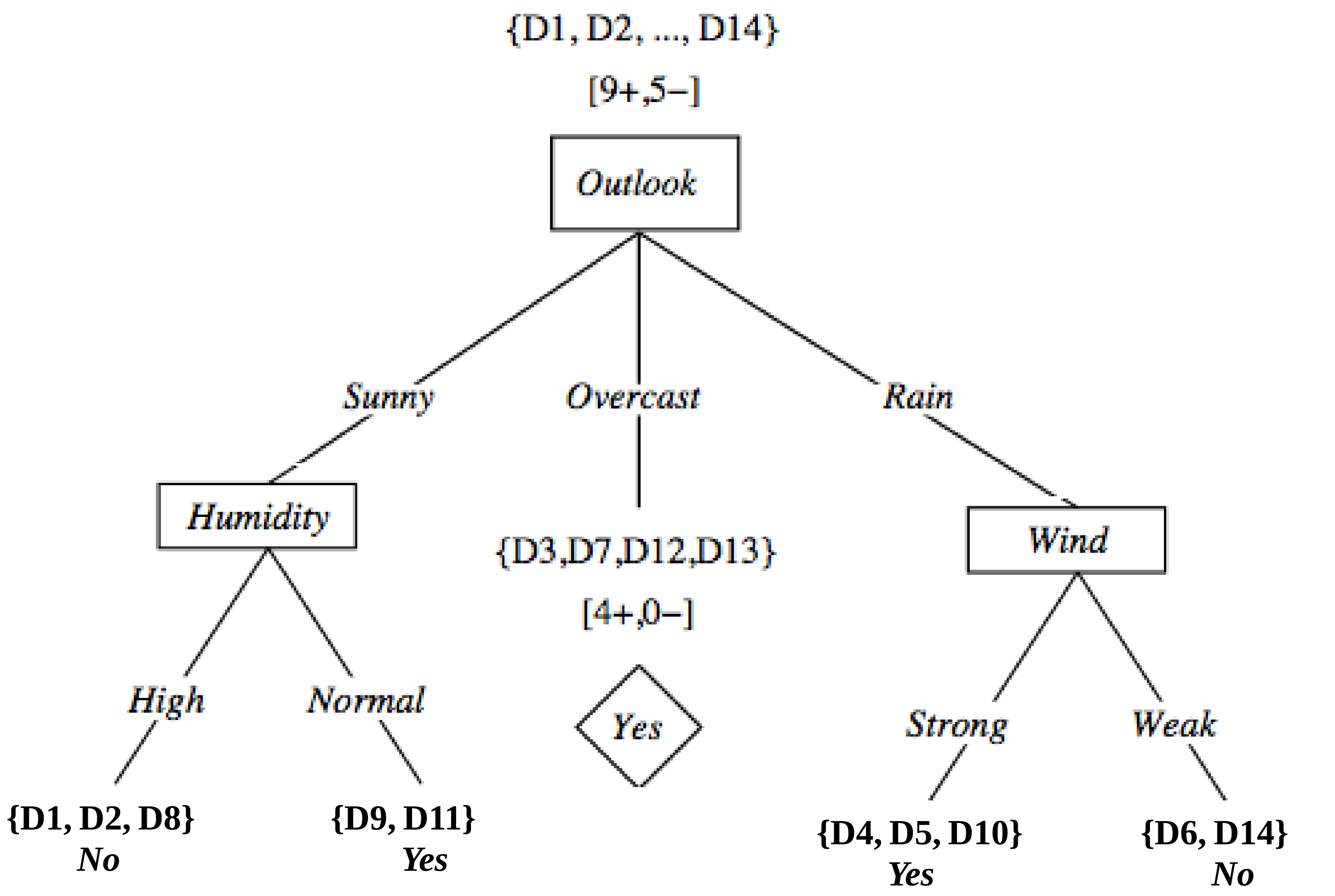
  $$Gain(S_{Sunny}, Humidity) = 0.970 - 3/5 \times 0.0 - 2/5 \times 0.0 = 0.970$$

  $$Gain(S_{Sunny}, Wind) = 0.970 - 2/5 \times 1.0 - 3/5 \times 0.918 = 0\ .019$$

  $$Gain(S_{Sunny}, Temp.) = 0.970 - 2/5 \times 0.0 - 2/5 \times 1.0 - 1/5 \times 0.0 = 0.570$$

- *Humidity* provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Humidity*
  - partition the training samples according to the value of *Humidity*

Second and third steps



Thanks