THE UNIVERSITY OF MELBOURNE

Decision Trees

Semester 1, 2021

Ling Luo

Outline

- Introduction
  - One R
  - Decision Trees
- ID3 Algorithm
  - Algorithm
  - Attribute Selection I: Information Gain
  - Attribute Selection II: Gain Ratio
- Discussion

One R (one rule)

- Simple baseline for discrete data classification
- Steps:
  - For each attribute:
    - Assign each level of that attribute to the most likely class.
    - Calculate error rate of this approach.
  - Compare error rates of all attributes and select the attribute with the highest classification accuracy – this attribute is the “one rule”.

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Outlook rule

If it's sunny say yes **2**, no **3**

If it's overcast say yes **4**, no **0**

If it's rainy say yes **3**, no **2**

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Outlook rule

If it's sunny say yes **2**, no **3**

If it's overcast say yes **4**, no **0**

If it's rainy say yes **3**, no **2**

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Outlook rule

If it's sunny say yes **2**, no **3**

If it's overcast say yes **4**, no **0**

If it's rainy say yes **3**, no **2**

Errors $\frac{4}{14}$

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Temperature rule

If it's hot say yes **2**, no **2**

If it's mild say yes **4**, no **2**

If it's cool say yes **3**, no **1**

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Temperature rule

If it's hot say yes **2**, no **2**

If it's mild say yes **4**, no **2**

If it's cool say yes **3**, no **1**

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Temperature rule

If it's hot say yes **2**, no **2**

If it's mild say yes **4**, no **2**

If it's cool say yes **3**, no **1**

Errors $\frac{5}{14}$

One R example

Outlook	Temp	Play?
sunny	hot	no
sunny	hot	no
overcast	hot	yes
rainy	mild	yes
rainy	cool	yes
rainy	cool	no
overcast	cool	yes
sunny	mild	no
sunny	cool	yes
rainy	mild	yes
sunny	mild	yes
overcast	mild	yes
overcast	hot	yes
rainy	mild	no

Outlook rule

Errors

$$\frac{4}{14}$$

Temperature rule

Errors

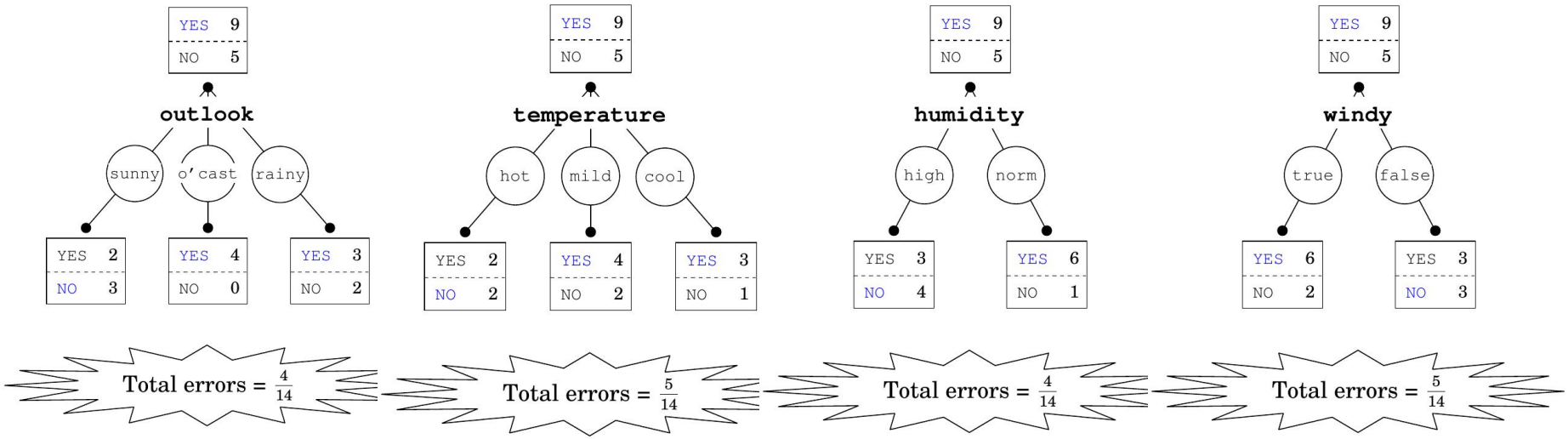
$$\frac{5}{14}$$

The Outlook rule performed better than Temp

Full Dataset: weather.nominal

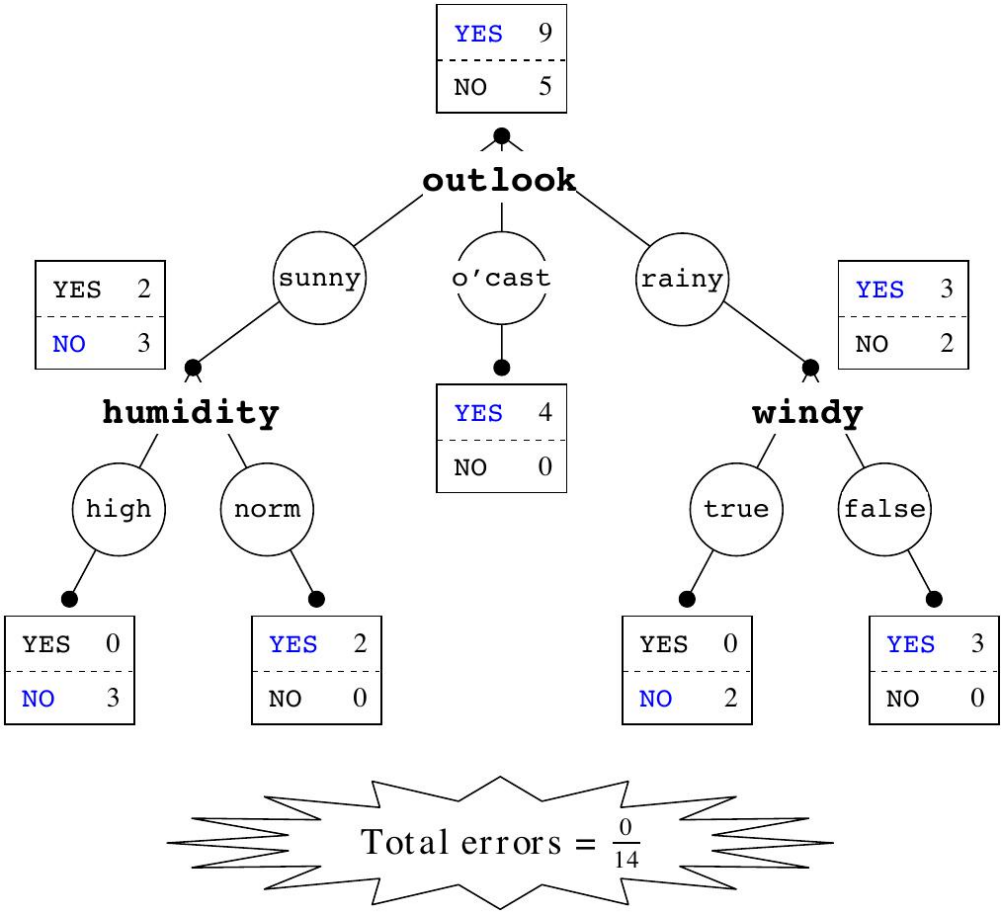
id	Outlook	Temp	Humidity	Windy	Play?
a	sunny	hot	high	FALSE	no
b	sunny	hot	high	TRUE	no
c	overcast	hot	high	FALSE	yes
d	rainy	mild	high	FALSE	yes
e	rainy	cool	normal	FALSE	yes
f	rainy	cool	normal	TRUE	no
g	overcast	cool	normal	TRUE	yes
h	sunny	mild	high	FALSE	no
i	sunny	cool	normal	FALSE	yes
j	rainy	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

From Decision Stumps to Decision Trees



- Limitations?
- One R is a **decision stump**. How can we construct decision trees (of arbitrary depth) which can capture complex feature interaction?

From Decision Stumps to Decision Trees



Constructing Decision Trees: ID3

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

FUNCTION ID3 (*Root*)

IF all instances at *Root* have the same class label*

THEN stop

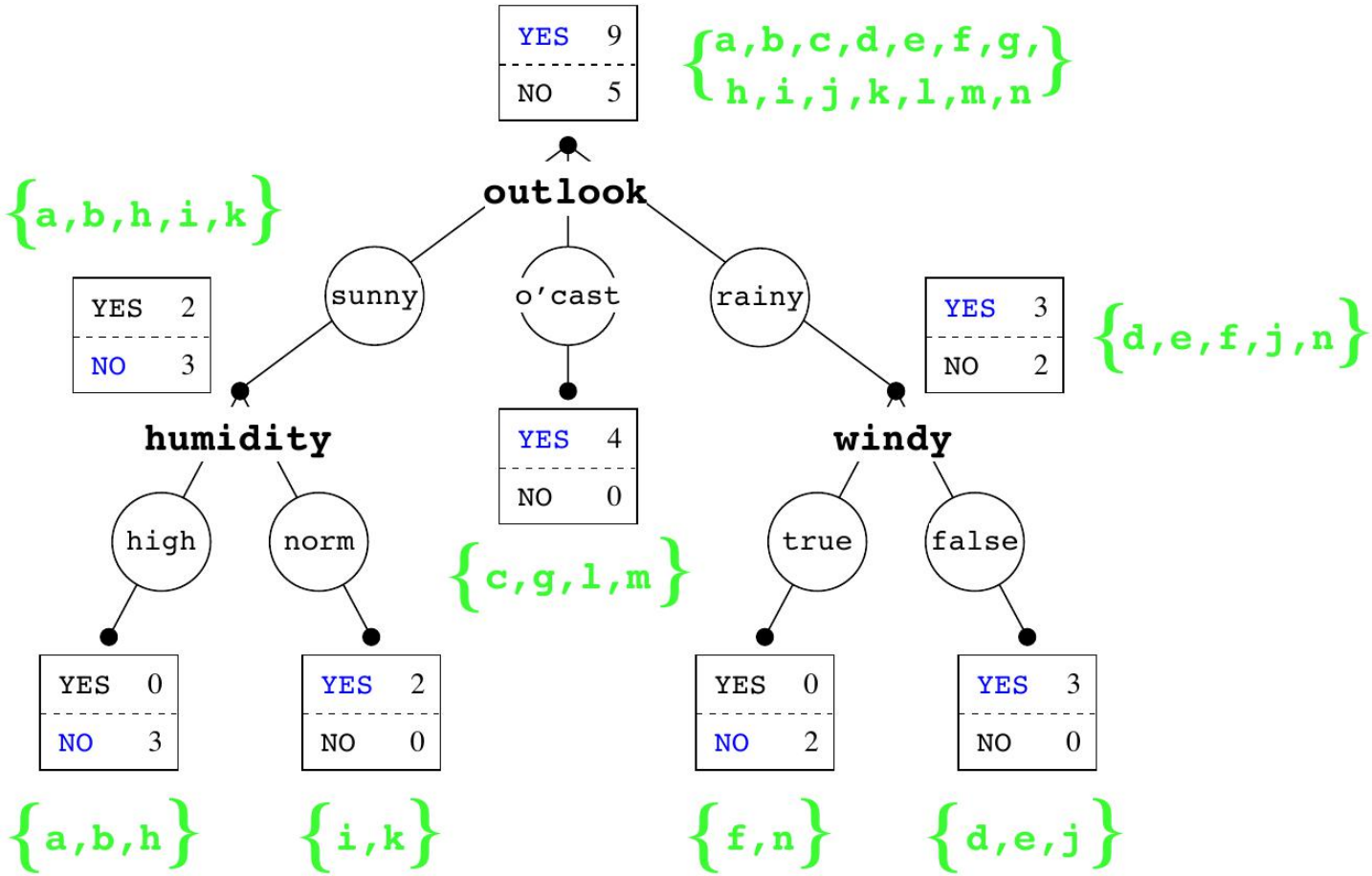
ELSE 1. Select an attribute to use in partitioning *Root* node instances

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 $LEAF_i$

* Note: we may not end up with pure leaves (all instances with the same class label).Therefore, our stopping criterion may actually be a threshold $purity(Root) > \theta$

Decision Tree



Classifying New Instances

- New instance
  - outlook = sunny, temp = hot, humidity = normal, windy = FALSE
- Classify new instances by traversing down the tree and classifying according to the majority class at the deepest reachable point in the tree structure

- Complications:
  - unobserved attribute–value pairs
  - missing values

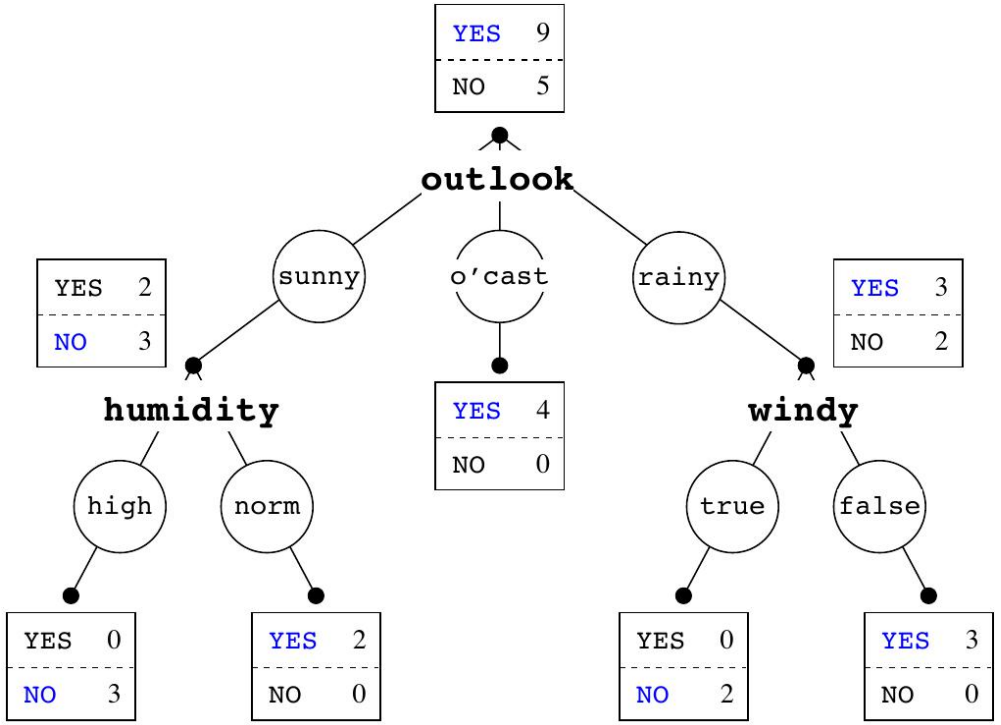
Classifying New Instances

Test Cases

outlook = sunny, temp = hot, humidity = normal, windy = FALSE

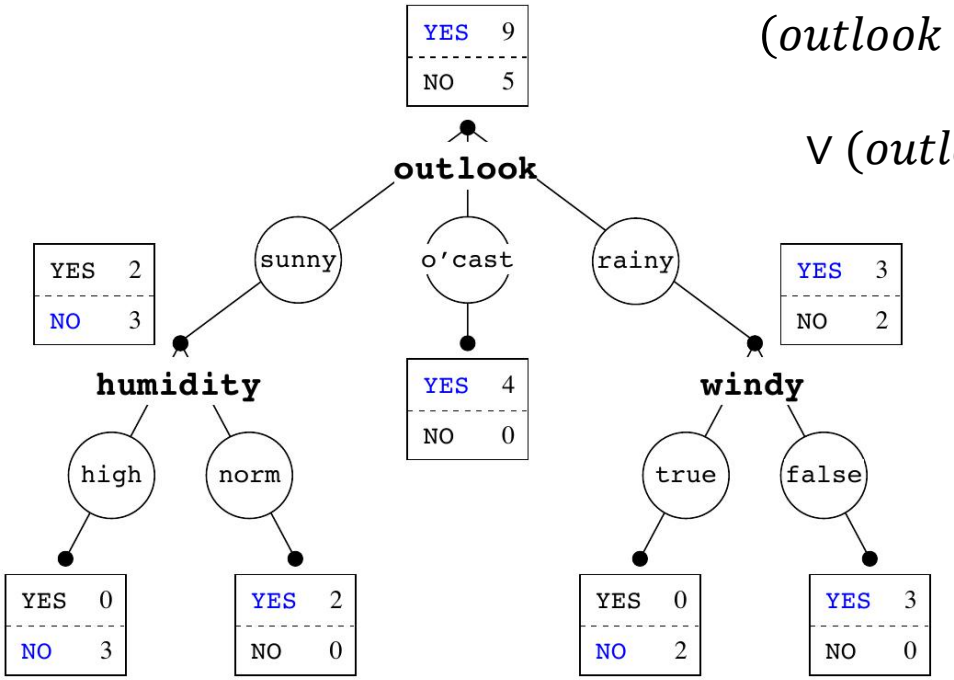
outlook = rainy, temp = hot, humidity = low, windy = FALSE

outlook = ?, temp = cool, humidity = high, windy = TRUE



Disjunctive Descriptions

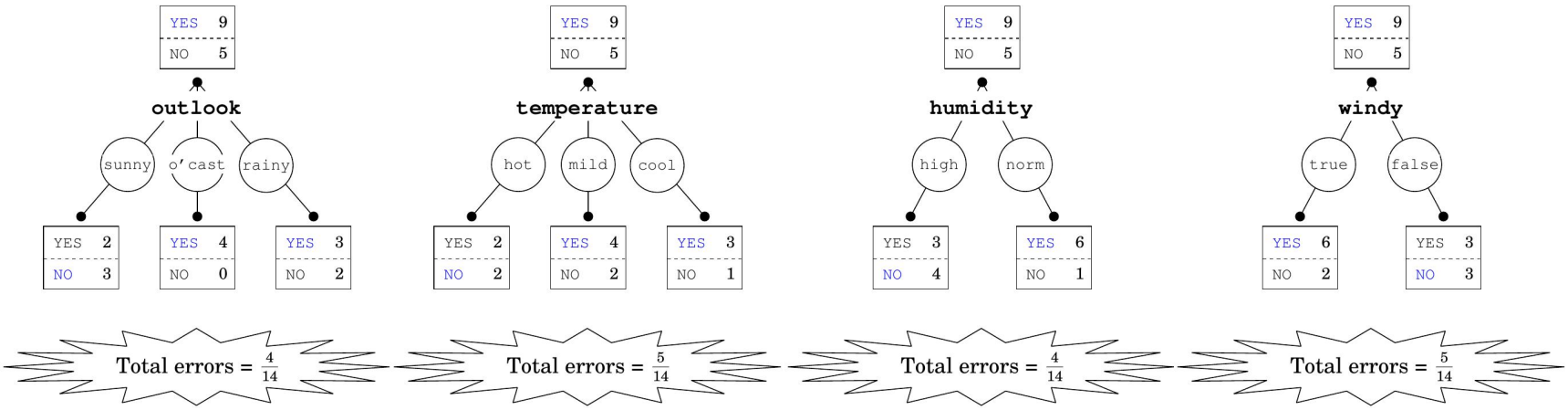
- Decision trees can naturally be read as disjunctive descriptions



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

Criterion for Attribute Selection

- How do we choose the attribute to partition the root node instances?



Entropy (1)

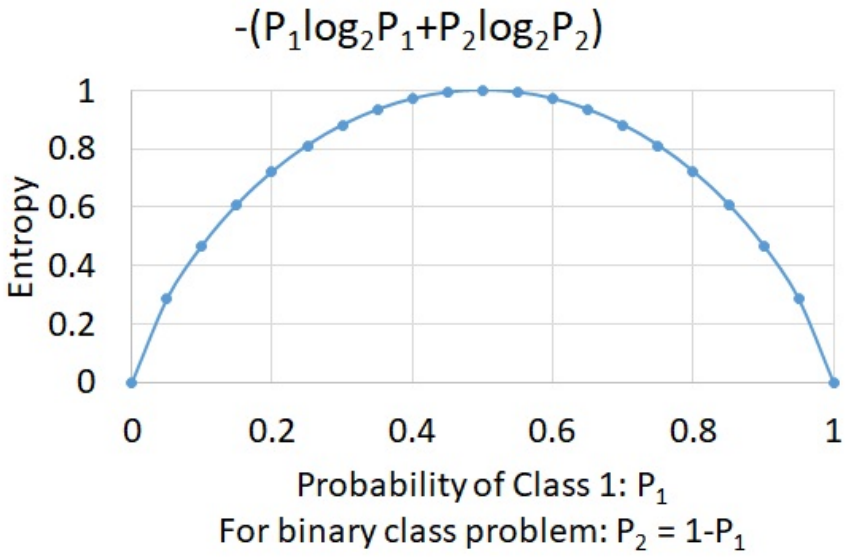
- A measure of **unpredictability**
- Given a probability distribution, the information (in bits) required to predict an event is the distribution's **entropy** or **information value**
- The entropy of a discrete random event x with possible states $1, \ldots, n$ is:

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

where $0\ \log_2 0\ \stackrel{\text{def}}{=}\ 0$

Entropy (2)

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



Entropy (3)

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

- **50 Y instances, 5 N instances**:

$$H = -\left[\frac{50}{55}\log_2\frac{50}{55} + \frac{5}{55}\log_2\frac{5}{55}\right] \approx 0.44\ bits$$

- **30 Y instances, 25 N instances**:

$$H = -\left[\frac{30}{55}\log_2\frac{30}{55} + \frac{25}{55}\log_2\frac{25}{55}\right] \approx 0.99\ bits$$

- 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**
- If the entropy **decreases**, then we have a better tree (more predictable)

Mean Information

- **Weighted average of the entropy** over the children after the split
- The **mean information** for a tree stump with m attribute values as:

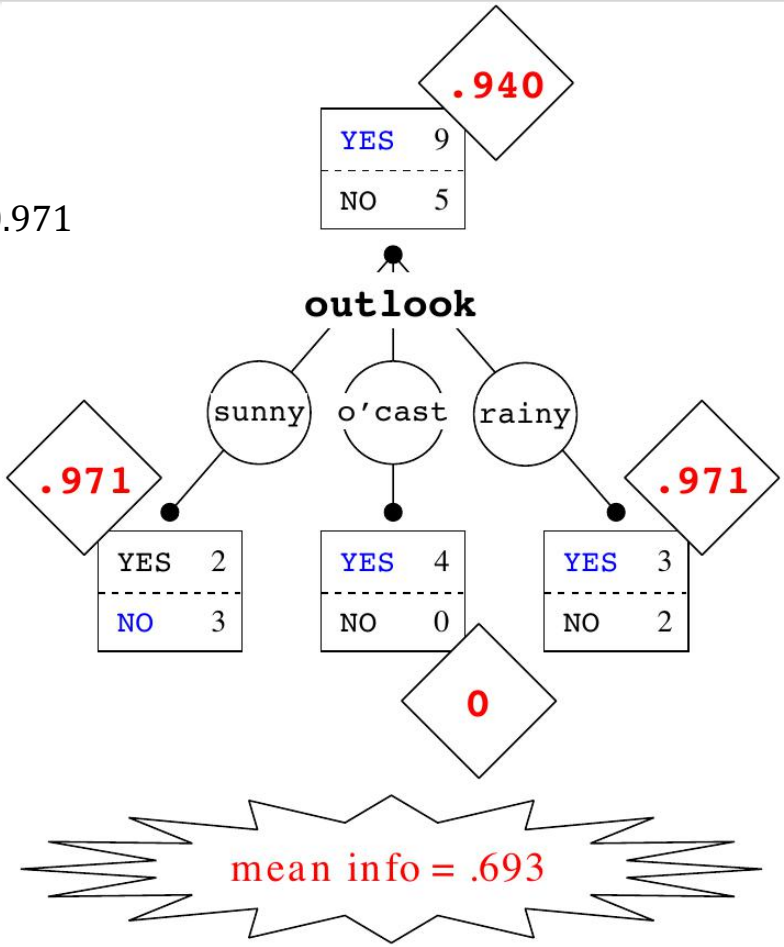
$$\text{Mean Info}(x_1, \ldots, x_m) = \sum_{j=1}^{m} P(x_j)\, H(x_j)$$

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

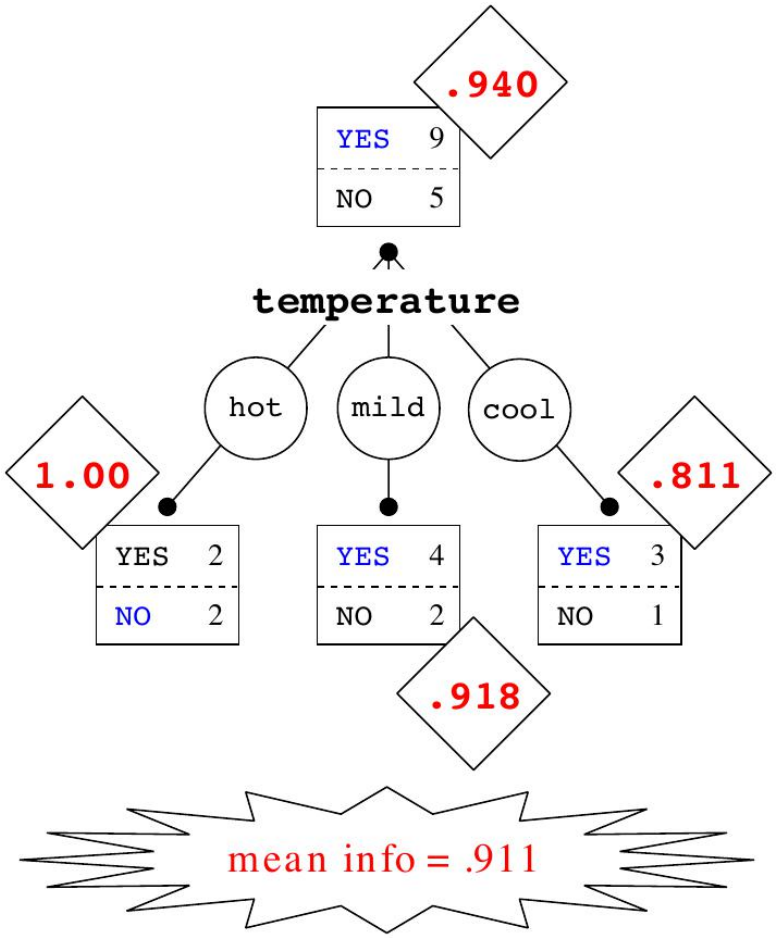
Mean Information (outlook)

- $H(R) = -\left[\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right] = 0.94$
- $H(rainy) = -\left[\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right] = -(-0.4422 - 0.5288) = 0.971$
- $H(overcast) = -\left[\frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4}\right] = 0$
- $H(sunny) = -\left[\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right] = 0.971$
- **MeanInfo** $= P(rainy)H(rainy) + p(overcast)H(overcast) +$

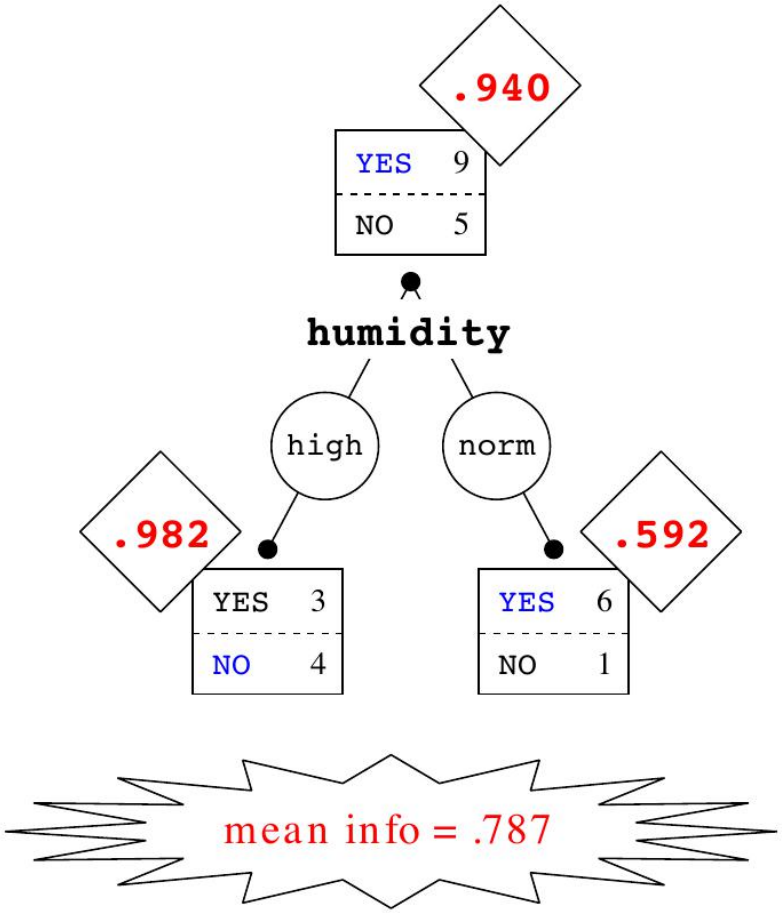
  $P(sunny)H(sunny) = \frac{5}{14} * 0.971 + 0 + \frac{5}{14} * 0.971 = 0.693$



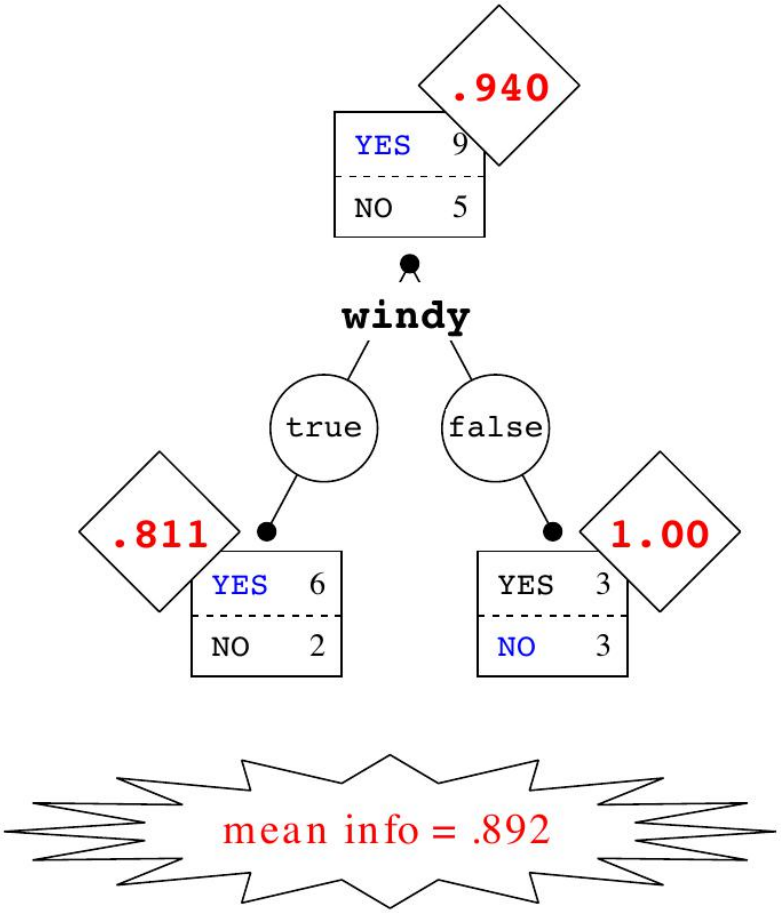
Mean Information (temperature)



Mean Information (humidity)



Mean Information (windy)



Attribution Selection: Information Gain

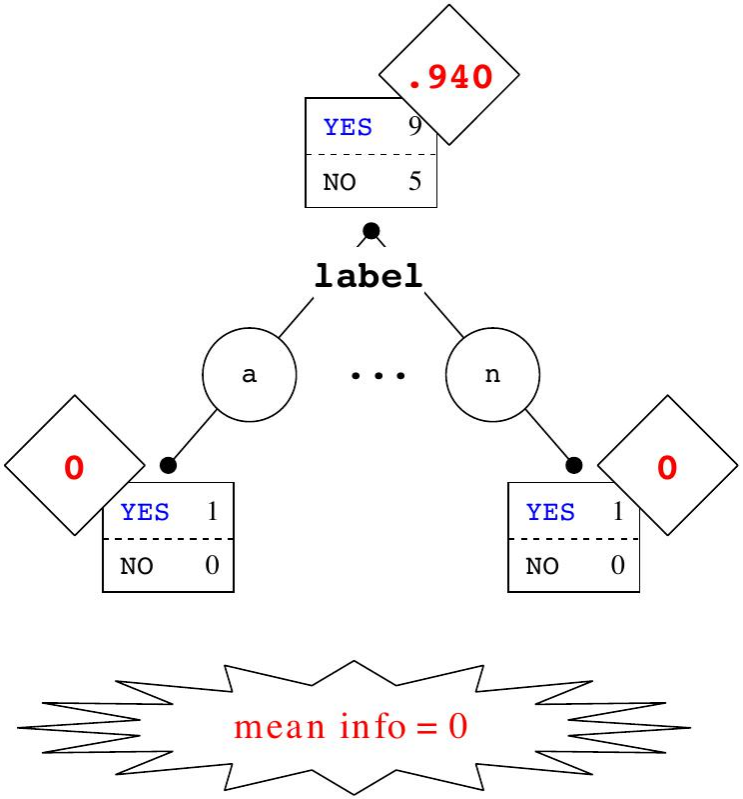
- Select attribute R_A (with values $x_1, \ldots, x_m$) best splits the instances at a given root node R according to information gain:

$$IG(R_A|R) = H(R) - \sum_{j=1}^{m} P(x_j)\, H(x_j)$$

$H(R) = 0.94$	
$\text{MeanInfo}(outlook) = 0.693$	$IG(outlook\|R) = 0.247$
$\text{MeanInfo}(temp) = 0.991$	$IG(temp\|R) = 0.029$
$\text{MeanInfo}(humidity) = 0.787$	$IG(humidity\|R) = 0.152$
$\text{MeanInfo}(windy) = 0.892$	$IG(windy\|R) = 0.048$

Mean Information (id label)

- What if we split instances using ID label?



IG = 0.94!

Information gain tends to prefer highly-branching attributes

Shortcomings of Information Gain

- Information gain tends to prefer highly-branching attributes
- Subsets are more likely to be pure (above a purity threshold) if there is a large number of attribute values
- This may result in overfitting/fragmentation

Gain Ratio

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

$$GR(R_A|R) = \frac{IG(R_A \mid R)}{SI(R_A \mid R)} = \frac{IG(R_A \mid R)}{H(R_A)}$$

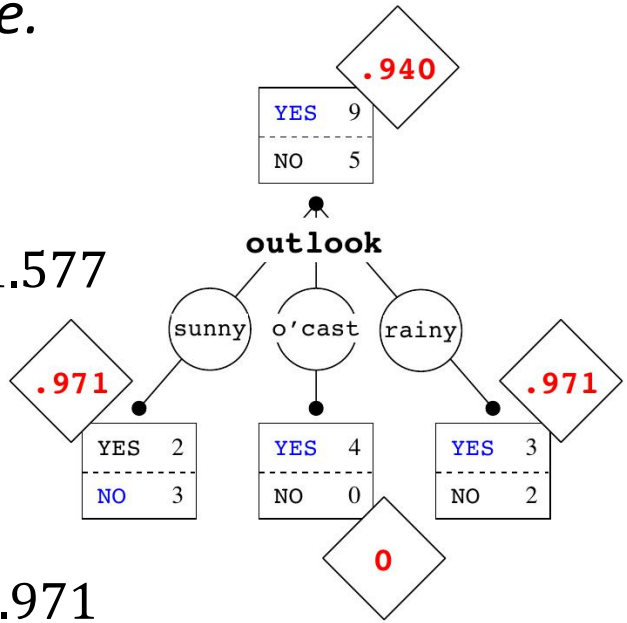
$$= \frac{H(R) - \sum_{j=1}^{m} P(x_j)\, H(x_j)}{-\sum_{j=1}^{m} P(x_j)\, \log_2 P(x_j)}$$

- Discourages the selection of attributes with many uniformly distributed values

Split Information

- Entropy of the attribute values, not with respect to target classes. Also called *Intrinsic Value.*

$$SI(outlook|R) = -\left[\frac{5}{14}\log_2\frac{5}{14} + \frac{4}{14}\log_2\frac{4}{14} + \frac{5}{14}\log_2\frac{5}{14}\right] = 1.577$$



- Contrast with mean information

$$\text{MeanInfo}(outlook) = \frac{5}{14} * 0.971 + 0 + \frac{5}{14} * 0.971 = 0.693$$

Gain Ratio: Example

IG(outlook|R) = 0.247

SI(outlook|R) = 1.577

GR(outlook|R) = 0.156

IG(temp|R) = 0.029

SI(temp|R) = 1.557

GR(temp|R) = 0.019

IG(humidity|R) = 0.152

SI(humidity|R) = 1.000

GR(humidity|R) = 0.152

IG(windy|R) = 0.048

SI(windy|R) = 0.985

GR(windy|R) = 0.049

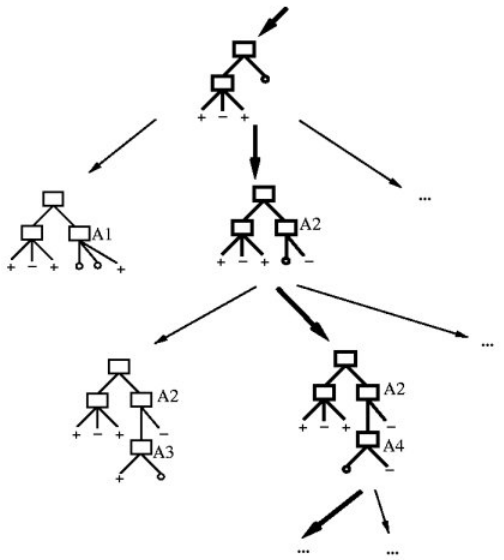
IG(label|R) = 0.940

SI(label|R) = 3.807

GR(label|R) = 0.247

Discussion of ID3 (1)

- ID3 is an inductive learning algorithm
- ID3 searches a space of hypotheses for one that fits the training examples.
- The hypothesis space is the set of possible decision trees.
- ID3 performs a simple-to-complex, hill-climbing search through this hypothesis space
  - beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data
  - no backtracking



Discussion of ID3 (2)

- We want to get the smallest tree (Occam's Razor; generalisability).
- Prefer the shortest hypothesis that fits the data.
  - a short hyp. that fits the data is unlikely to be a coincidence
  - a long hyp. that fits data might be a coincidence

Practical Properties of ID3

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

Variants of Decision Trees

- **Random Trees:** use a sample of the possible attributes at a given node
  - Helps to account for irrelevant attributes
  - Basis for Random Forest, of which more later
- **C4.5:** extension of ID3
  - Handles both continuous and discrete values
  - Prune branches to avoid overfitting

Summary

- 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?

Readings

- Mitchell, Tom (1997). Machine Learning. Chapter 3: Decision Tree Learning.
- Tan et al. Introduction to Data Mining (2018, 2nd edition). Section 3.3, Decision Tree Classifier.