Pattern Classification and Recognition:

Ensemble Learning

ECE 681

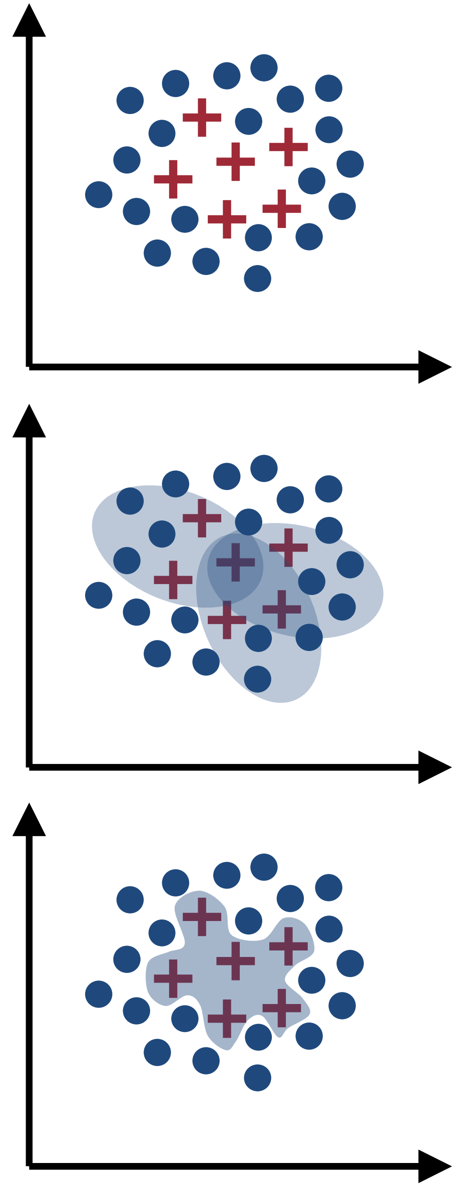
Spring 2016

Stacy Tantum, Ph.D.

Committees

"A camel is a horse designed by a committee."

Decisions by Committee

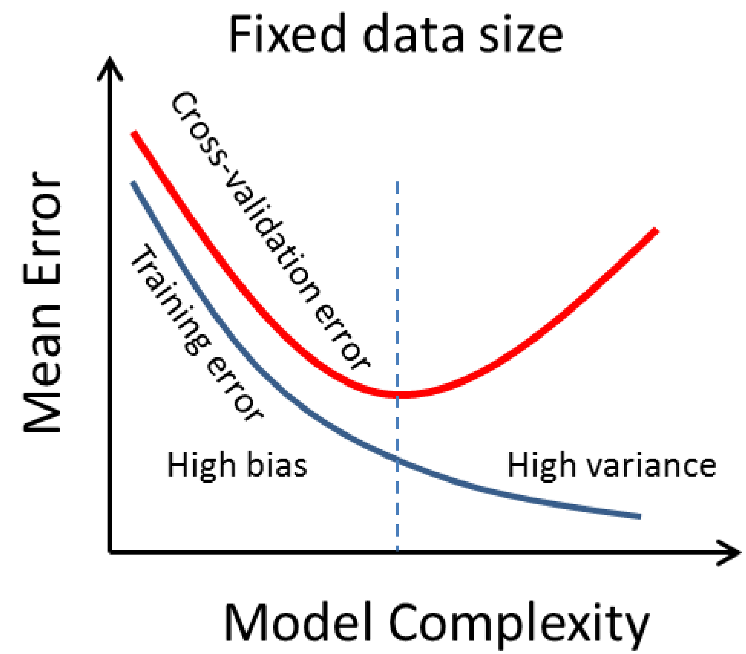


Which classifiers?
How to make complementary?
How to combine?

Generalization Error

A lingering question…

- How to ensure performance on the training set fairly (accurately?) predicts performance on a test set?
- Average/ expected difference between training set and testing set performance



$\rightarrow$ Ensemble learning, or committee machines

Classifier (In)Stability

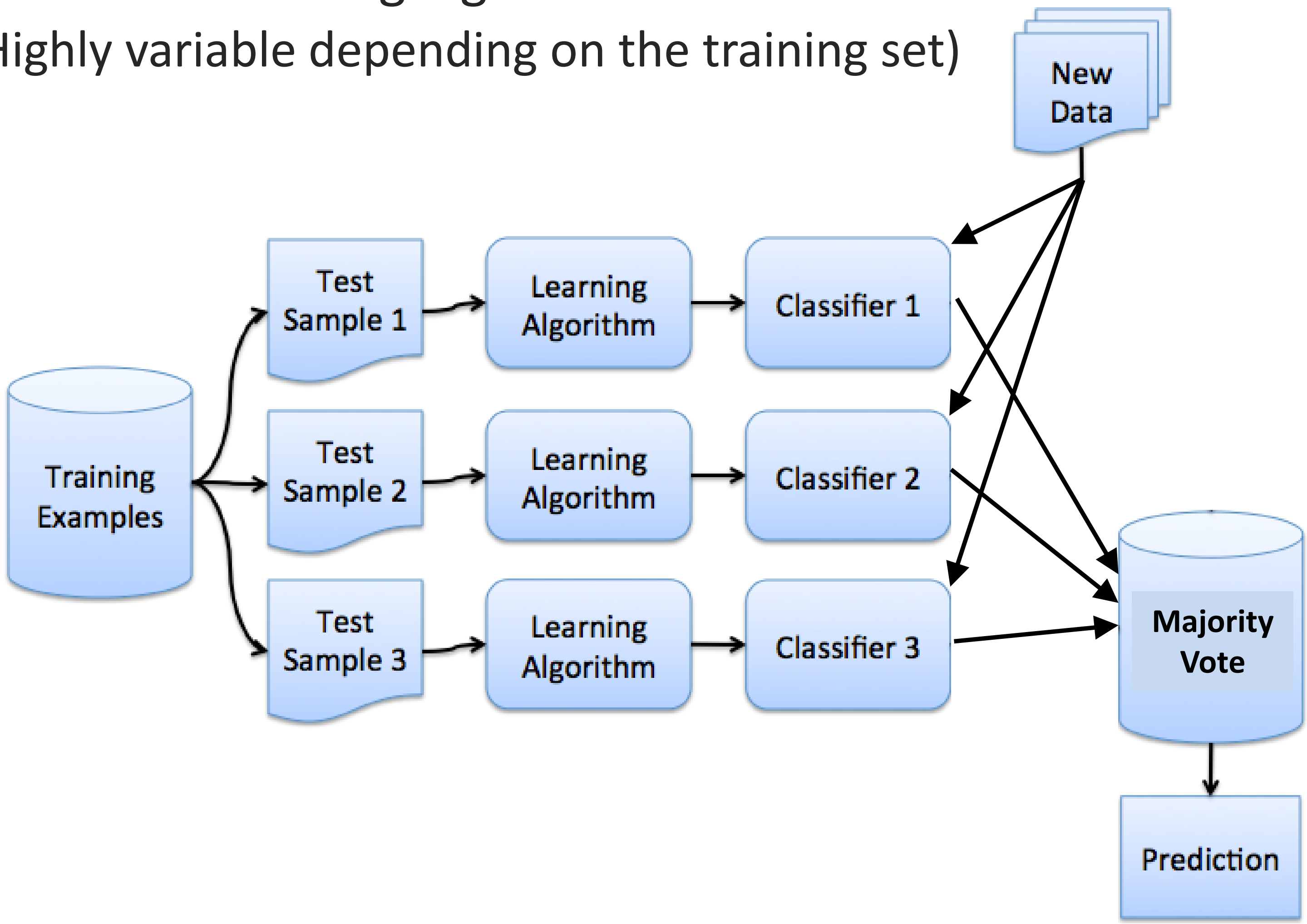
UNSTABLE CLASSIFIERS	STABLE CLASSIFIERS
Small changes in data set $\rightarrow$ major changes in classifier	Small changes in data set $\rightarrow$ small changes in data set
High variance (from overfitting)	Low variance
Low bias	(Possibly) high bias (from underfitting)
Neural networks, Decision trees	Linear discriminant analysis

Exploit classifier instability to create a diverse set of classifiers using subsets of the data

Bagging (Bootstrap Aggregating)

Useful if the learning algorithm is unstable

- (Highly variable depending on the training set)



Boosting

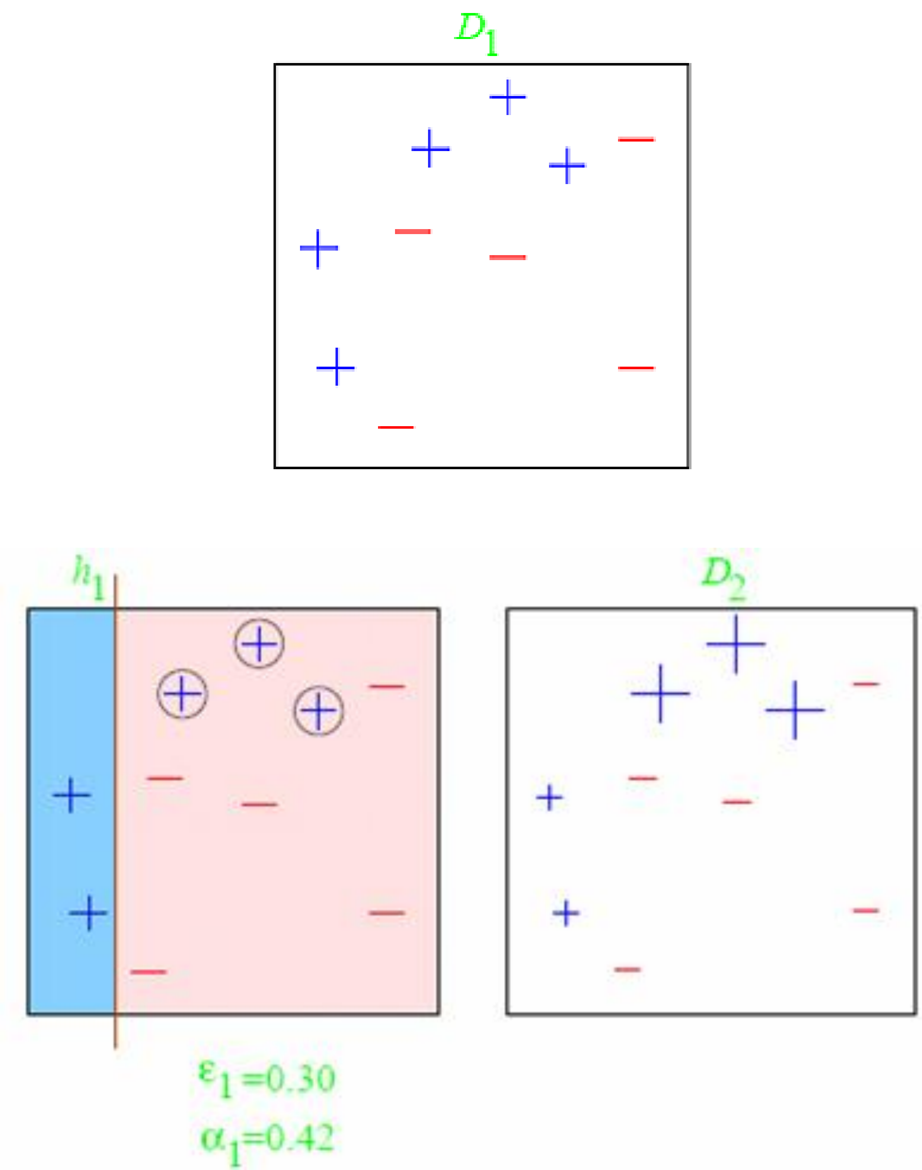
Useful if the learning algorithm is complex and unstable

- Highly variable depending on the training set

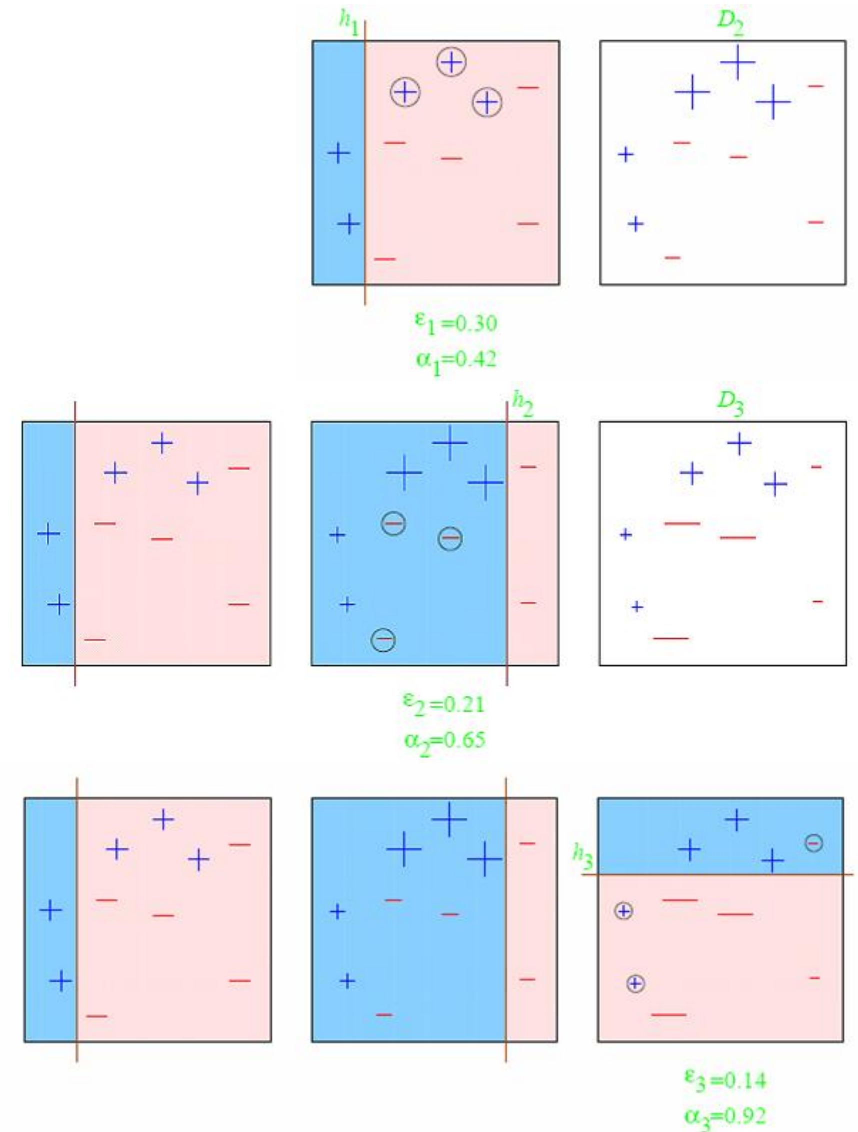
Combine weak classifiers to create a strong classifier

AdaBoost (adaptive boosting)

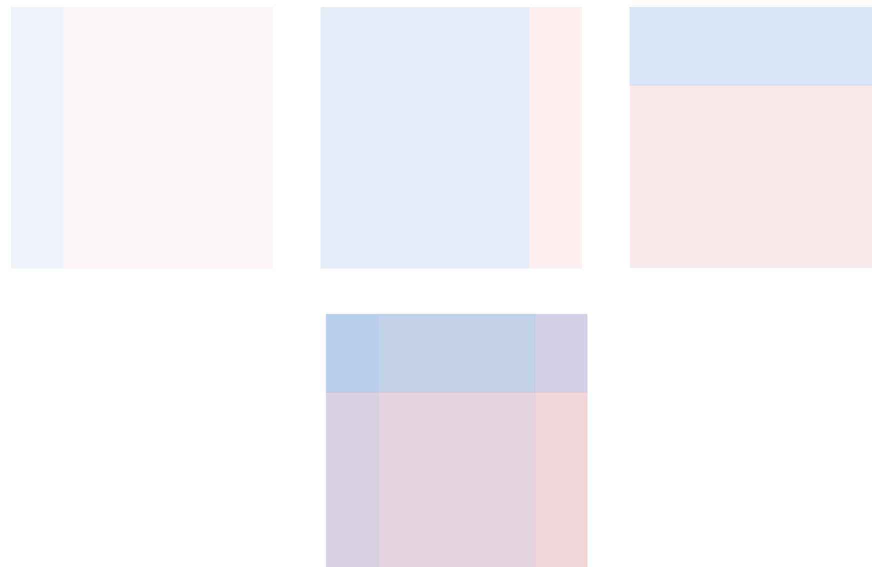
- Upweight training previously incorrectly classified training samples



Boosting



h_1
D_2
$\varepsilon_1=0.30$
$\alpha_1=0.42$
h_2
D_3
$\varepsilon_2=0.21$
$\alpha_2=0.65$
h_3
$\varepsilon_3=0.14$
$\alpha_3=0.92$



Bagging vs. Boosting

BAGGING

Different data samples for each classifier

Majority vote

BOOSTING

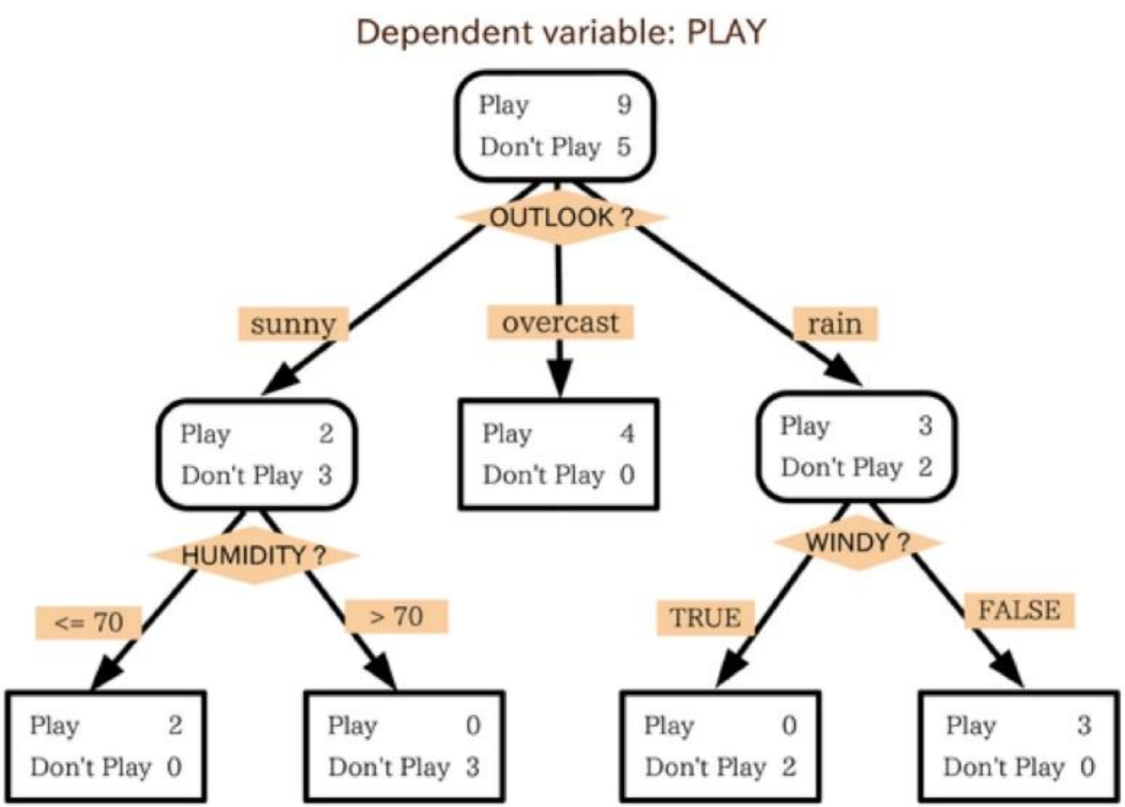
Different weighting of data for each classifier

Weighted vote

Classification Trees

Sequence of (typically binary) decisions

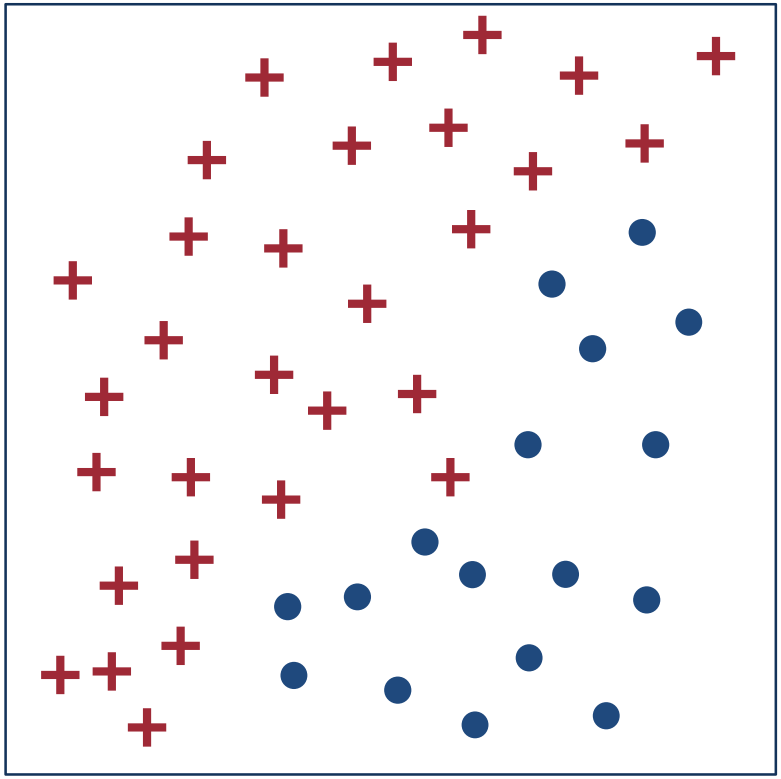
Continue until all data in the *terminal node* are a single class



Classification Trees

Sequence of (typically binary) decisions

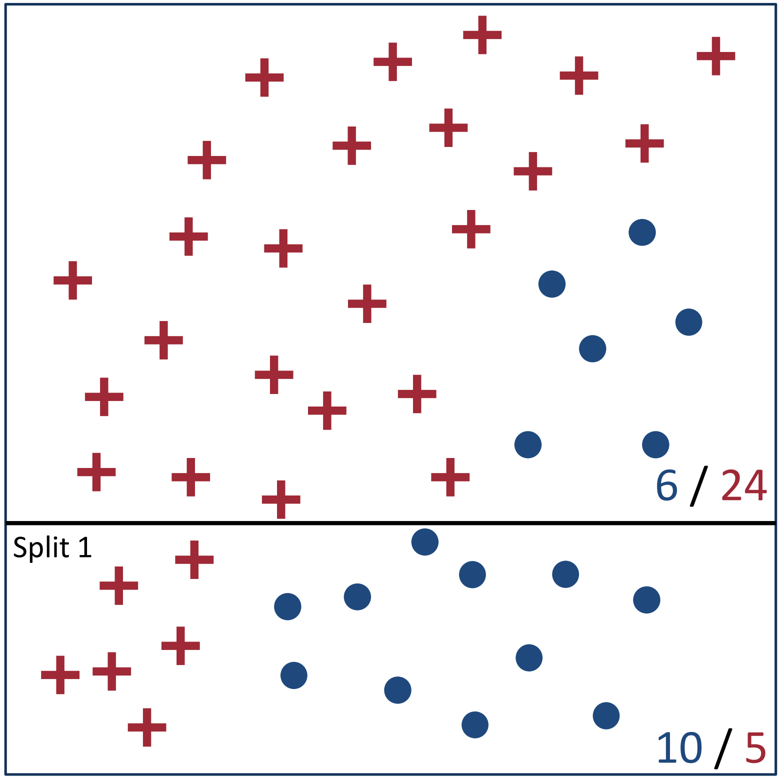
Continue until all data points in the *terminal node* are a single class



Classification Trees

Sequence of (typically binary) decisions

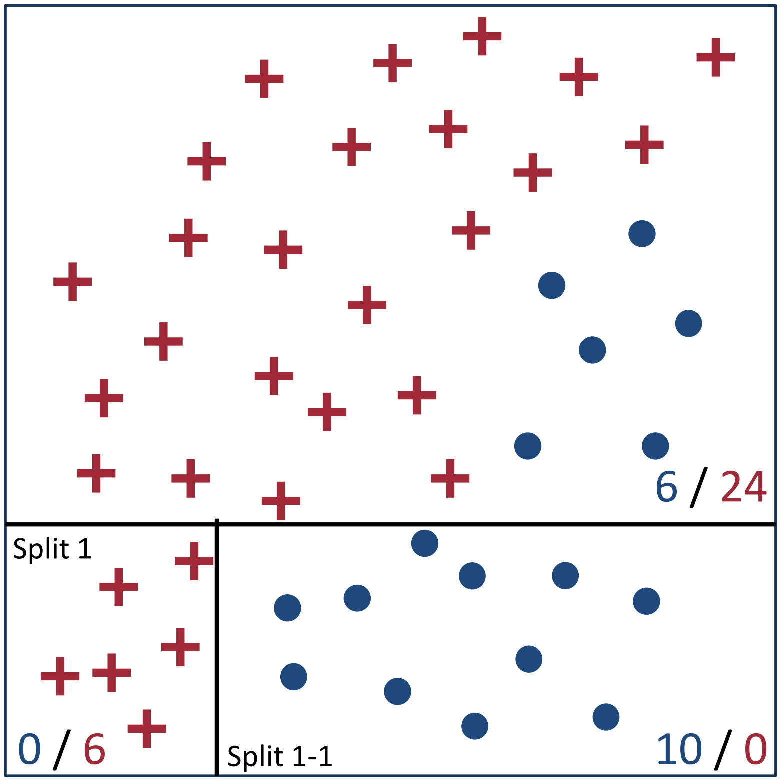
Continue until all data points in the *terminal node* are a single class



Classification Trees

Sequence of (typically binary) decisions

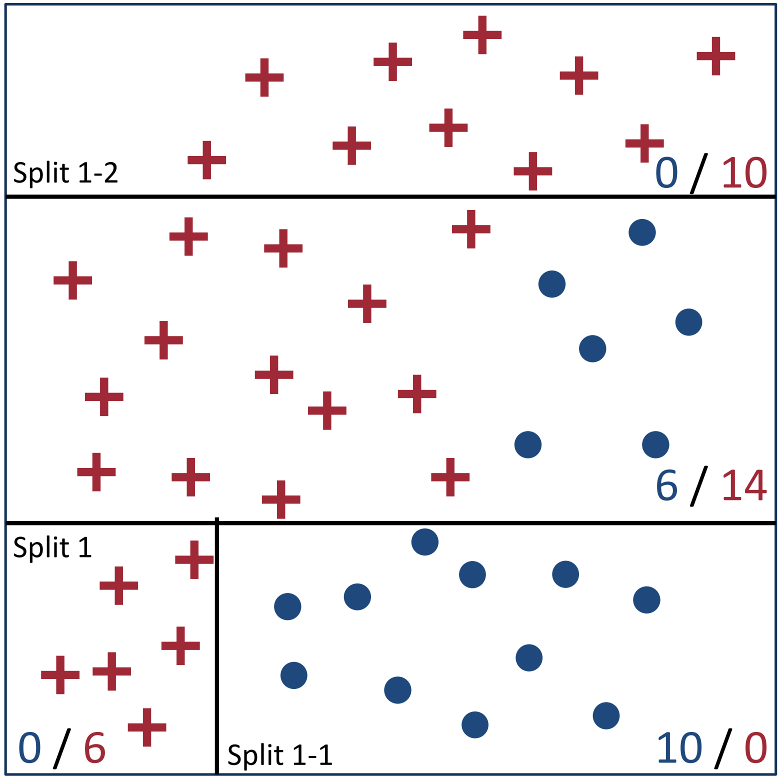
Continue until all data points in the *terminal node* are a single class



Classification Trees

Sequence of (typically binary) decisions

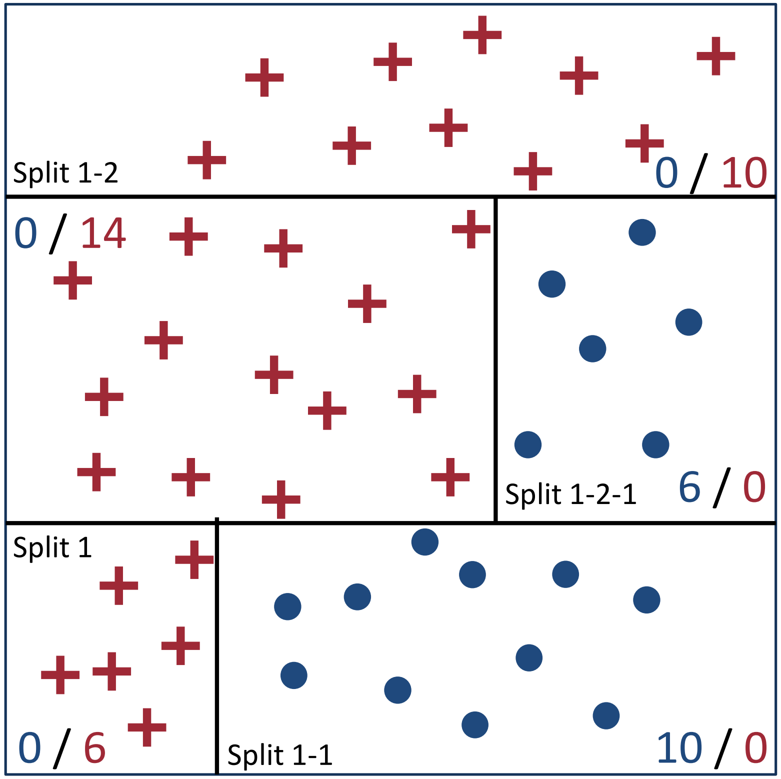
Continue until all data points in the *terminal node* are a single class



Classification Trees

Sequence of (typically binary) decisions

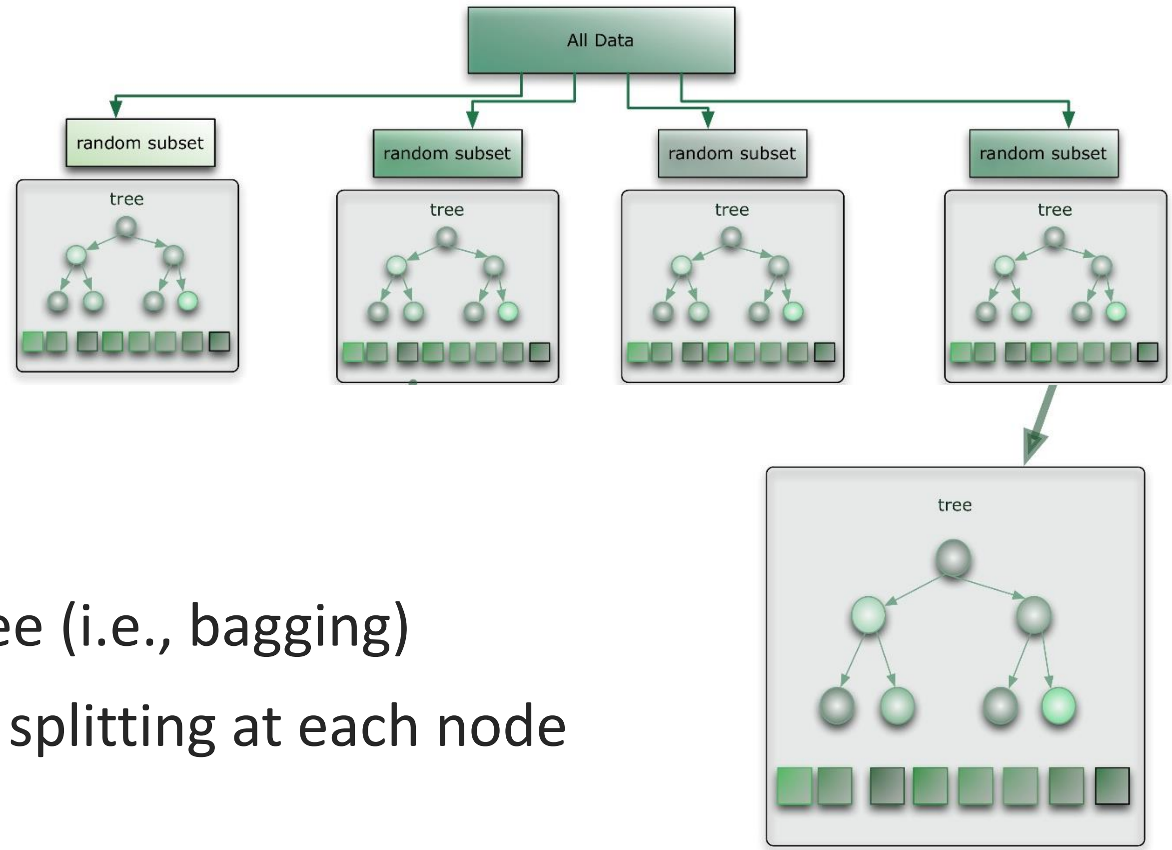
Continue until all data points in the *terminal node* are a single class



Random Forests

Collection of random trees

At each node:
choose some small subset of variables at random
find a variable (and a value for that variable) which optimizes the split



Randomize:

- Data used for each tree (i.e., bagging)
- Features available for splitting at each node