



Decision Trees

By

Amritansh

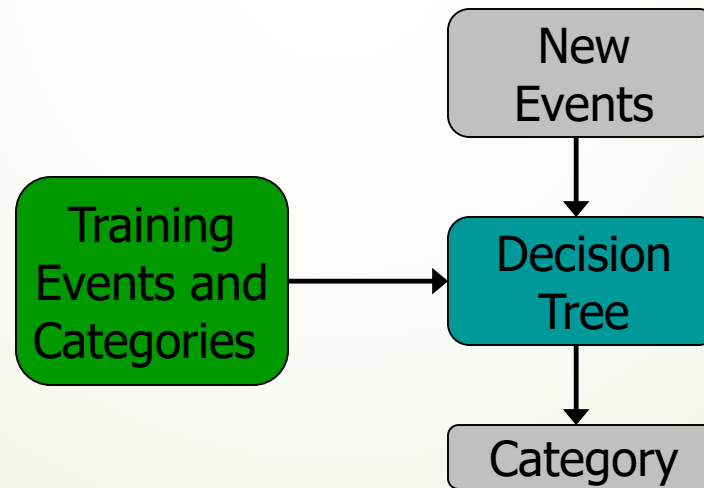
Introduction

- Given an event, predict its category.
Examples:
 - Who won a given ball game?
 - How should we file a given email?
 - What word sense was intended for a given occurrence of a word?
- Event = list of features. Examples:
 - Ball game: Which players were on offense?
 - Email: Who sent the email?
 - Disambiguation: What was the preceding word?

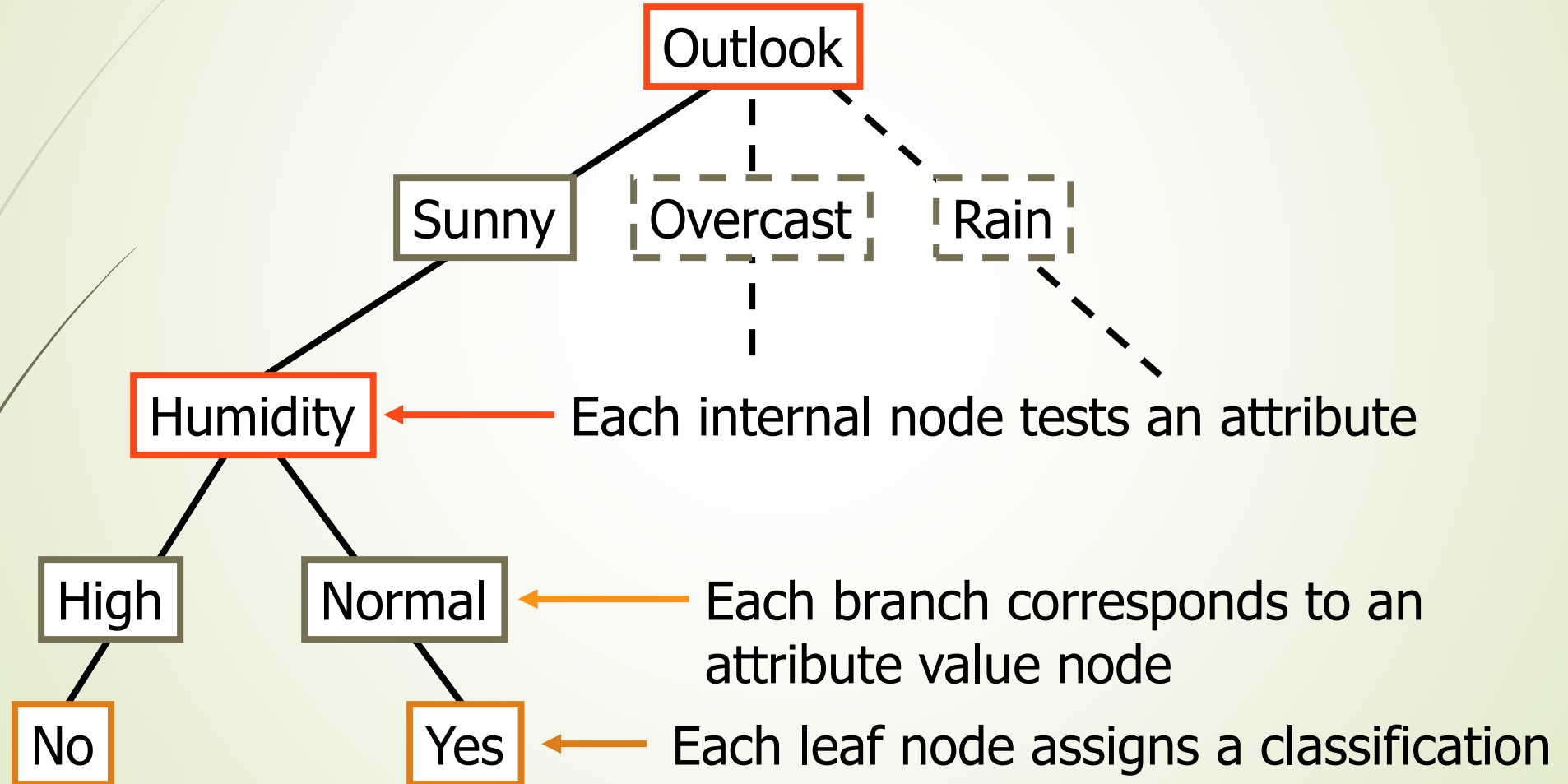
Introduction

3

- Use a decision tree to predict categories for new events.
- Use training data to build the decision tree.



Decision Tree for PlayTennis



Word Sense Disambiguation

- Given an occurrence of a word, decide which sense, or meaning, was intended.
- Example: "run"
 - run1: move swiftly (I ran to the store.)
 - run2: operate (I run a store.)
 - run3: flow (Water runs from the spring.)
 - run4: length of torn stitches (Her stockings had a run.)
 - etc.

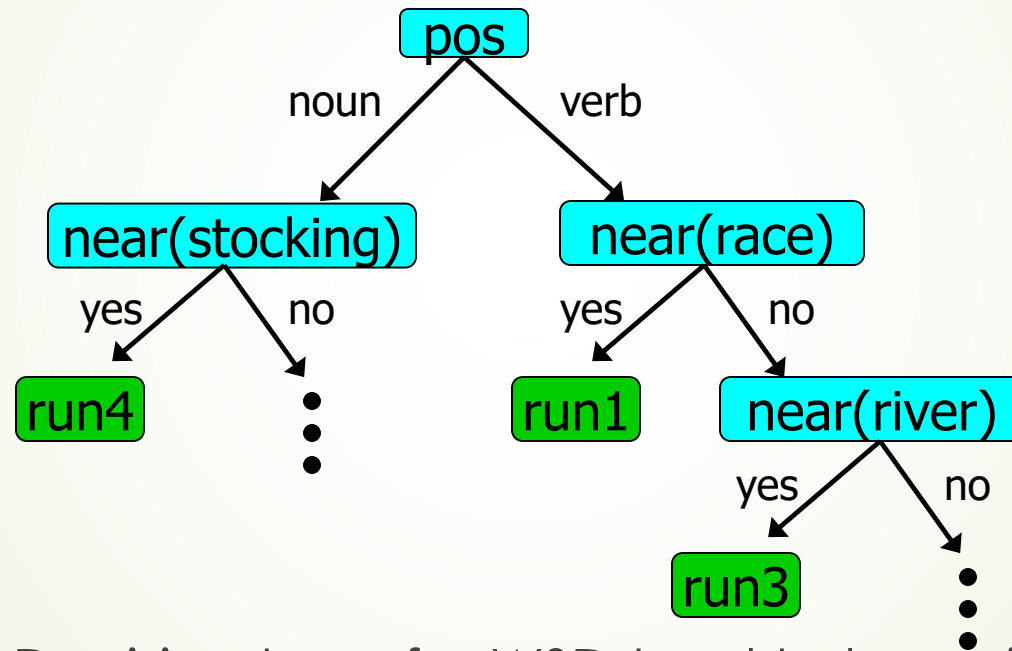
Word Sense Disambiguation

- Categories
 - Use **word sense labels** (run1, run2, etc.) to name the possible categories.
- Features
 - Features describe the *context* of the word we want to disambiguate.
 - Possible features include:
 - **near(w)**: is the given word near an occurrence of word w?
 - **pos**: the word's part of speech
 - **left(w)**: is the word immediately preceded by the word w?
 - etc.

Word Sense Disambiguation

7

➤ Example decision tree:



(Note: Decision trees for WSD tend to be quite large)

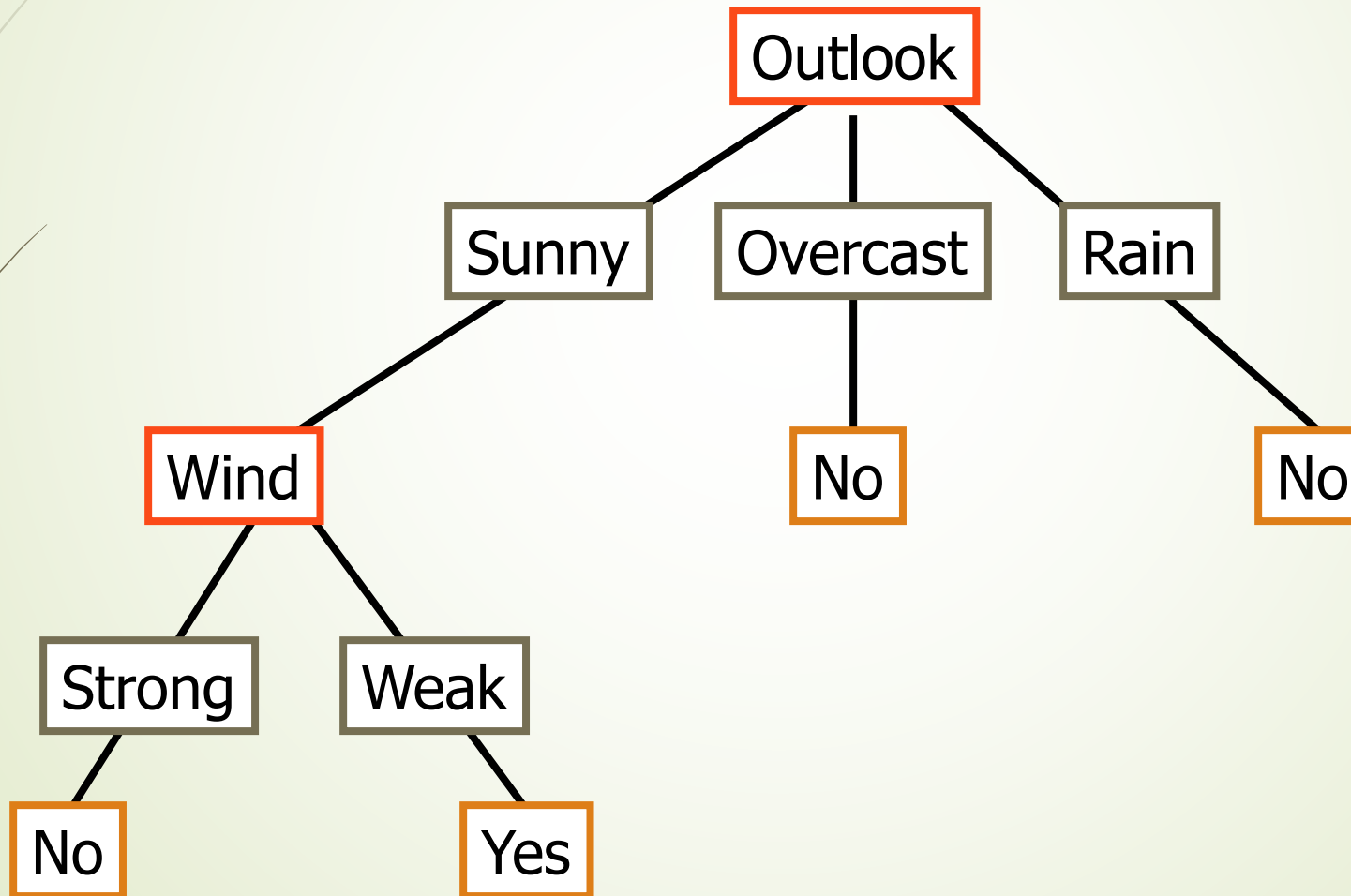
WSD: Sample Training Data

8

Features				Word Sense
pos	near(race)	near(river)	near(stockings)	
noun	no	no	no	run4
verb	no	no	no	run1
verb	no	yes	no	run3
noun	yes	yes	yes	run4
verb	no	no	yes	run1
verb	yes	yes	no	run2
verb	no	yes	yes	run3

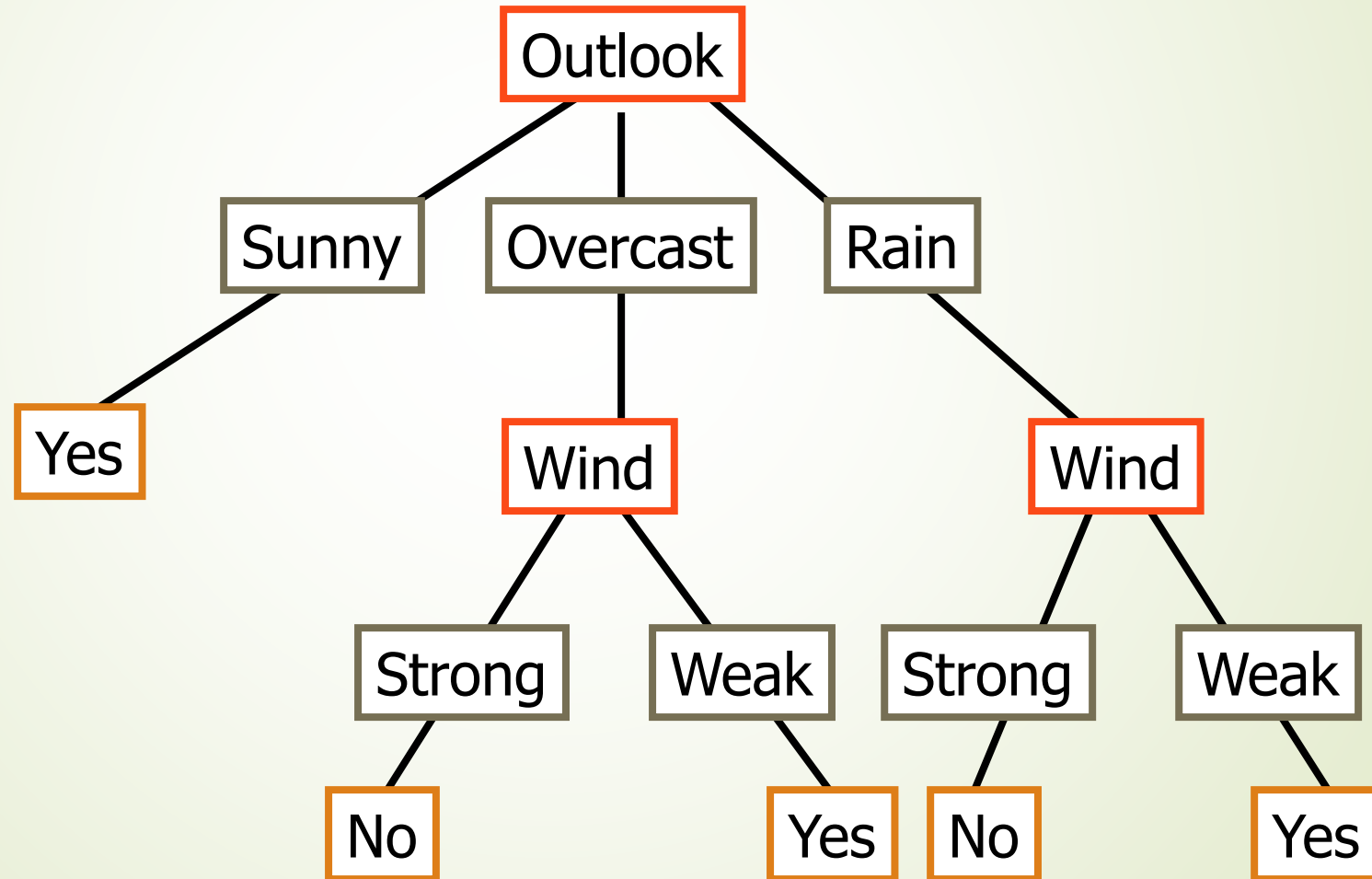
Decision Tree for Conjunction

Outlook=Sunny \wedge Wind=Weak



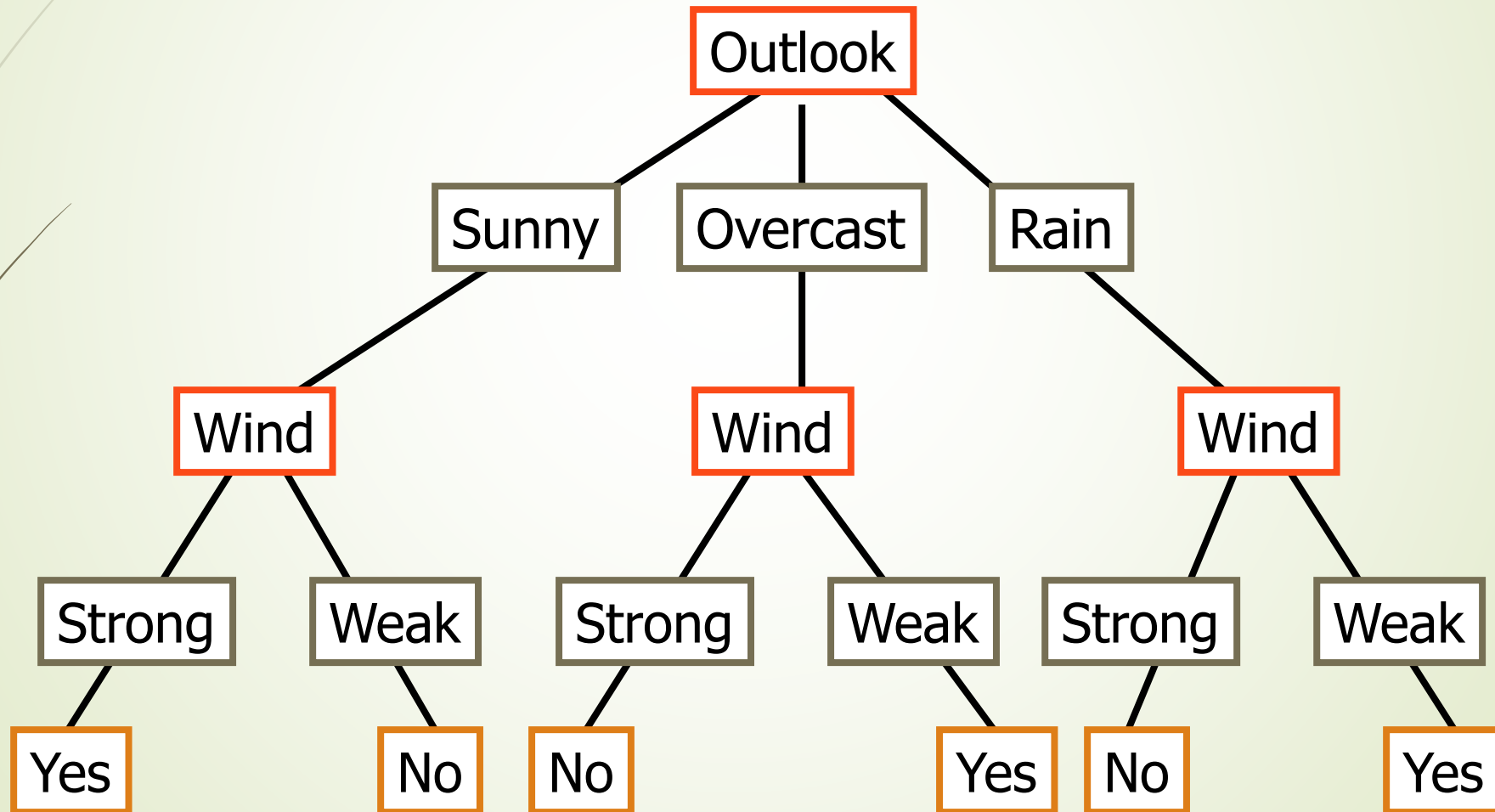
Decision Tree for Disjunction

Outlook=Sunny \vee Wind=Weak



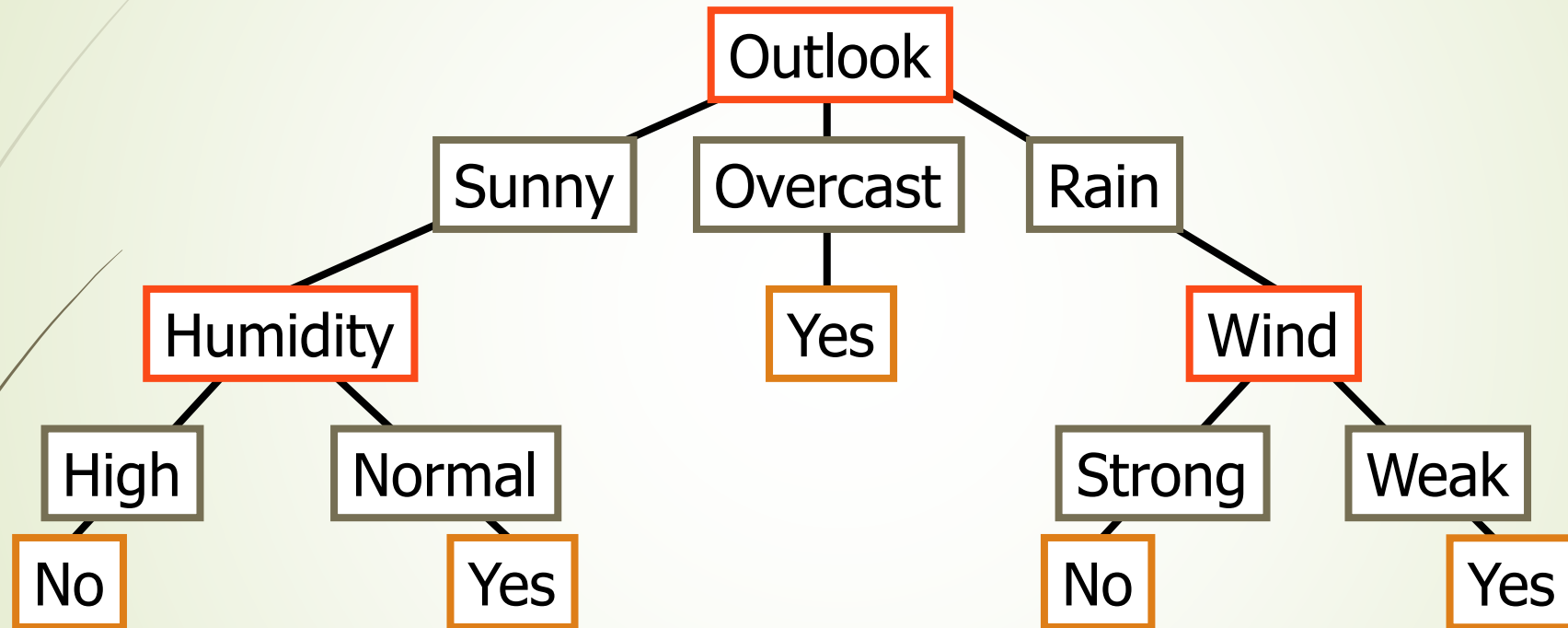
Decision Tree for XOR

Outlook=Sunny XOR Wind=Weak



Decision Tree

- decision trees represent disjunctions of conjunctions



(Outlook=Sunny \wedge Humidity=Normal)

✓ (Outlook=Overcast)

✓ (Outlook=Rain \wedge Wind=Weak)

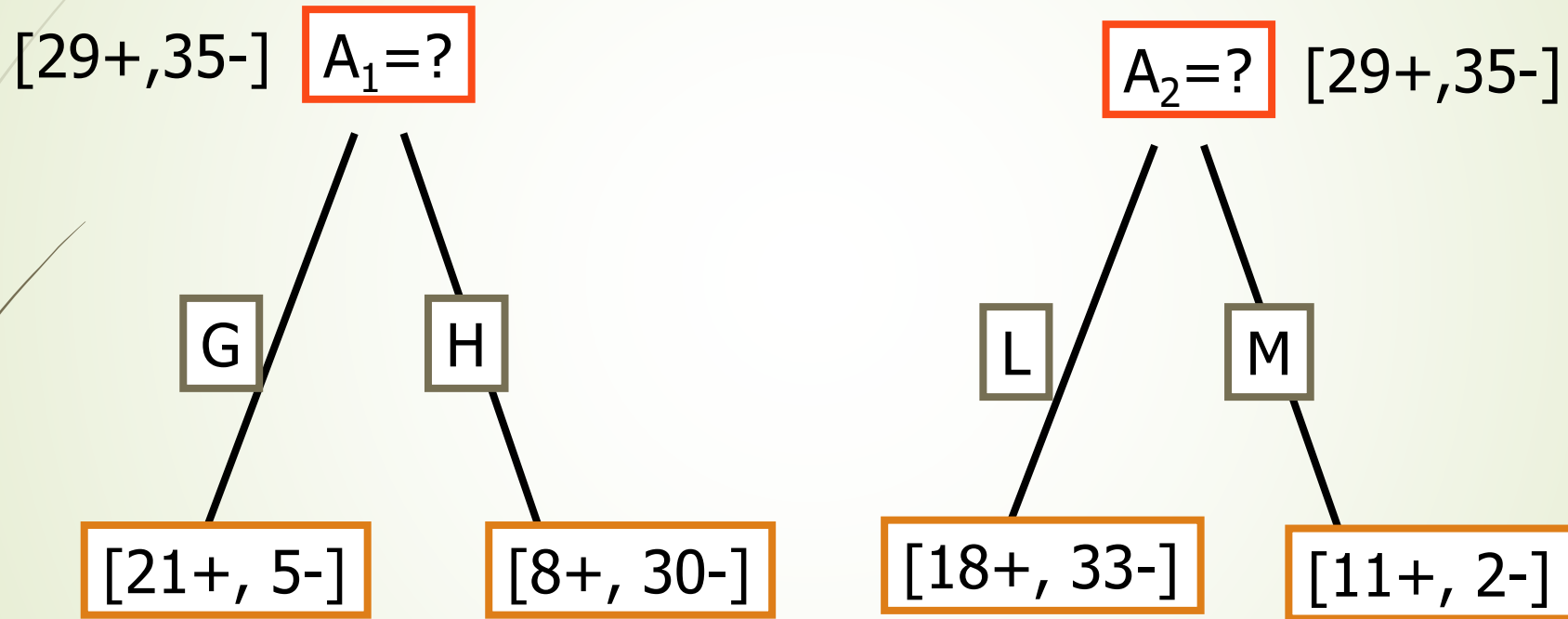
When to consider Decision Trees

- ▶ Instances describable by attribute-value pairs
- ▶ Target function is discrete valued
- ▶ Disjunctive hypothesis may be required
- ▶ Possibly noisy training data
- ▶ Missing attribute values
- ▶ Examples:
 - ▶ Medical diagnosis
 - ▶ Credit risk analysis
 - ▶ Object classification for robot manipulator (Tan 1993)

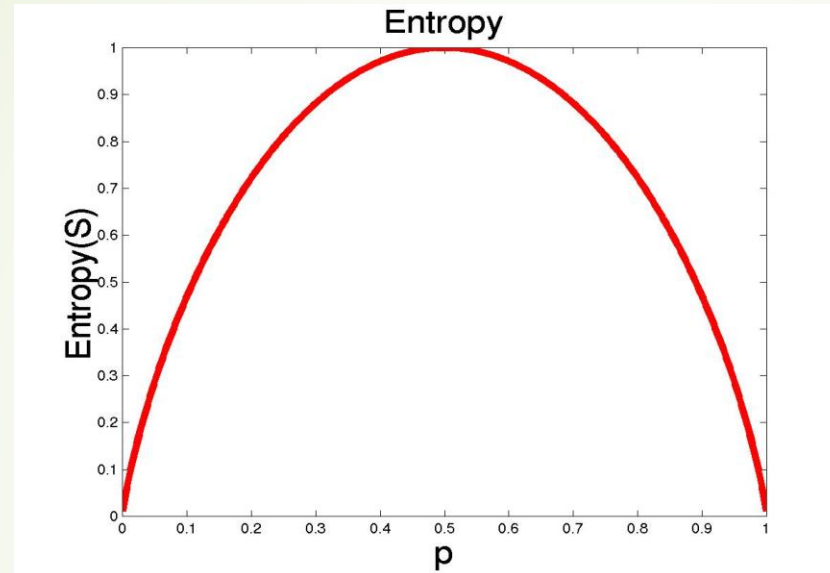
Top-Down Induction of Decision Trees ID3

1. $A \leftarrow$ the “best” decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A create new descendant
4. Sort training examples to leaf node according to the attribute value of the branch
5. If all training examples are perfectly classified (same value of target attribute) stop, else iterate over new leaf nodes.

Which attribute is best?



Entropy



- S is a sample of training examples
- p_+ is the proportion of positive examples
- p_- is the proportion of negative examples
- Entropy measures the impurity of S

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Entropy

- ▶ Entropy(S) = expected number of bits needed to encode class (+ or -) of randomly drawn members of S (under the optimal, shortest length-code)

Why?

- ▶ Information theory optimal length code assign $-\log_2 p$ bits to messages having probability p .
- ▶ So the expected number of bits to encode (+ or -) of random member of S :

$$-p_+ \log_2 p_+ - p_- \log_2 p_-$$

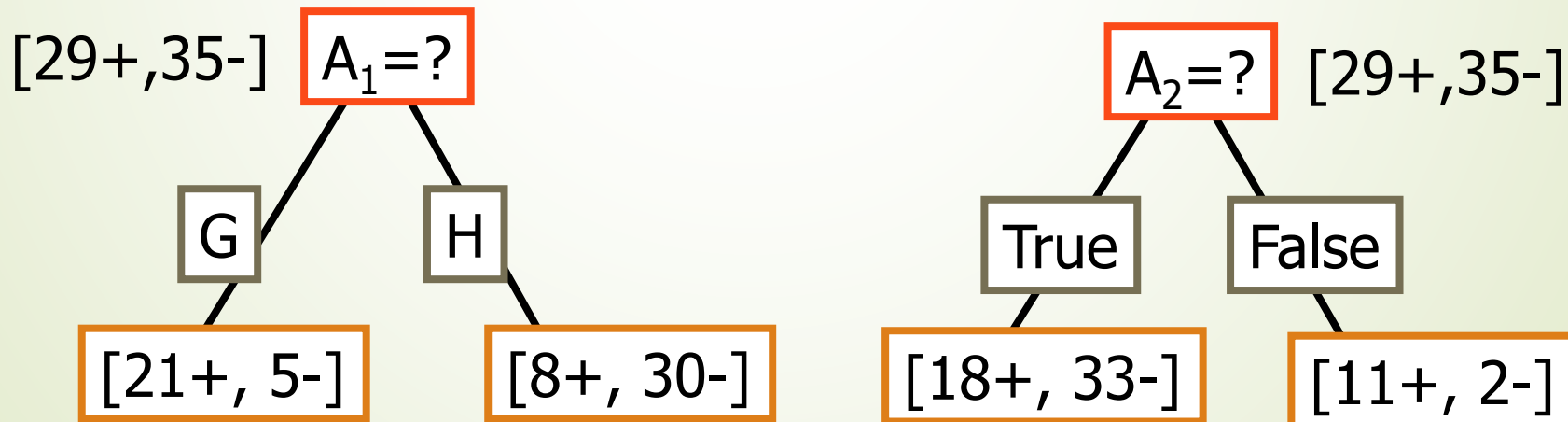
Information Gain (S=E)

18

- Gain(S,A): expected reduction in entropy due to sorting S on attribute A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in D_A} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Entropy([29+, 35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64 \\ = 0.99$$



Information Gain

$$\text{Entropy}([21+, 5-]) = 0.71$$

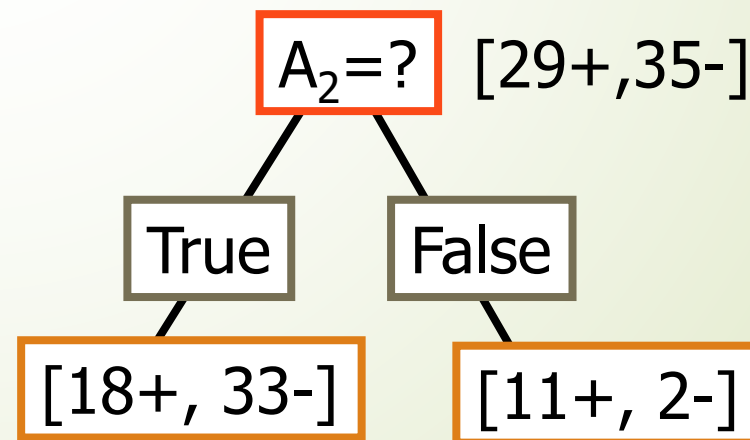
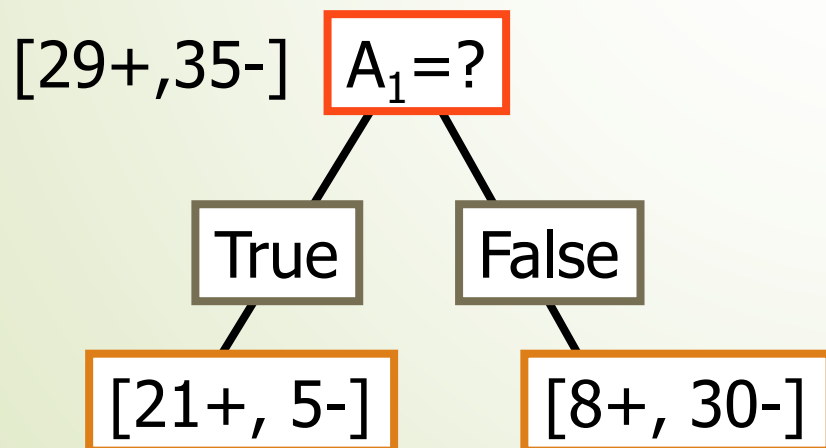
$$\text{Entropy}([8+, 30-]) = 0.74$$

$$\begin{aligned} \text{Gain}(S, A_1) &= \text{Entropy}(S) \\ &\quad - 26/64 * \text{Entropy}([21+, 5-]) \\ &\quad - 38/64 * \text{Entropy}([8+, 30-]) \\ &= 0.27 \end{aligned}$$

$$\text{Entropy}([18+, 33-]) = 0.94$$

$$\text{Entropy}([11+, 2-]) = 0.62$$

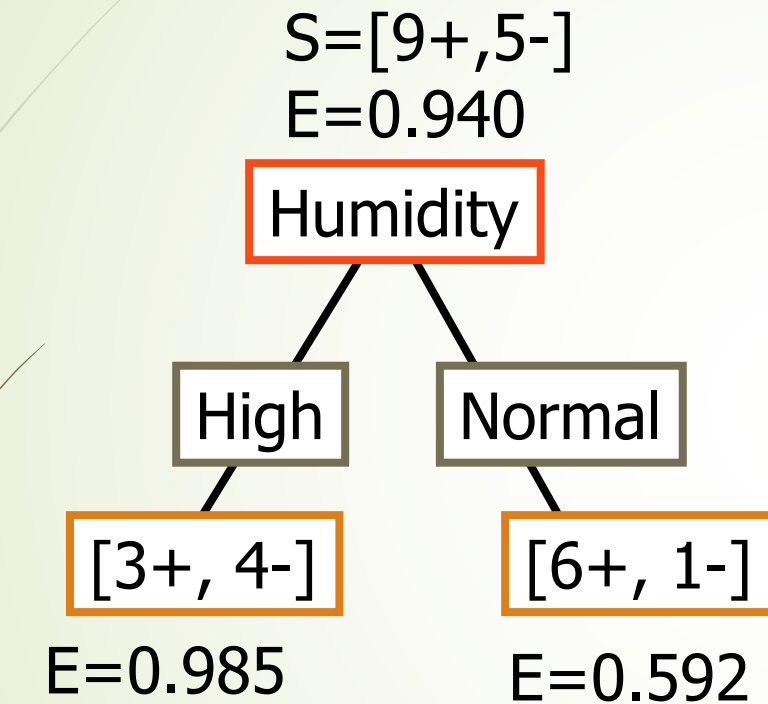
$$\begin{aligned} \text{Gain}(S, A_2) &= \text{Entropy}(S) \\ &\quad - 51/64 * \text{Entropy}([18+, 33-]) \\ &\quad - 13/64 * \text{Entropy}([11+, 2-]) \\ &= 0.12 \end{aligned}$$



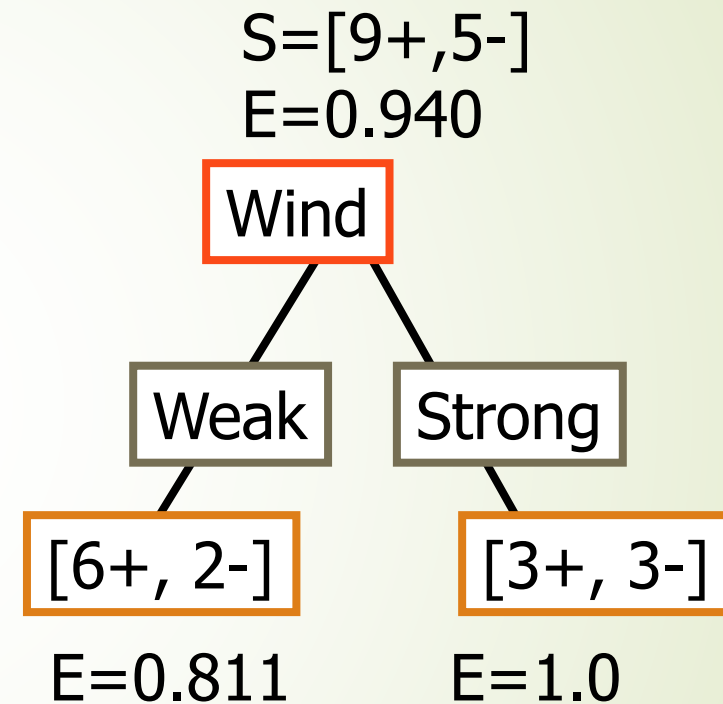
Training Examples

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
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	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	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

Selecting the Next Attribute



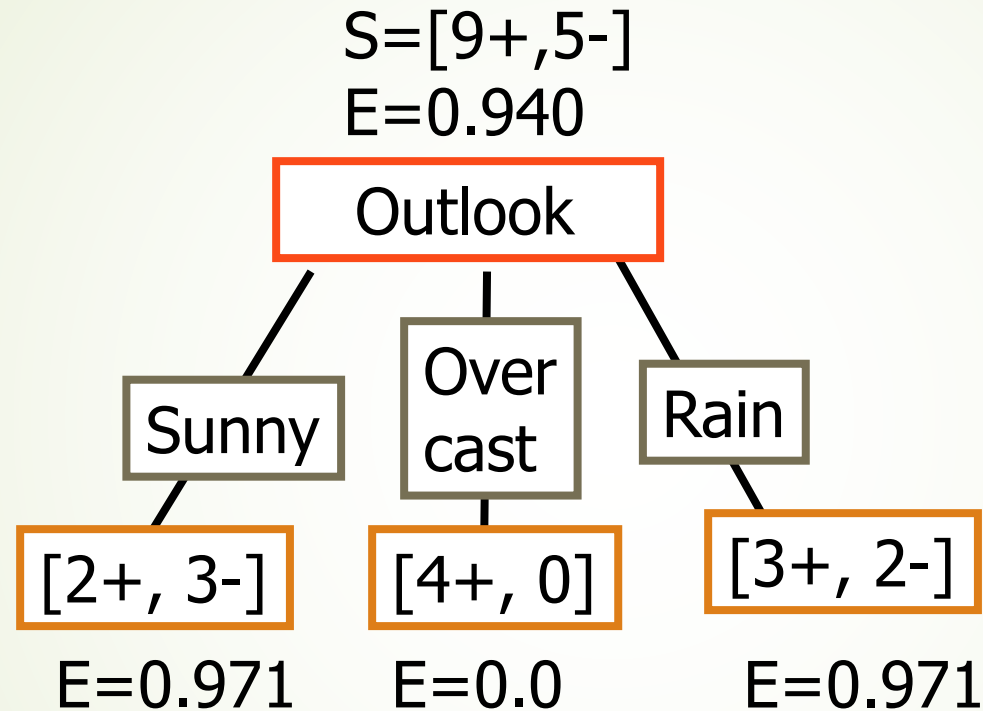
$$\begin{aligned}
 \text{Gain}(S, \text{Humidity}) &= 0.940 - (7/14) * 0.985 \\
 &\quad - (7/14) * 0.592 \\
 &= 0.151
 \end{aligned}$$



$$\begin{aligned}
 \text{Gain}(S, \text{Wind}) &= 0.940 - (8/14) * 0.811 \\
 &\quad - (6/14) * 1.0 \\
 &= 0.048
 \end{aligned}$$

Humidity provides greater info. gain than Wind, w.r.t target classification.

Selecting the Next Attribute



$$\begin{aligned} \text{Gain}(S, \text{Outlook}) &= 0.940 - (5/14) * 0.971 \\ &\quad - (4/14) * 0.0 - (5/14) * 0.0971 \\ &= 0.247 \end{aligned}$$

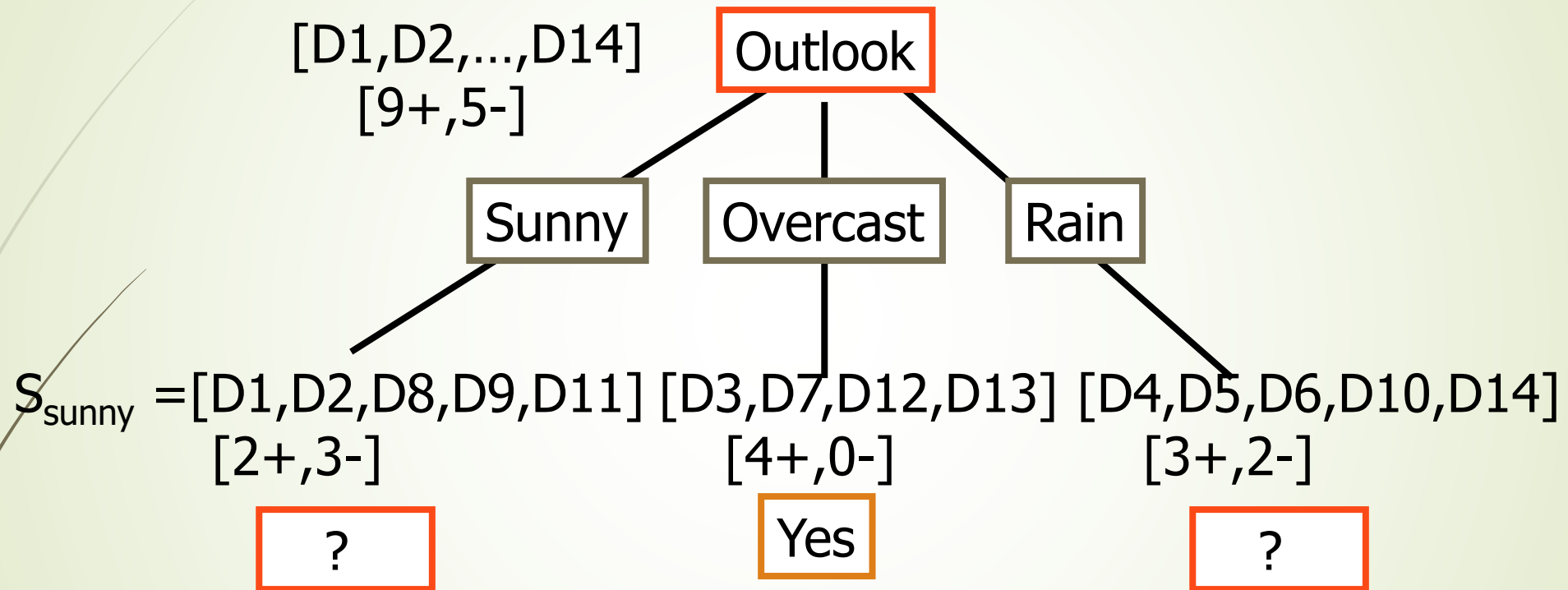
Selecting the Next Attribute

The information gain values for the 4 attributes are:

- $\text{Gain}(S, \text{Outlook}) = 0.247$
- $\text{Gain}(S, \text{Humidity}) = 0.151$
- $\text{Gain}(S, \text{Wind}) = 0.048$
- $\text{Gain}(S, \text{Temperature}) = 0.029$

where S denotes the collection of training examples

ID3 Algorithm

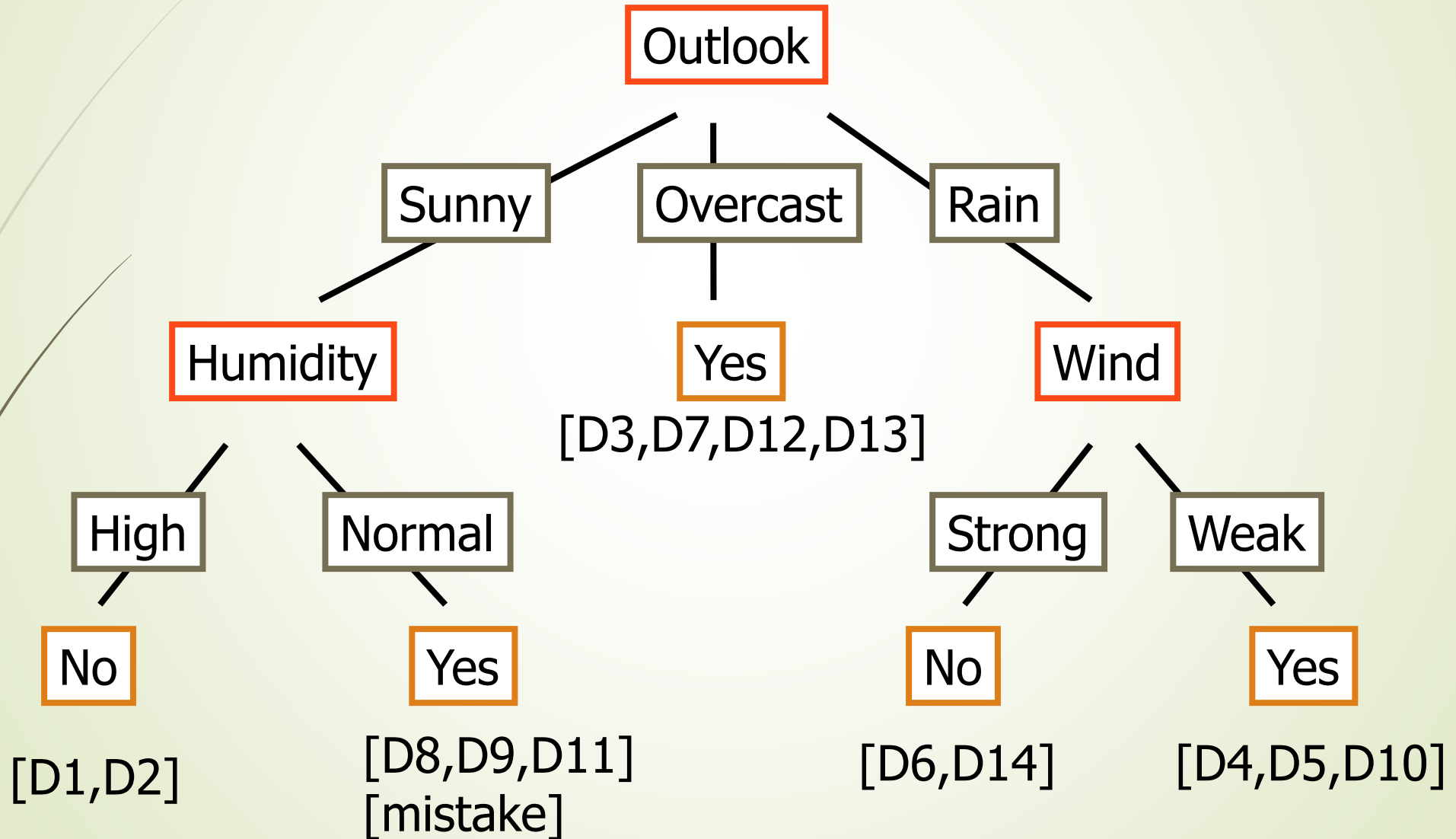


$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = 0.970 - (3/5)0.0 - 2/5(0.0) = 0.970$$

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

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = 0.970 - (2/5)1.0 - 3/5(0.918) = 0.019$$

ID3 Algorithm



Occam's Razor

26

"If two theories explain the facts equally well, then the simpler theory is to be preferred"

Arguments in favor:

- Fewer short hypotheses than long hypotheses
- A short hypothesis that fits the data is unlikely to be a coincidence
- A long hypothesis that fits the data might be a coincidence

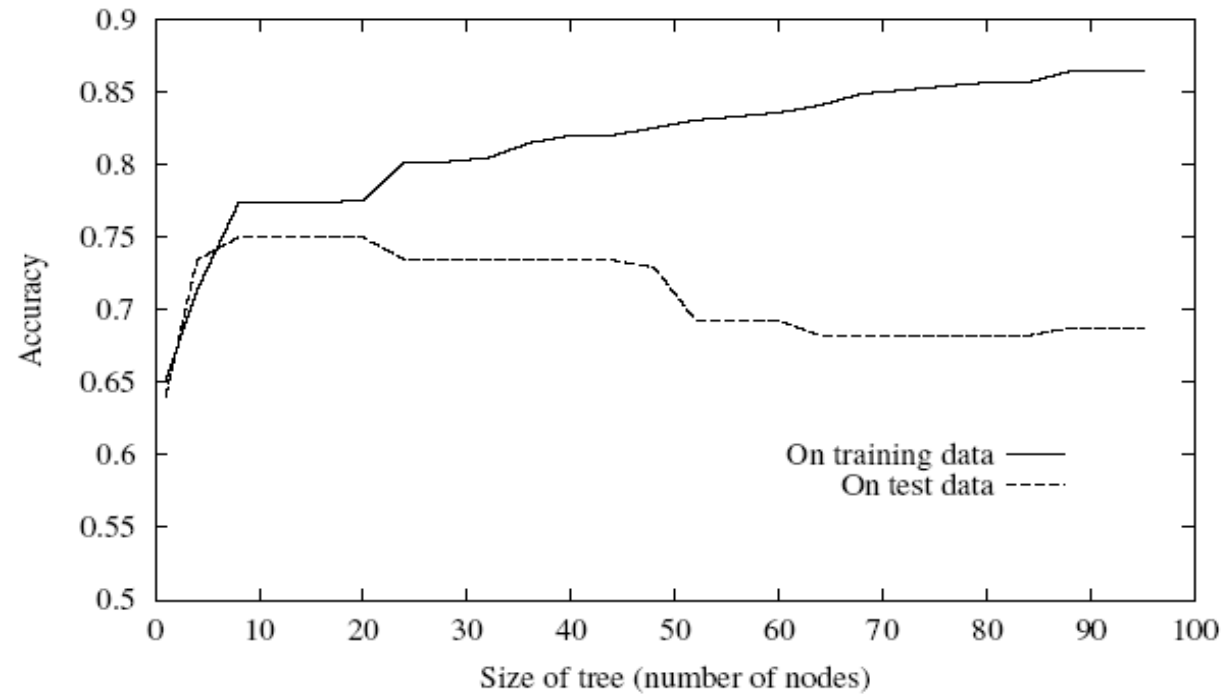
Arguments opposed:

- There are many ways to define small sets of hypotheses

Overfitting

27

- One of the biggest problems with decision trees is **Overfitting**



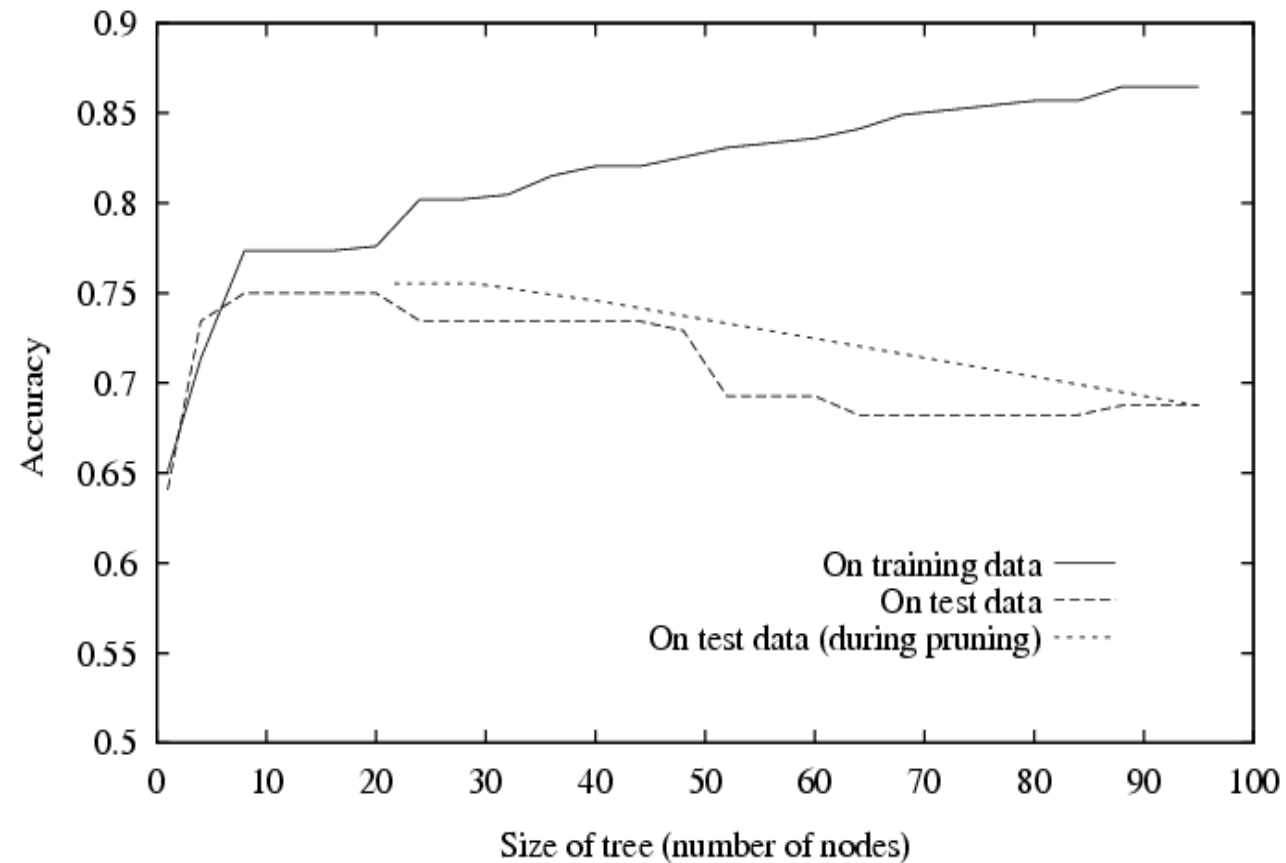
Avoid Overfitting

- Stop growing when split not statistically significant
- Grow full tree, then post-prune

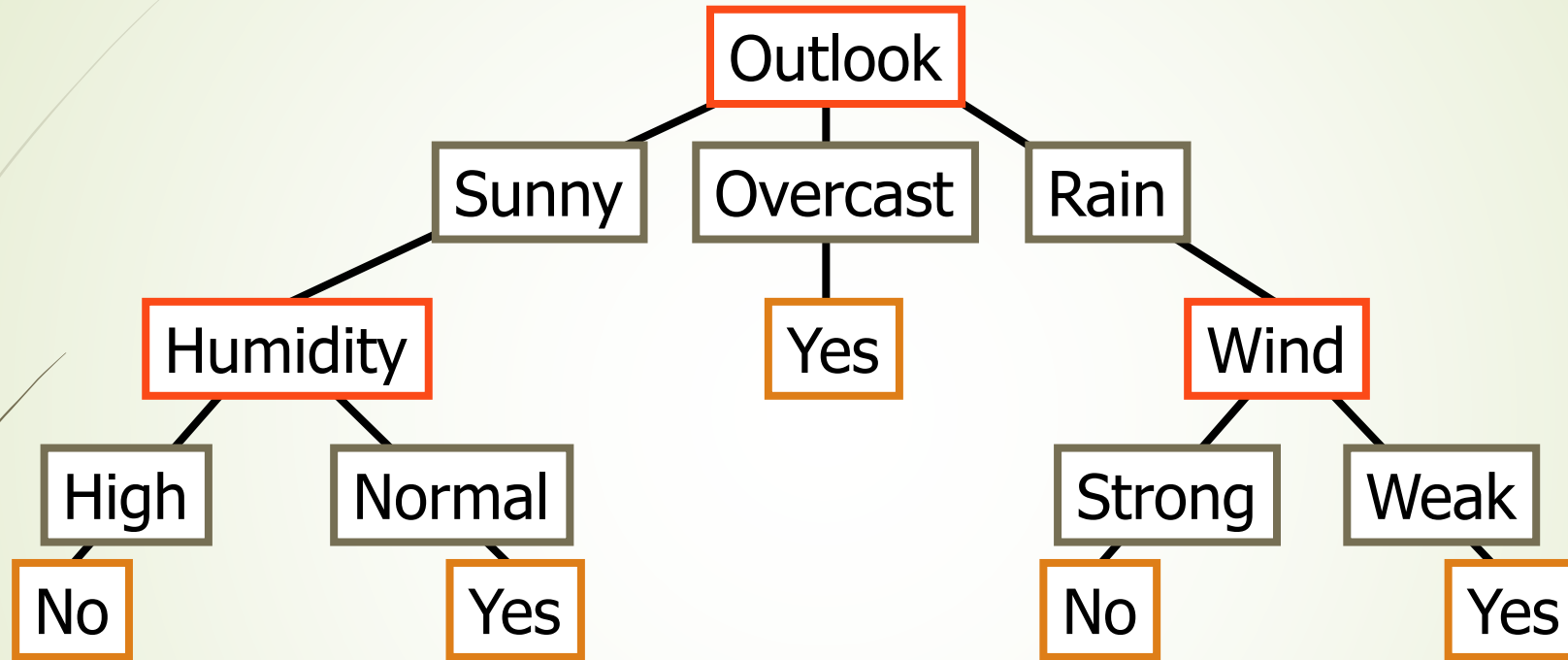
Select “best” tree:

- measure performance over training data
- measure performance over separate validation data set
- $\min(|\text{tree}| + |\text{misclassifications}(\text{tree})|)$

Effect of Reduced Error Pruning



Converting a Tree to Rules



R_1 : If (Outlook=Sunny) \wedge (Humidity=High) Then PlayTennis=No

R_2 : If (Outlook=Sunny) \wedge (Humidity=Normal) Then PlayTennis=Yes

R_3 : If (Outlook=Overcast) Then PlayTennis=Yes

R_4 : If (Outlook=Rain) \wedge (Wind=Strong) Then PlayTennis=No

R_5 : If (Outlook=Rain) \wedge (Wind=Weak) Then PlayTennis=Yes

Continuous Valued Attributes

Create a discrete attribute to test continuous

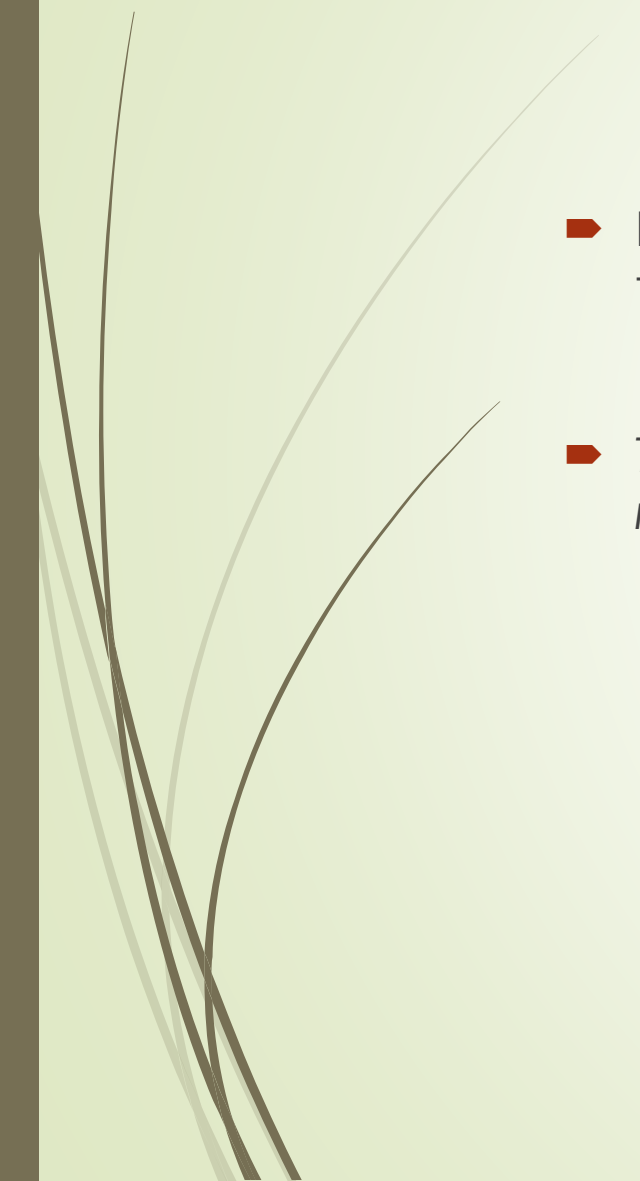
- Temperature = 24.5°C
- (Temperature > 20.0°C) = {true, false}

Where to set the threshold?

Temperature	15°C	18°C	19°C	22°C	24°C	27°C
PlayTennis	No	No	Yes	Yes	Yes	No



Random forest classifier

- Random forest classifier, an extension to bagging which uses *de-correlated* trees.
 - *To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction*
- 

Random Forest Classifier

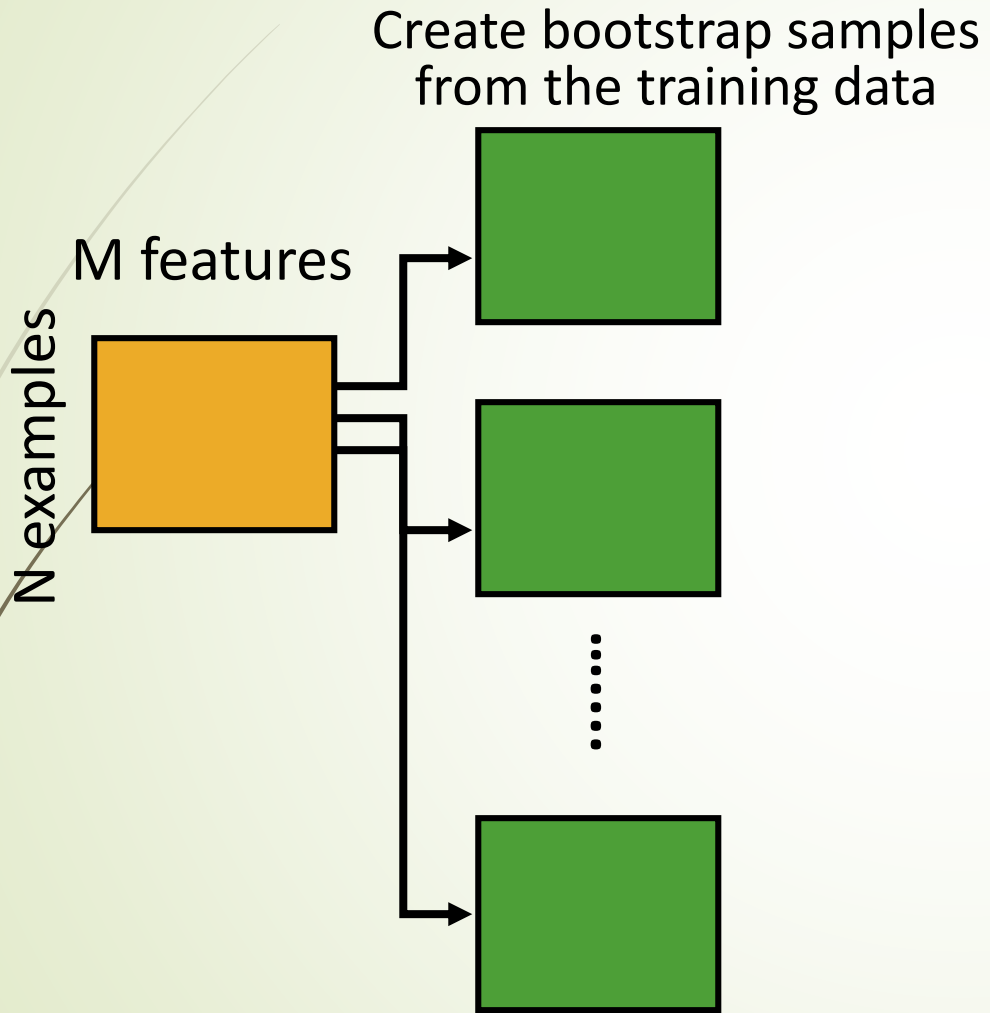
Training Data

M features

N examples

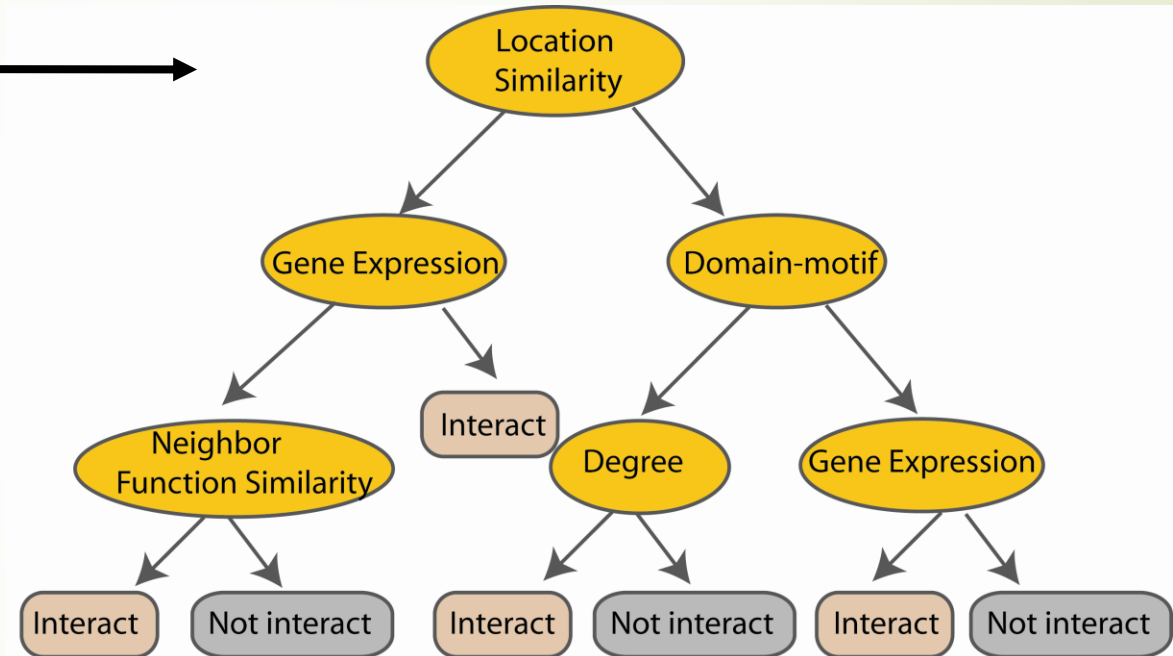
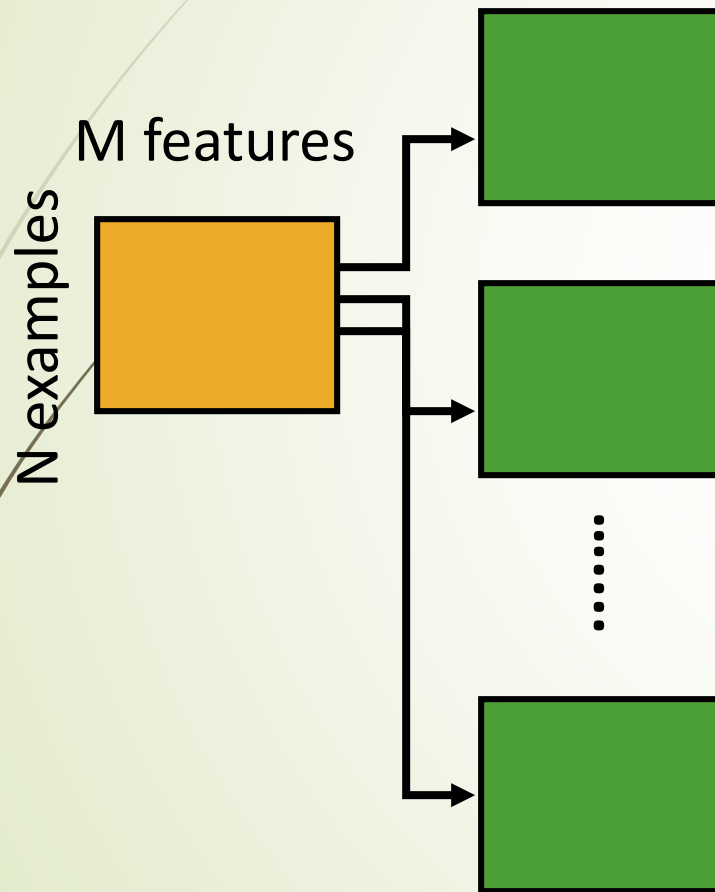


Random Forest Classifier



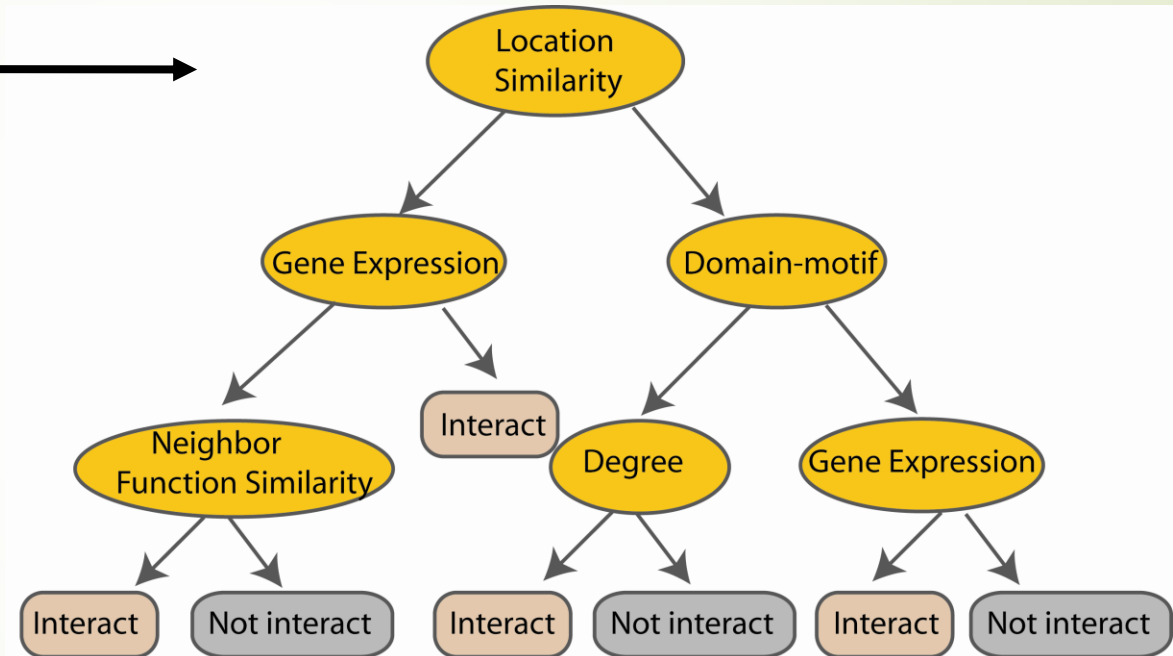
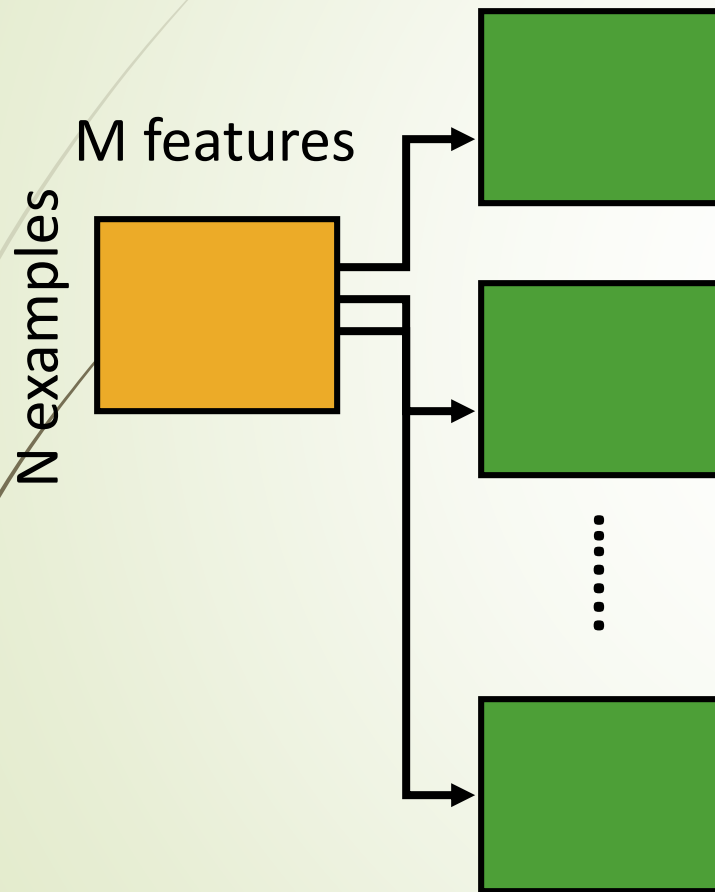
Random Forest Classifier

Construct a decision tree



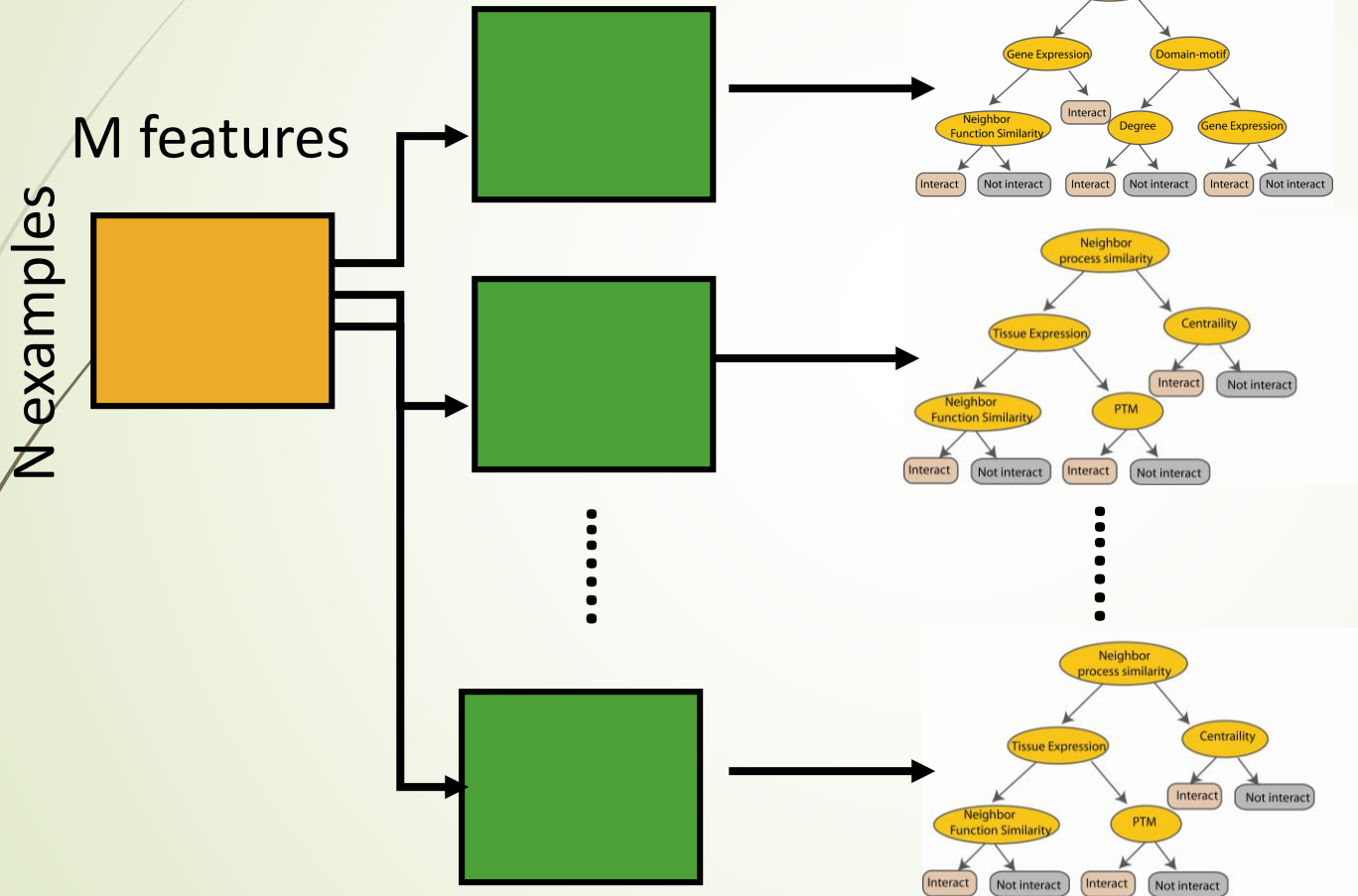
Random Forest Classifier

At each node in choosing the split feature
choose only among $m < M$ features

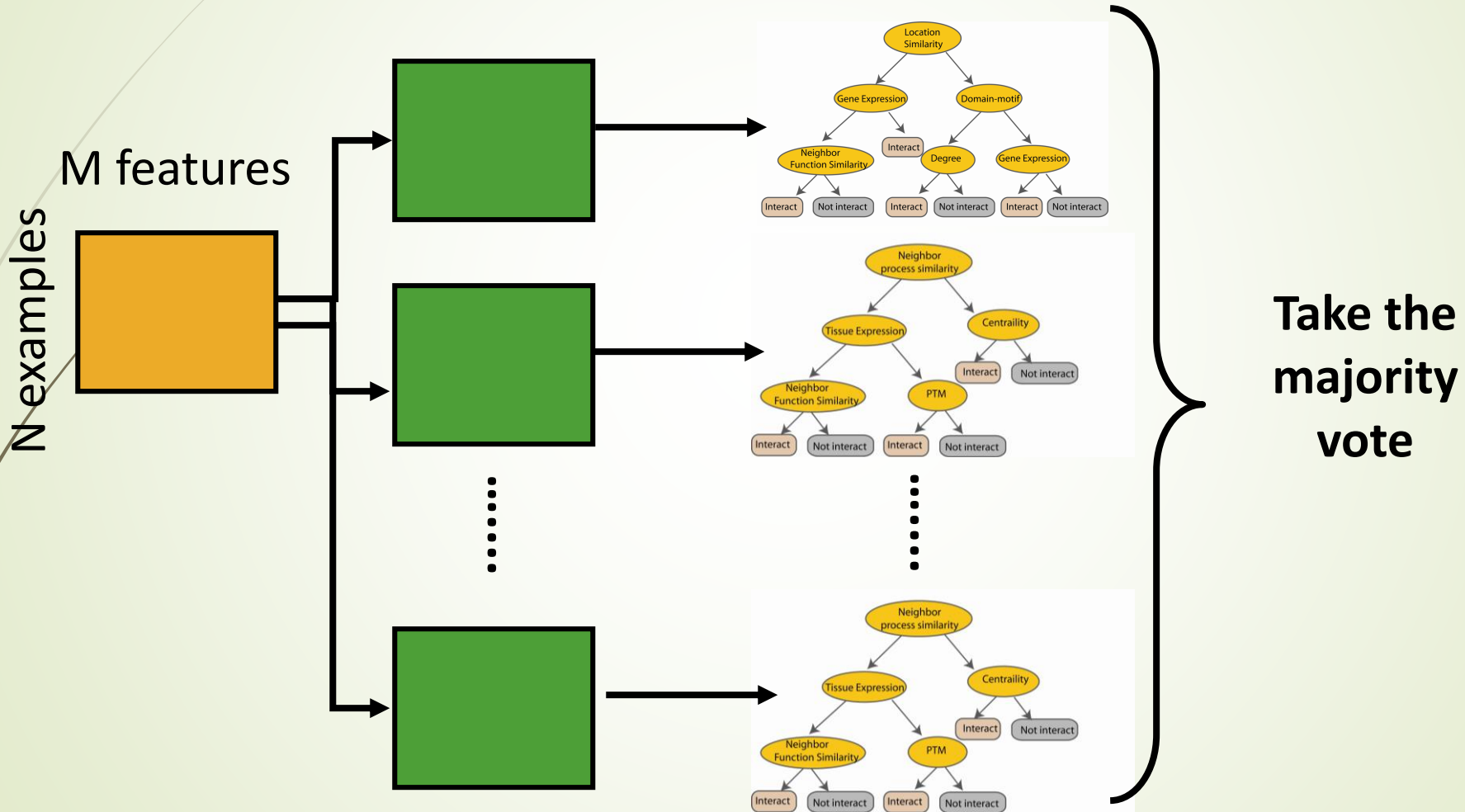


Random Forest Classifier

Create decision tree
from each bootstrap sample



Random Forest Classifier



Unknown Attribute Values

What if some examples have missing values of A ?

Use training example anyway sort through tree

- ▶ If node n tests A , assign most common value of A among other examples sorted to node n .
- ▶ Assign most common value of A among other examples with same target value
- ▶ Assign probability p_i to each possible value v_i of A
 - ▶ Assign fraction p_i of example to each descendant in tree

Classify new examples in the same fashion

Cross-Validation

- Estimate the accuracy of an hypothesis induced by a supervised learning algorithm
- Predict the accuracy of an hypothesis over future unseen instances
- Select the optimal hypothesis from a given set of alternative hypotheses
 - Pruning decision trees
 - Model selection
 - Feature selection
- Combining multiple classifiers (boosting)