COMP30027 Machine Learning
Decision Trees

Semester 1, 2019

Jeremy Nicholson & Tim Baldwin & Karin Verspoor

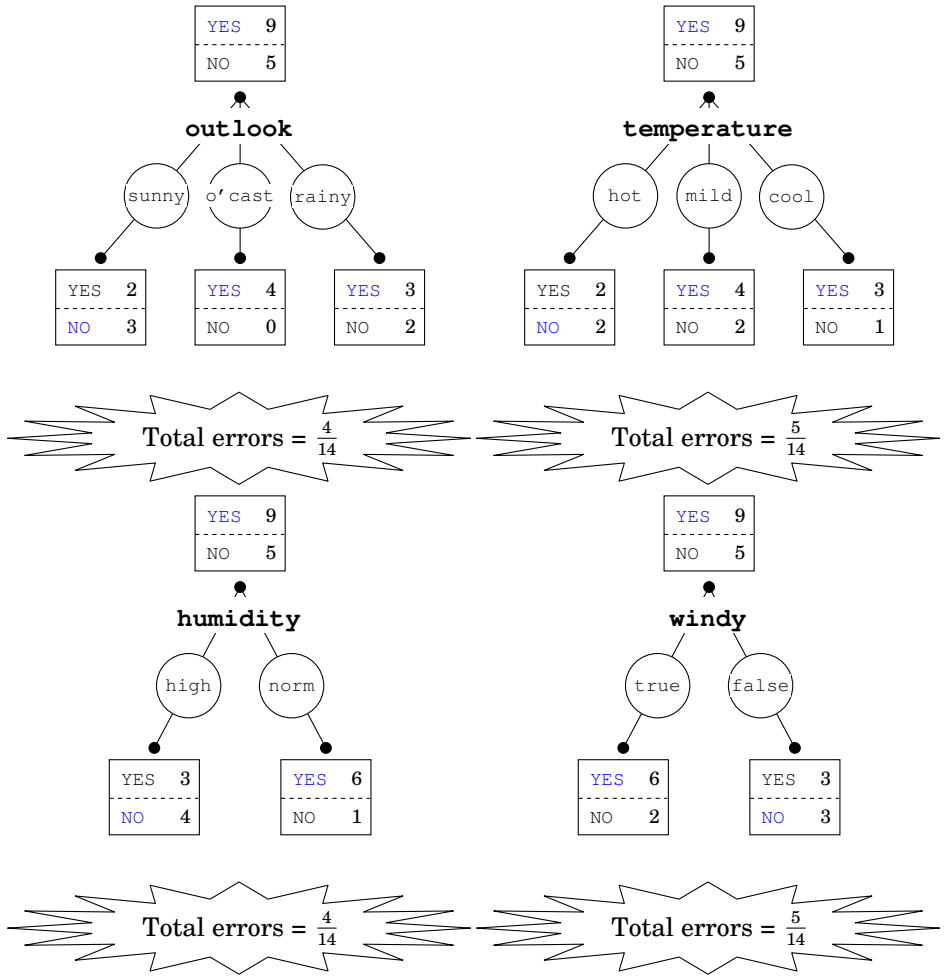
THE UNIVERSITY OF MELBOURNE

Lecture Outline

From Decision Stumps to Decision Trees

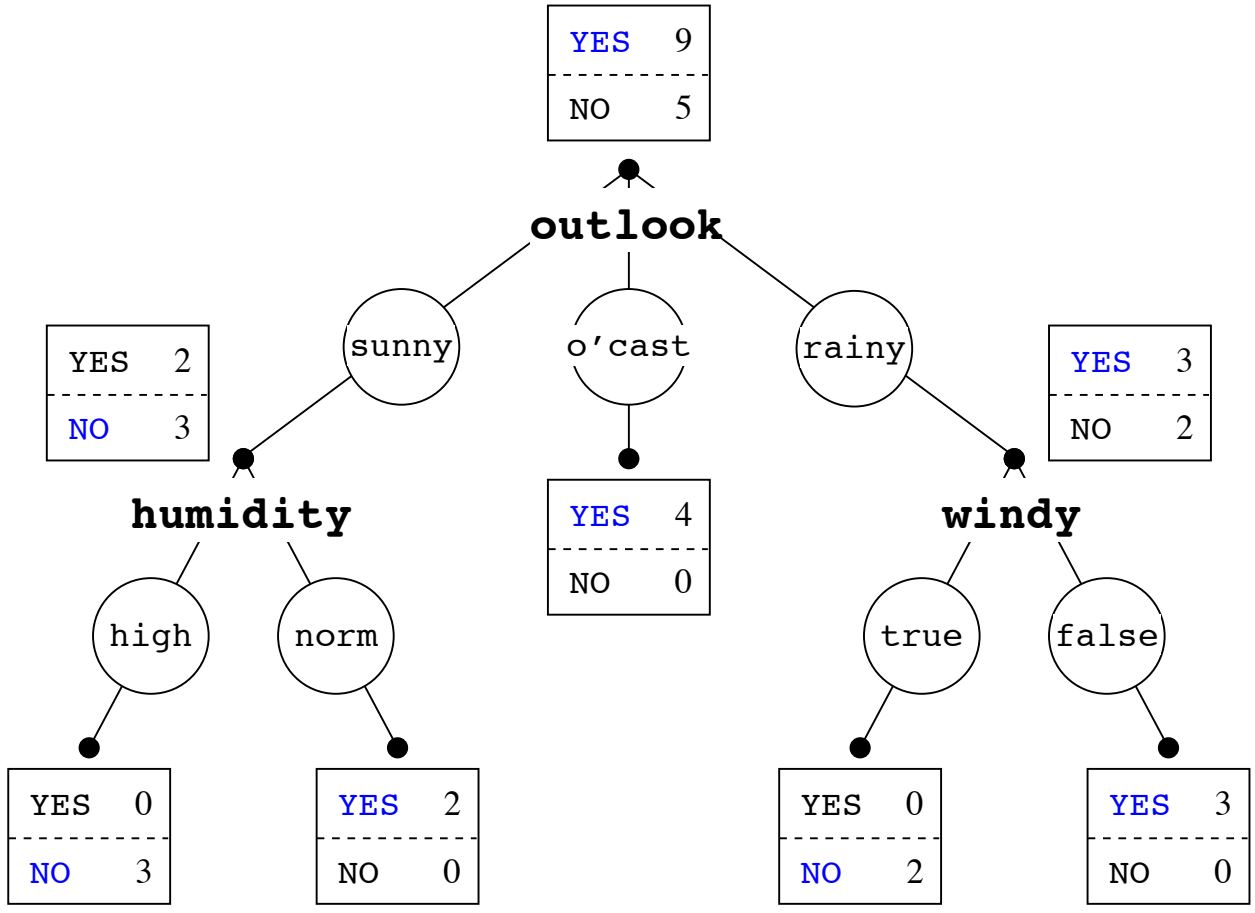
We have seen decision stumps in action in the context of 1-R

Given the obvious myopia of decision stumps, how can we construct **decision trees** (of arbitrary depth) which have the ability to capture complex feature interaction?



Full `weather.nominal` Dataset

	Outlook	Temperature	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



Total errors = $\frac{0}{14}$

Lecture Outline

Constructing Decision Trees: ID3

- **Basic method:** construct decision trees in recursive divide-and-conquer fashion

  FUNCTION ID3 (Root)

  IF all instances at root have same class**

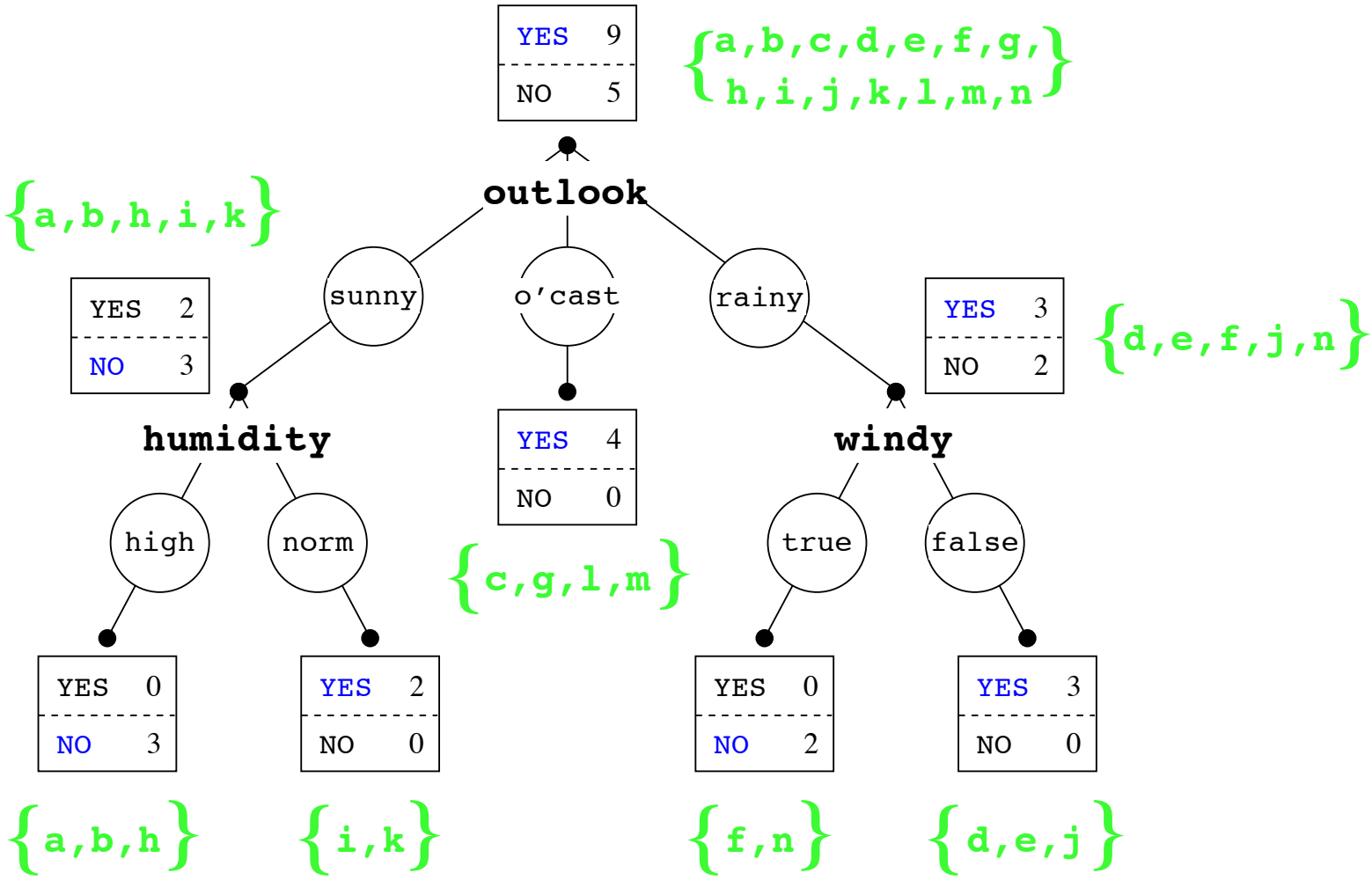
  THEN stop

  ELSE 1. Select a new attribute to use in partitioning root node instances

  2. Create a branch for each attribute value and partition up root node instances according to each value

  3. Call ID3($LEAF_i$) for each leaf node $LEAF_i$

- **This is overly simplified, as we will discuss momentarily

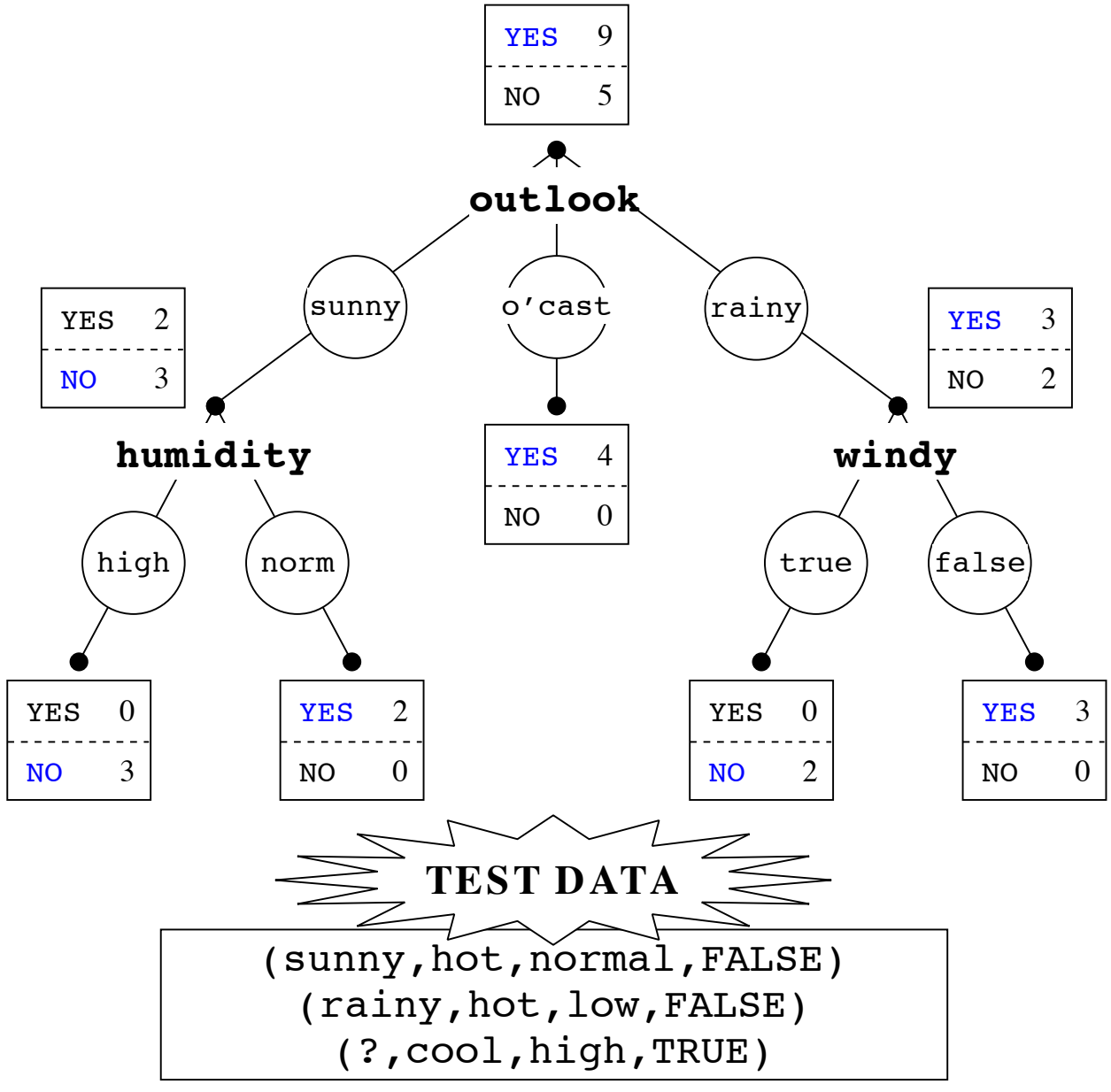

YES 9
NO 5
{a,b,c,d,e,f,g,
h,i,j,k,l,m,n}
outlook
{a,b,h,i,k}
sunny
o'cast
rainy
YES 2
NO 3
YES 3
NO 2
{d,e,f,j,n}
humidity
YES 4
NO 0
windy
high
norm
{c,g,l,m}
true
false
YES 0
NO 3
YES 2
NO 0
YES 0
NO 2
YES 3
NO 0
{a,b,h}
{i,k}
{f,n}
{d,e,j}

Classifying Novel Instances

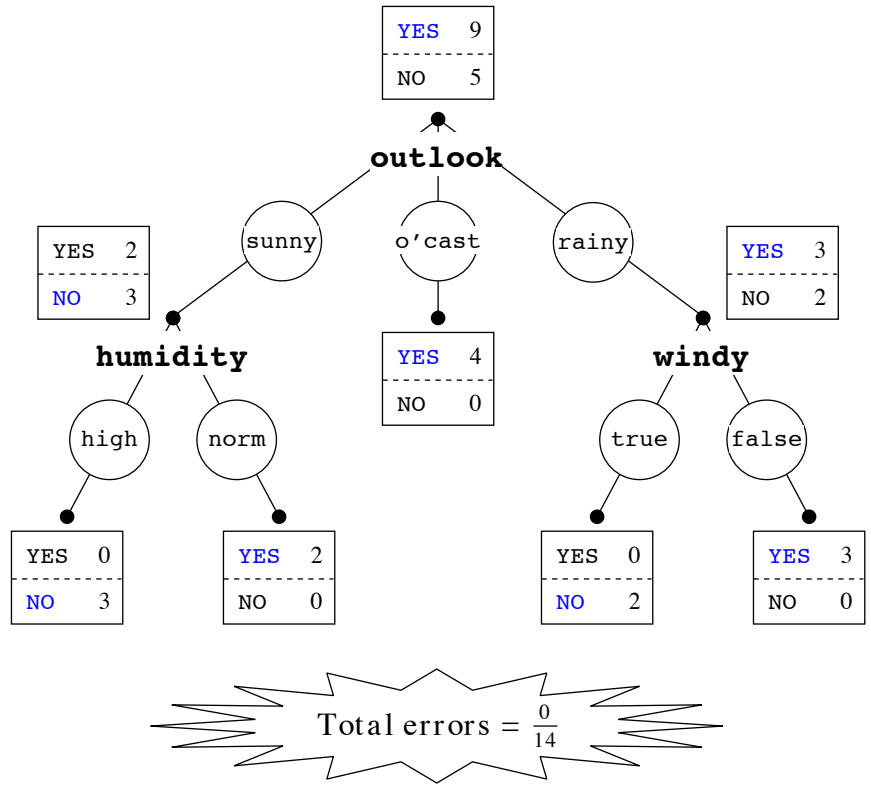
Having constructed the decision tree, we classify novel instances by traversing down the tree and classifying according to the label at the deepest reachable point in the tree structure (leaf)

Complications:

- unobserved attribute–value pairs
- missing values


YES 9
NO 5
outlook
sunny
o'cast
rainy
YES 2
NO 3
humidity
YES 4
NO 0
windy
YES 3
NO 2
high
norm
true
false
YES 0
NO 3
YES 2
NO 0
YES 0
NO 2
YES 3
NO 0
TEST DATA
(sunny,hot,normal,FALSE)
(rainy,hot,low,FALSE)
(?,cool,high,TRUE)

Disjunctive descriptions



Decision Trees can be read as a disjunction; for example, `Yes`:

$$(outlook = sunny \wedge humidity = normal)$$
$$\vee (outlook = overcast)$$
$$\vee (outlook = rainy \wedge windy = false)$$

Criterion for Attribute Selection

How do we choose the attribute to partition the instances at a given node?

We want to get the smallest tree (Occam's Razor; generalisability). Prefer the shortest hypothesis that fits the data.

In favor:

- Fewer short hypotheses than long hypotheses
  - a short hyp. that fits the data unlikely to be a coincidence
  - a long hyp. that fits data might be a coincidence

Against:

- Many ways to define small sets of hypotheses

Entropy (again!) I

- A measure of **unpredictability**
- Given a probability distribution, the information (in bits) required to predict an event is the distribution's **entropy** or **information value**
- The entropy of a discrete random event x with possible states $1, ..n$ is:

$$H(x) = -\sum_{i=1}^{n} P(i) \log_2 P(i)$$

where $0 \log_2 0 =^{def} 0$

Entropy (again!) II

- If most of the probability mass is assigned to a single event:
  - Entropy is low
  - The event is **predictable**
- If the probability mass is evenly divided between many events:
  - Entropy is high
  - The event is **unpredictable**

Entropy (again!) III

In the context of Decision Trees, we are looking at the class distribution at a node:

- 50 Y instances, 5 N instances:

$$\begin{aligned} H &= -[\frac{50}{55}\log_2(\frac{50}{55}) + \frac{5}{55}\log_2(\frac{5}{55})] \\ &\approx 0.44 \text{ bits} \end{aligned}$$

- 30 Y instances, 25 N instances:

$$\begin{aligned} H &= -[\frac{30}{55}\log_2(\frac{30}{55}) + \frac{25}{55}\log_2(\frac{25}{55})] \\ &\approx 0.99 \text{ bits} \end{aligned}$$

We want leaves with **low entropy**!

Information Gain

- The expected reduction in entropy caused by knowing the value of an attribute.
- Compare:
  - the entropy before splitting the tree using the attribute's values
  - the weighted average of the entropy over the children after the split (**Mean Information**)
- If the entropy **decreases**, then we have a better tree (more predictable)

Mean Information Associated with a Decision Stump

- We calculate the mean information for a tree stump with m attribute values as:

$$\text{Mean Info}(x_1, .., x_m) = \sum_{i=1}^{m} P(x_i)H(x_i)$$

where $H(x_i)$ is the entropy of the class distribution for the instances at node x_i

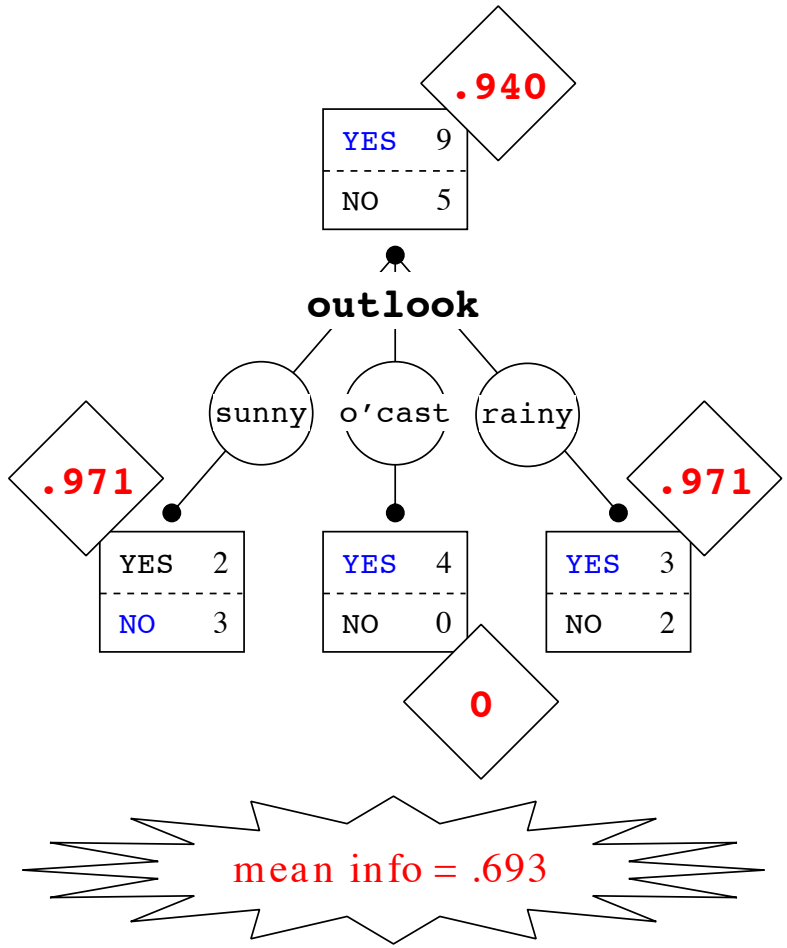
Mean Information (outlook)

$H(rainy) =$
$-(3/5)\log_2(3/5) + (2/5\log_2(2/5))) =$
$-(-0.4422 - 0.5288) = 0.971$

$H(overcast) =$
$-((4/4)\log_2(4/4) + (0/4\log_2(0/4))) = 0$

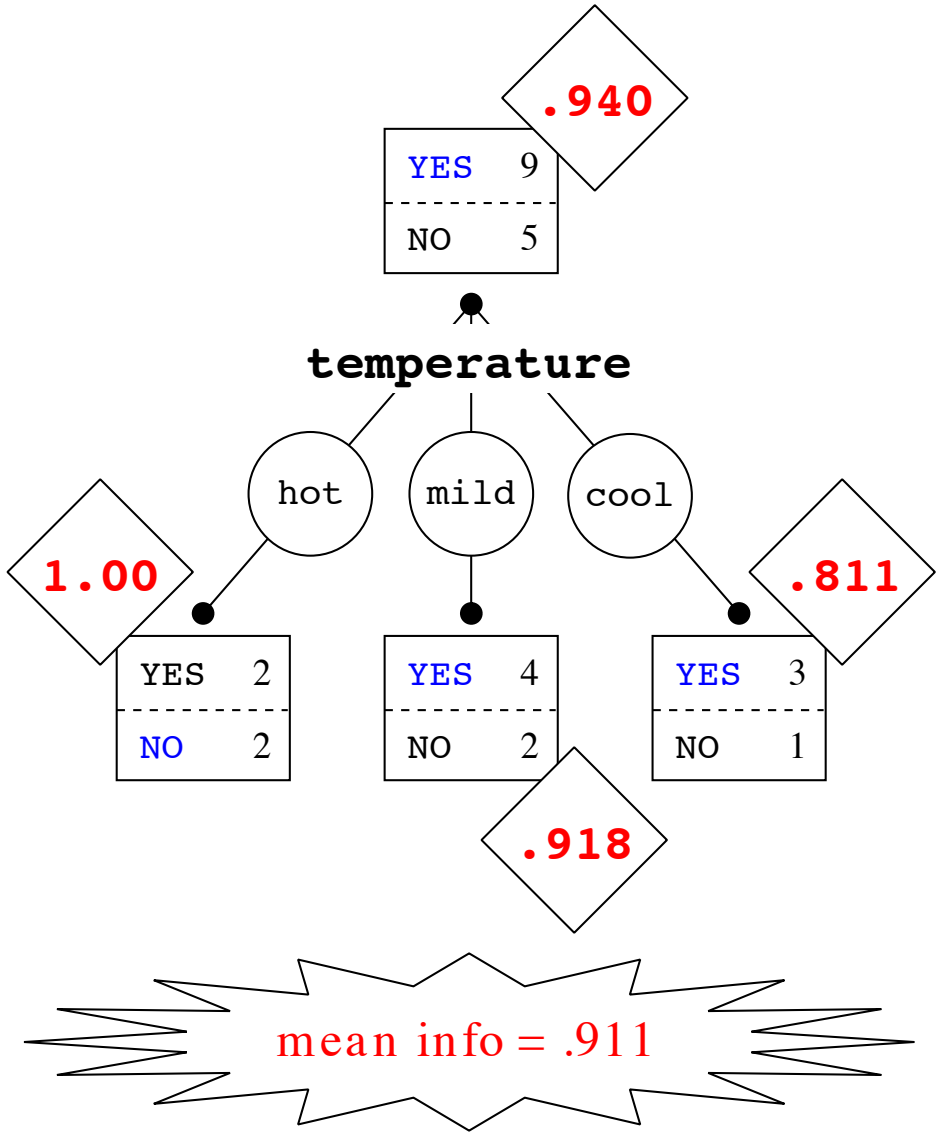
$H(sunny) =$
$-((2/5)\log_2(2/5) + (3/5\log_2(3/5))) = 0.971$

$H(R) =$
$-((9/14)\log_2(9/14) + (5/14)\log_2(5/14)) =$
$-(-.4098 - 0.5305) = 0.940$

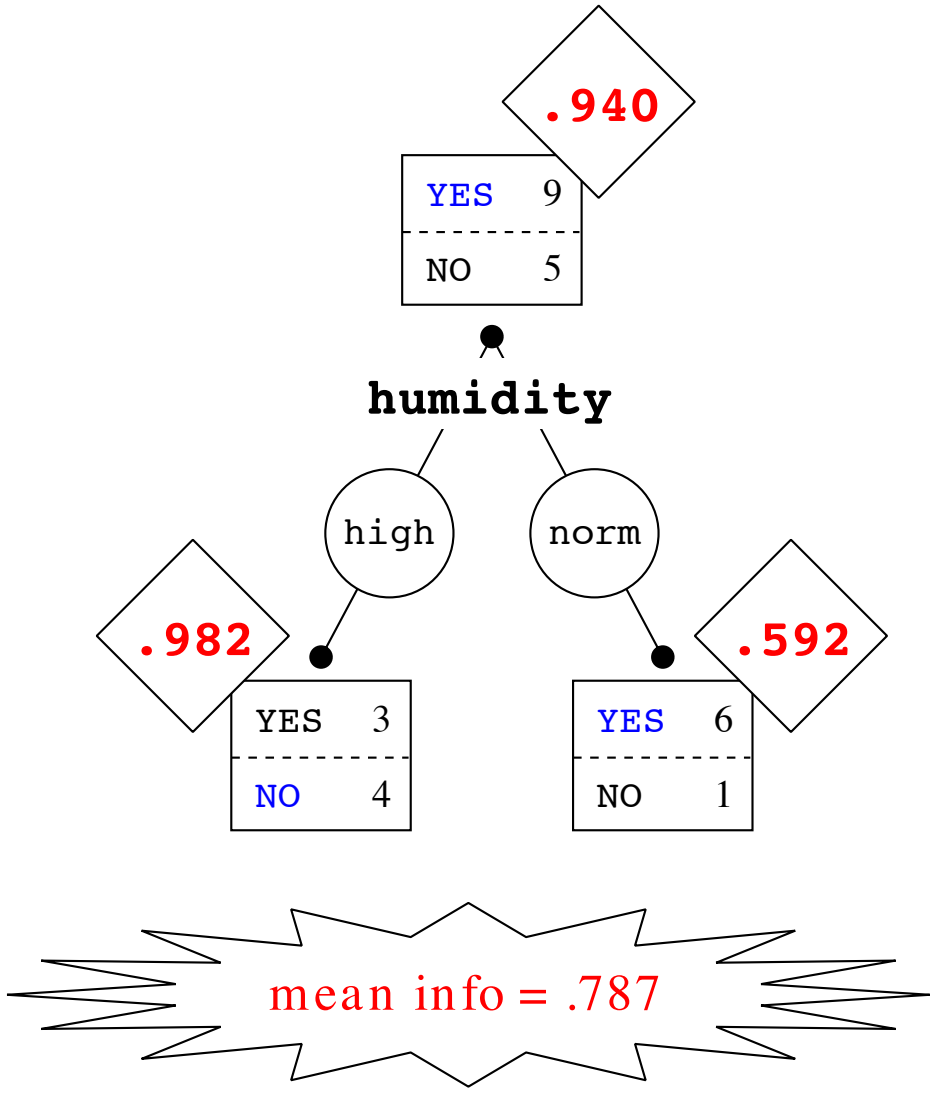


$Mean\_info(outlook) = P(rainy)H(rainy) + P(overcast)H(overcast) +$
$P(sunny)H(sunny) = 5/14 * 0.971 + 0 + 5/14 * 0.971$

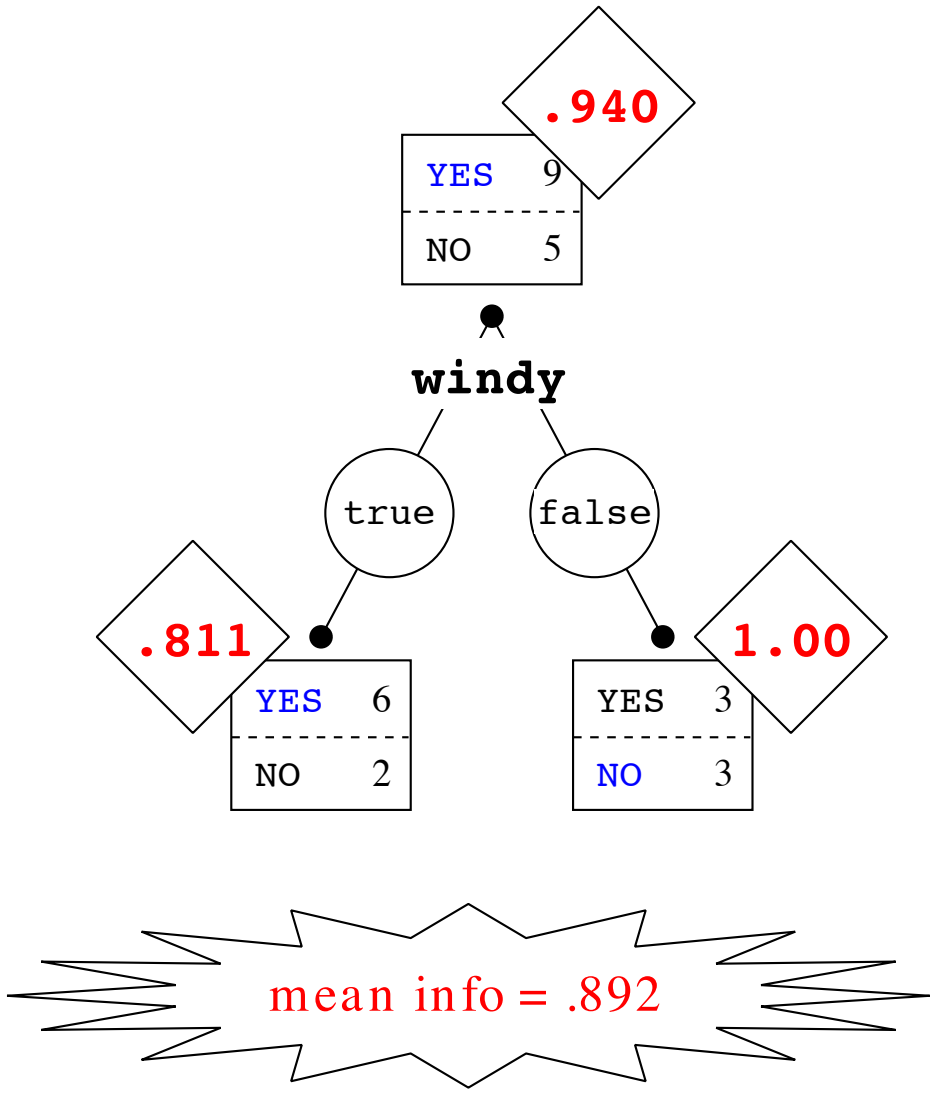
Mean Information (temperature)



Mean Information (humidity)



Mean Information (windy)



Attribute Selection: Information Gain

- We determine which attribute R_A (with values $x_1, ...x_m$) best partitions the instances at a given root node R according to **information gain** (IG):

$$IG(R_A|R) = H(R) - \sum_{i=1}^{m} P(x_i)H(x_i)$$

$$\begin{aligned} IG(outlook|R) &= 0.247 \\ IG(temperature|R) &= 0.029 \\ IG(humidity|R) &= 0.152 \\ IG(windy|R) &= 0.048 \end{aligned}$$

$H(R) = 0.94$
$Mean\_info(outlook) = 0.693$
$Mean\_info(temperature) = 0.911$
$Mean\_info(humidity) = 0.787$
$Mean\_info(windy) = 0.892$

Attribute Selection: Information Gain

- We determine which attribute R_A (with values $x_1, \ldots x_m$) best partitions the instances at a given root node R according to **information gain**:

$$IG(R_A|R) = H(R) - \sum_{i=1}^{m} P(x_i)H(x_i)$$

$$\begin{aligned} IG(outlook|R) &= 0.247 \\ IG(temperature|R) &= 0.029 \\ IG(humidity|R) &= 0.152 \\ IG(windy|R) &= 0.048 \end{aligned}$$

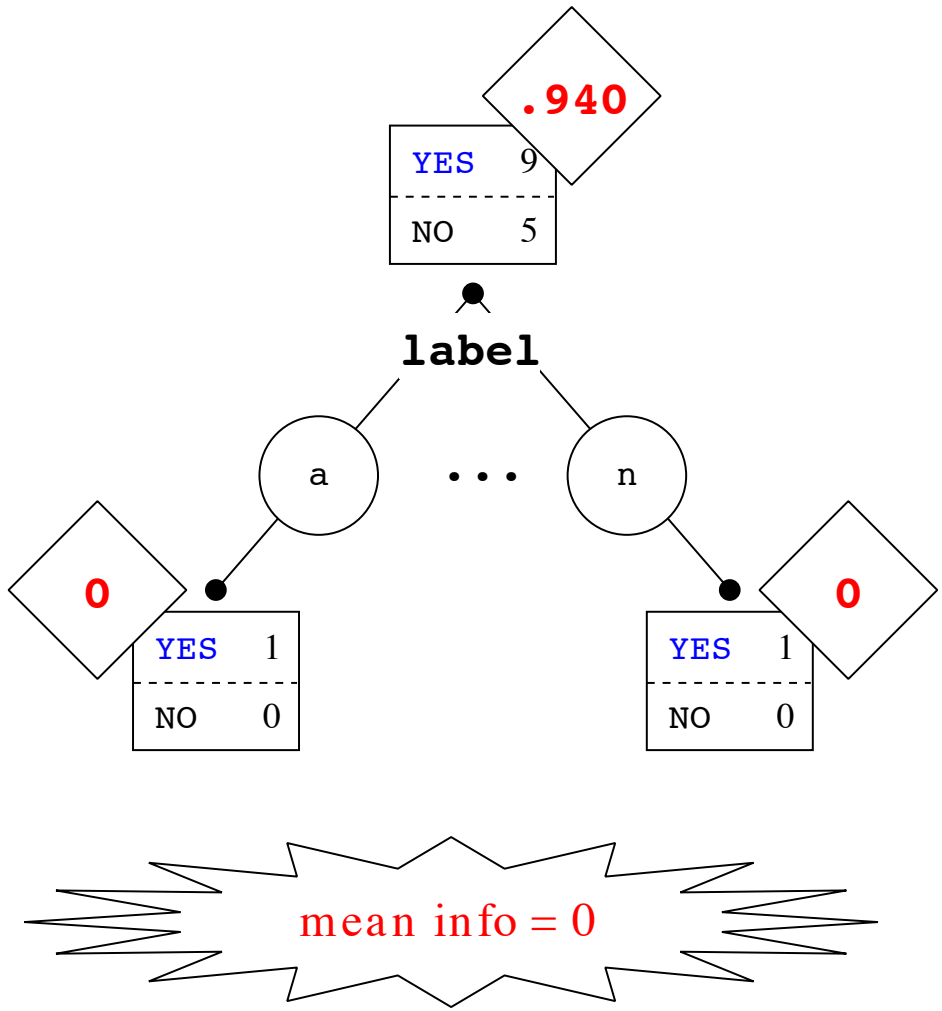
$H(R) = 0.94$
$Mean\_info(outlook) = 0.693$
$Mean\_info(temperature) = 0.911$
$Mean\_info(humidity) = 0.787$
$Mean\_info(windy) = 0.892$

Shortcomings of Information Gain

- Information gain tends to prefer highly-branching attributes:
    - A subset of instances is more likely to be homogeneous (all of a single class) if there are only a few instanes
    - Attribute with many values will have fewer instances at each child node
- This may result in overfitting/fragmentation

Mean Information (label)

- Information gain tends to prefer highly-branching attributes



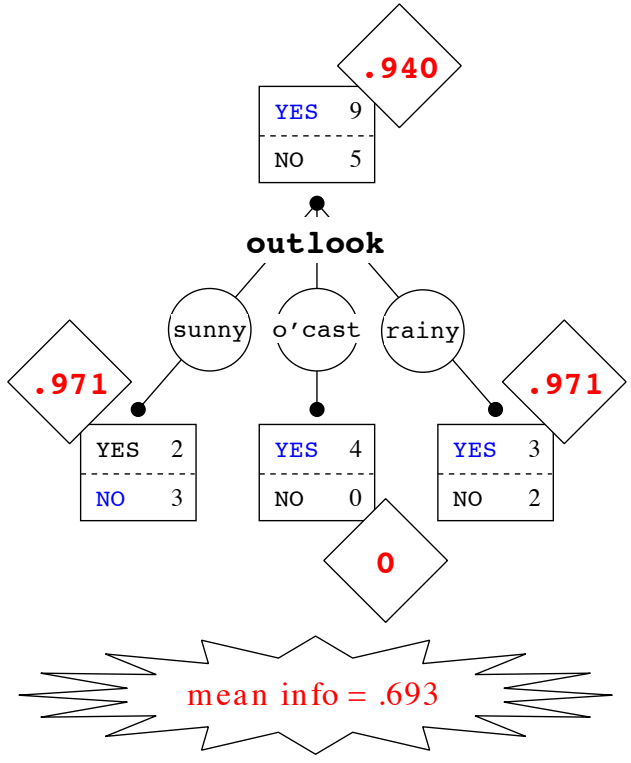
Solution: Gain Ratio

- **Gain ratio (GR)** reduces the bias for information gain towards highly-branching attributes by normalising relative to the **split information**
- **Split info (SI)** is the entropy of a given split (evenness of the distribution of instances to attribute values)

$$\begin{aligned} GR(R_A|R) &= \frac{IG(R_A|R)}{SI(R_A|R)} = \frac{IG(R_A|R)}{H(R_A)} \\ &= \frac{H(R) - \sum_{i=1}^{m} P(x_i)H(x_i)}{-\sum_{i=1}^{m} P(x_i)\log_2 P(x_i)} \end{aligned}$$

- Split Info sometimes called *Intrinsic Value*
- Discourages the selection of attributes with many uniformly distributed values

Split Info



NB: Entropy of distribution of instances to attribute *values* (disregarding classes, unlike Mean Info)

$SI(\texttt{outlook}|R) = -((5/14)\log_2(5/14) + (4/14)\log_2(4/14) + (5/14)\log_2(5/14)) = 1.577$

Gain Ratio: Example

$IG(\texttt{outlook}|R) = 0.247$
$SI(\texttt{outlook}|R) = 1.577$
$GR(\texttt{outlook}|R) = 0.156$

$IG(\texttt{humidity}|R) = 0.152$
$SI(\texttt{humidity}|R) = 1.000$
$GR(\texttt{humidity}|R) = 0.152$

$IG(\texttt{label}|R) = 0.940$
$SI(\texttt{label}|R) = 3.807$
$GR(\texttt{label}|R) = 0.247$

$IG(\texttt{temperature}|R) = 0.029$
$SI(\texttt{temperature}|R) = 1.557$
$GR(\texttt{temperature}|R) = 0.019$

$IG(\texttt{windy}|R) = 0.048$
$SI(\texttt{windy}|R) = 0.985$
$GR(\texttt{windy}|R) = 0.049$

Stopping criteria I

The definition of ID3 above suggests that:

- We recurse until the instances at a node are of the same class
- This is consistent with our usage of entropy: if all of the instances are of a single class, the entropy of the distribution is 0
- Considering other attributes cannot "improve" an entropy of 0 — the Info Gain is 0 by definition

This helps to ensure that the tree remains compact (Occam's Razor)

Stopping criteria II

The definition of ID3 above suggests that:

- The Info Gain/Gain Ratio allows us to choose the (seemingly) best attribute at a given node
- However, it is also an approximate indication of how much absolute improvement we expect from partitioning the data according to the values of a given attribute
- An Info Gain of 0 means that there is no improvement; a very small improvement is often unjustifiable
- Typical modification of ID3: choose best attribute only if IG/GR is greater than some threshold τ
- Other similar approaches use **pruning** — post-process the tree to remove undesirable branches (with few instances, or small IG/GR improvements)

Stopping criteria III

The definition of ID3 above suggests that:

- We might observe improvement through every layer of the tree
- We then run out of attributes, even though one or more leaves could be improved further
- Fall back to majority class label for instances at a leaf with a mixed distribution — unclear what to do with ties
- Possibly can be taken as evidence that the given attributes are insufficient for solving the problem

Lecture Outline

ID3 is an inductive learning algorithm

- ID3 can be characterized as searching a space of hypotheses for one that fits the training examples.
- The hypothesis space searched by ID3 is the set of possible decision trees.
- ID3 performs a simple-to-complex, hill-climbing search through this hypothesis space (with no backtracking), beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.

Practical Properties of ID3 Decision Trees

- Highly regarded among basic supervised learners
- Fast to train, even faster to classify
- Susceptible to the effects of irrelevant features
- Some quirks to account for missing/continuous feature values

Variants of Decision Trees

ID3 is not the only (nor most popular) Decision Tree learner:

- **Oblivious Decision Trees** require the same attribute at every node in a layer
- **Random Tree** only uses a sample of the possible attributes at a given node
  - Helps to account for irrelevant attributes
  - Basis for a better Decision Tree variant: **Random Forest**, of which more later

Summary

- Describe the basic decision tree induction method used in ID3
- What is information gain, how is it calculated and what is its primary shortcoming?
- What is gain ratio, and how does it attempt to overcome the shortcoming of information gain?
- What are the theoretical and practical properties of ID3-style decision trees?

Mitchell, Tom (1997). Machine Learning. Chapter 3: *Decision Tree Learning*.
Tan et al (2006) Introduction to Data Mining. Section 4.3, pp 150-171.