

Bagging, Random Forest , & Boosting

Dr Mohamed Bennasar

Outline

Ensemble Learning

Bagging

Random Forests

Boosting

Overfitting Issue with Decision Trees

- DTs can fit exactly against its training data.
- Generalisation of overfitted model is questionable.
- Overfitting to the noise in the training data can cause high variance
- Two approaches:
 - 1. Stop growing the tree when further splitting does not improve the performance.
 - 2. Grow a full tree, then prune the tree, by eliminating nodes.

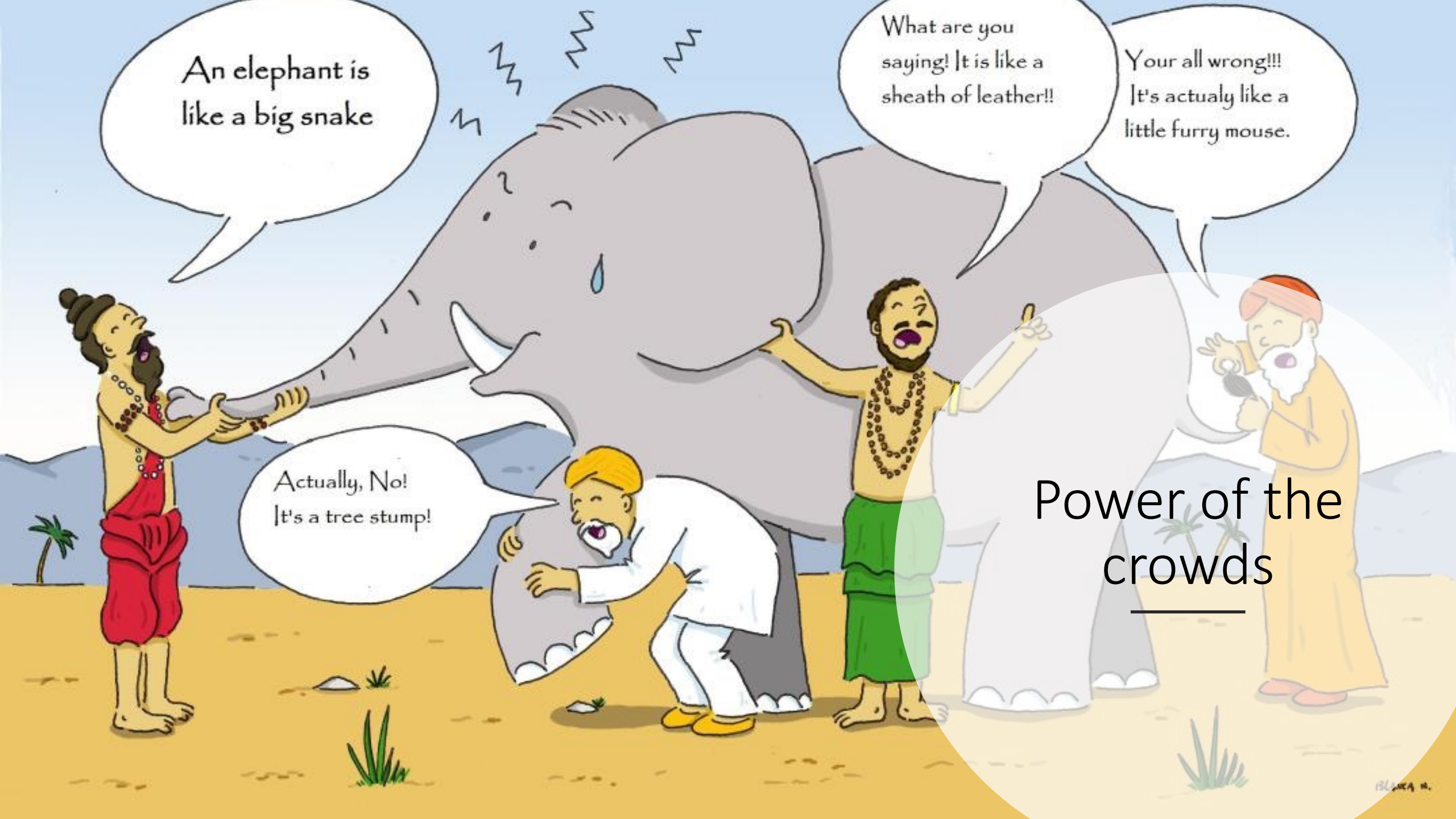
What if we learn multiple trees?

- **Ensemble Learning:**

- An ensemble is a machine learning model that combines the predictions from two or more models.
- The models that contribute to the ensemble, referred to as ensemble members, may be the same type or different types and may or may not be trained on the same training data.
- The predictions made by the ensemble members are combined.

Why Use Ensemble Method

- When a single decision tree does not perform well.
- An ensemble can make better predictions and achieve better performance than any single contributing model.
- An ensemble reduces the spread or dispersion of the predictions and model performance.



An elephant is
like a big snake

What are you
saying! It is like a
sheath of leather!!

Your all wrong!!!
It's actually like a
little furry mouse.

Actually, No!
It's a tree stump!

Power of the
crowds

Building a Random Tree

Game	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Overcast	Mild	High	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Game	Outlook	Temperature	Humidity	Windy	Play
11	Sunny	Mild	Normal	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
7	Overcast	Cool	Normal	TRUE	Yes
6	Rainy	Cool	Normal	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
1	Sunny	Hot	High	FALSE	No
6	Rainy	Cool	Normal	TRUE	No

Bagging

Bagging, also known as **Bootstrap** aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy dataset.

In bagging, several training data samples are generated by selecting a random sample of data in a training set with replacement—meaning that the individual data points can be chosen more than once.

After several training data samples are generated, Decision trees are then trained independently.

Majority voting is used to combine the predictions of the of individual trees.

Bagging

The original data set

Game	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Overcast	Mild	High	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Game	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Overcast	Mild	High	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Bootstrap samples 1

Game	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
1	Sunny	Hot	High	FALSE	No
6	Rainy	Cool	Normal	TRUE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No

Bootstrap samples 2

Game	Outlook	Temperature	Humidity	Windy	Play
3	Overcast	Hot	High	FALSE	Yes
2	Sunny	Hot	High	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
4	Rainy	Mild	High	FALSE	Yes
8	Sunny	Mild	High	FALSE	No
12	Overcast	Mild	High	TRUE	Yes
6	Rainy	Cool	Normal	TRUE	No

•

•

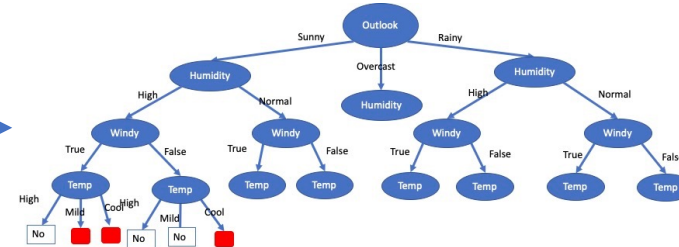
Bootstrap samples N

Game	Outlook	Temperature	Humidity	Windy	Play
11	Sunny	Mild	Normal	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
7	Overcast	Cool	Normal	TRUE	Yes
6	Rainy	Cool	Normal	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
1	Sunny	Hot	High	FALSE	No
6	Rainy	Cool	Normal	TRUE	No

Bagging

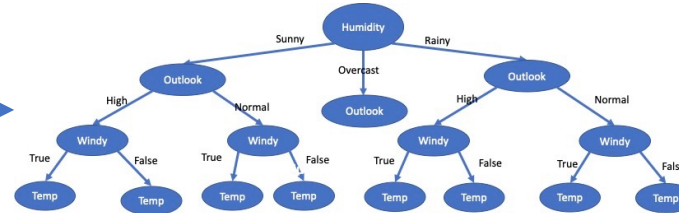
Bootstrap samples 1

Game	Outlook	Temperature	Humidity	Windy	Play
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
1	Sunny	Hot	High	FALSE	No
6	Rainy	Cool	Normal	TRUE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	No



Bootstrap samples 2

Game	Outlook	Temperature	Humidity	Windy	Play
3	Overcast	Hot	High	FALSE	Yes
2	Sunny	Hot	High	TRUE	No
7	Overcast	Cool	Normal	TRUE	Yes
4	Rainy	Mild	High	FALSE	Yes
8	Sunny	Mild	High	FALSE	No
12	Overcast	Mild	High	TRUE	Yes
6	Rainy	Cool	Normal	TRUE	No

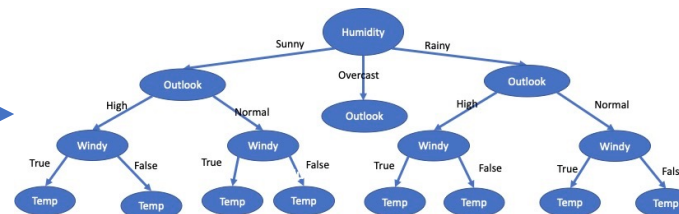


•

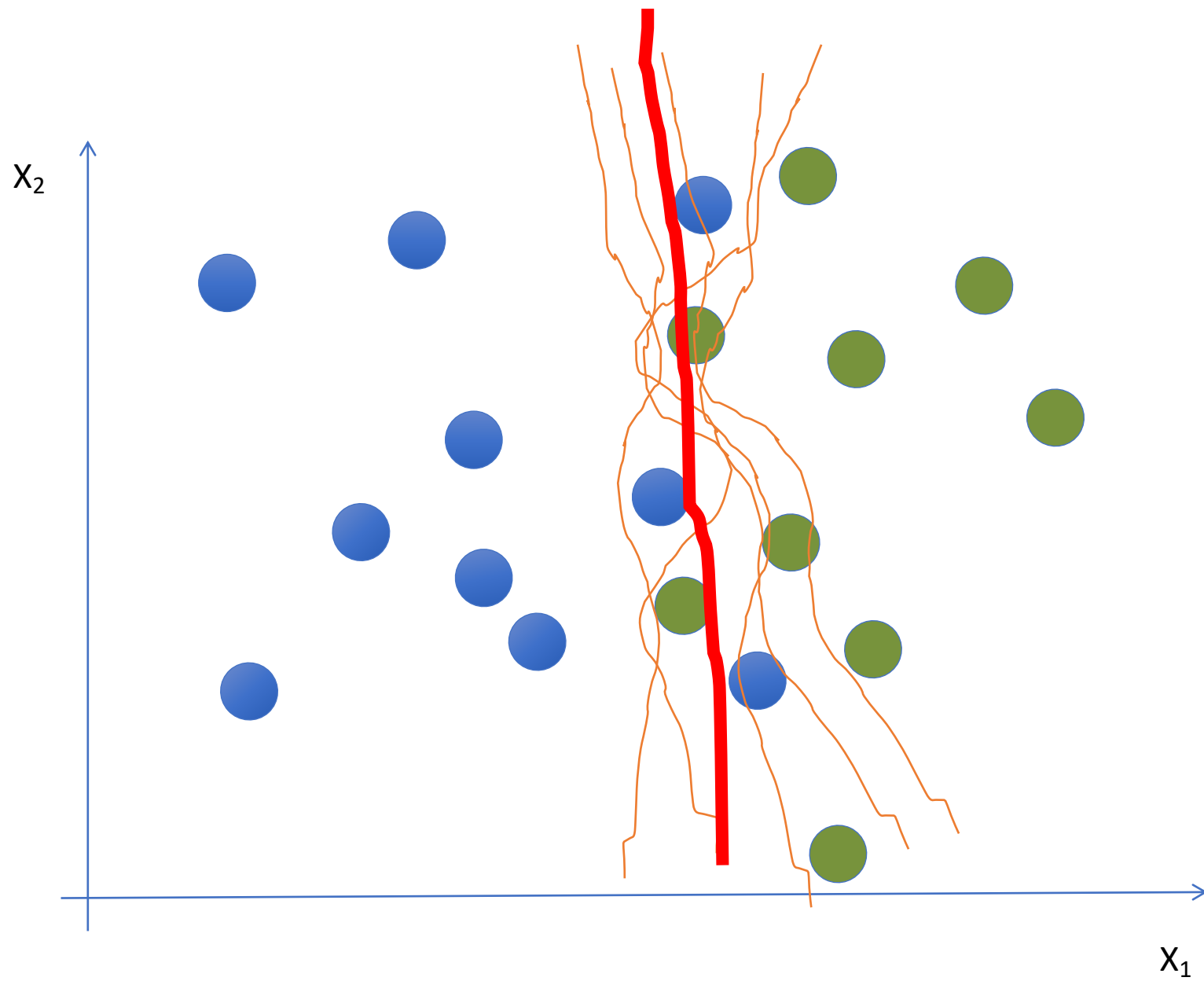
•

Bootstrap samples N

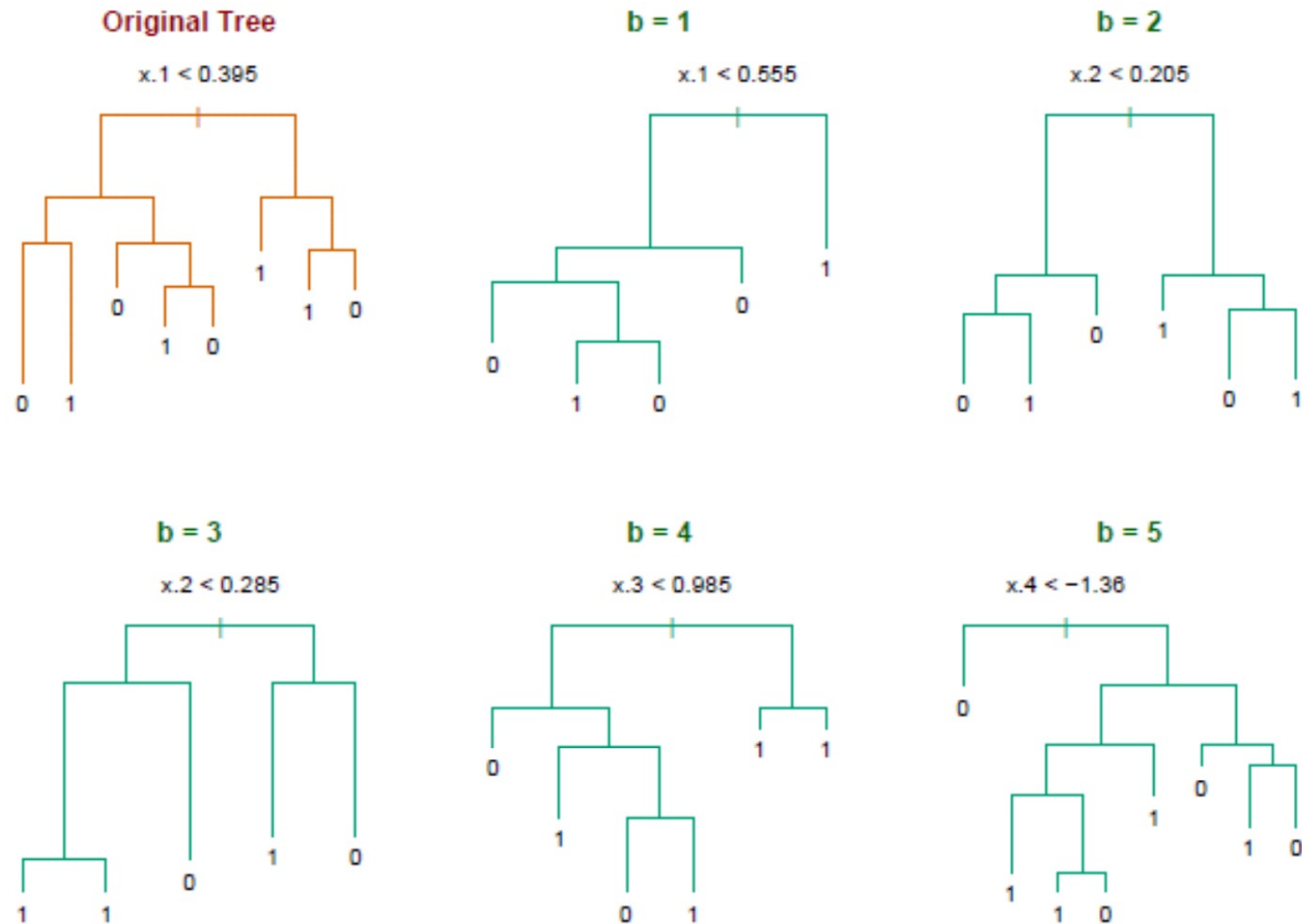
Game	Outlook	Temperature	Humidity	Windy	Play
11	Sunny	Mild	Normal	TRUE	Yes
13	Overcast	Hot	Normal	FALSE	Yes
7	Overcast	Cool	Normal	TRUE	Yes
6	Rainy	Cool	Normal	TRUE	No
3	Overcast	Hot	High	FALSE	Yes
1	Sunny	Hot	High	FALSE	No
6	Rainy	Cool	Normal	TRUE	No



**Take the
majority
vote**



Bagging decision trees



Out-of-Bag Error Estimation

- No cross validation?
- Remember, in bootstrapping we sample with replacement, and therefore **not all observations are used for each bootstrap sample**. On average 1/3 of them are not used!
- We call them out-of-bag samples (OOB)
- We can predict the response for the i -th observation using each of the trees in which that observation was OOB and do this for n observations
- Calculate overall OOB MSE or classification error

Interpretability of Bagged Trees

Recall that bagging typically results in improved accuracy over prediction using a single tree.

Unfortunately, however, it can be difficult to interpret the resulting model. But one main attraction of decision trees is their interpretability!

However, when we bag a large number of trees, it is no longer possible to represent the resulting statistical learning procedure using a single tree, and it is no longer clear which variables are most important to the procedure.

Thus, bagging improves prediction accuracy at the expense of interpretability

Advantage of Bagging

- Reduces overfitting (variance).
- Normally uses one type of classifier.
- Easy to parallelize.

Bagging

Say for each sample b , we calculate $f^b(x)$, then:

$$\hat{f}_{avg}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x)$$

How?

Bootstrap

Construct B (hundreds) of trees (no pruning)

Learn a classifier for each bootstrap sample and average them

Very effective

Bagging - issues

- Consider the following: If we come across a dataset, we can fit a single Decision Tree on it. So far, so good. But, if we do it again, the result will (nearly) be the same in the case of Decision Trees.
- Suppose that there is one very strong predictor (feature) in the data set, along with several other moderately strong predictors. Then all bagged trees will select the strong predictor at the top of the tree and therefore all trees will look similar.
- Averaging these models gives us another with the same bias, but with lower variance.

Resolve Bagging - issues

We can penalize the splitting (like in pruning) with a penalty term that depends on the number of times a predictor is selected at a given length

We can restrict how many times a predictor can be used

We only allow a certain number of predictors

Resolve Bagging - issues

What if we consider only a subset of the features at each split?

We will still get correlated trees unless ...we **randomly** select the subset !

Random Forest

Random Forests

Random Forests were invented by Leo Breiman, 1999. The idea here is to fit numerous Decision Trees on the training set in a special way, giving an equally large number of tree models (=functions). Afterwards, these trees are combined into a single model

As in bagging, we build a number of decision trees on bootstrapped training samples each time a split in a tree is considered, a random sample of m features is chosen as split candidates from the full feature.

Note that if $m = p$, then this is bagging.

Random Forests

Random forests are popular. Leo Breiman's and Adele Cutler maintains a random forest website where the software is freely available, and of course it is included in every ML/STAT package

<http://www.stat.berkeley.edu/~breiman/RandomForests/>

Random Forests Algorithm

For $b = 1$ to B :

(a) Draw a bootstrap sample Z^* of size N from the training data.

(b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

- i. Select m variables at random from the p variables.
- ii. Pick the best variable/split-point among the m .
- iii. Split the node into two daughter nodes.

Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x we do: for classification: majority vote.

Random Forest Tuning

The inventors make the following recommendations:

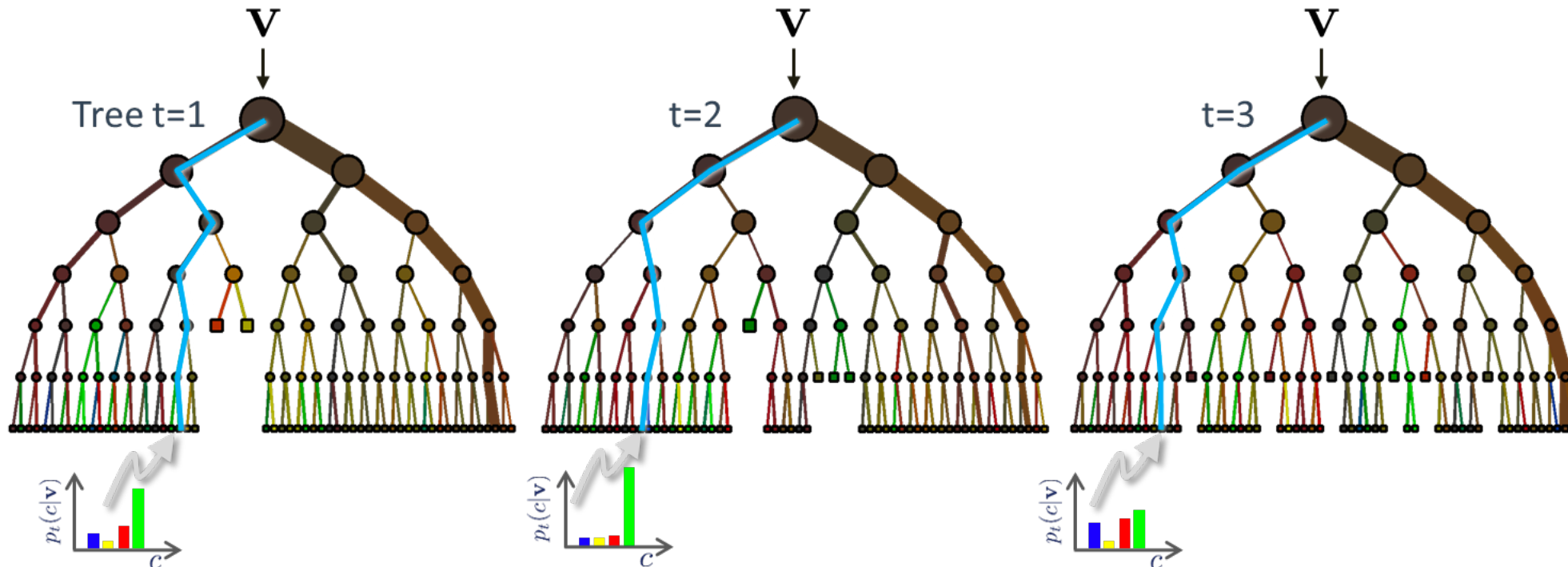
- For classification, the default value for m is $\sqrt{p}$ and the minimum node size is one.

In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.

Like with Bagging, we can use OOB and therefore RF can be fit in one sequence, with cross-validation being performed along the way. Once the OOB error stabilizes, the training can be terminated.

Classification of testing data using Random Forest

During testing the same unlabelled test input data v is pushed through each component tree. At each internal node a test is applied and the data point sent to the appropriate child. The process is repeated until a leaf is reached.



Example

- 4,718 genes measured on tissue samples from 349 patients.
- Each gene has different expression
- Each of the patient samples has a qualitative label with 15 different levels: either normal or 1 of 14 different types of cancer.

Use random forests to predict cancer type based on the 500 genes that have the largest variance in the training set.

Random Forests Issues

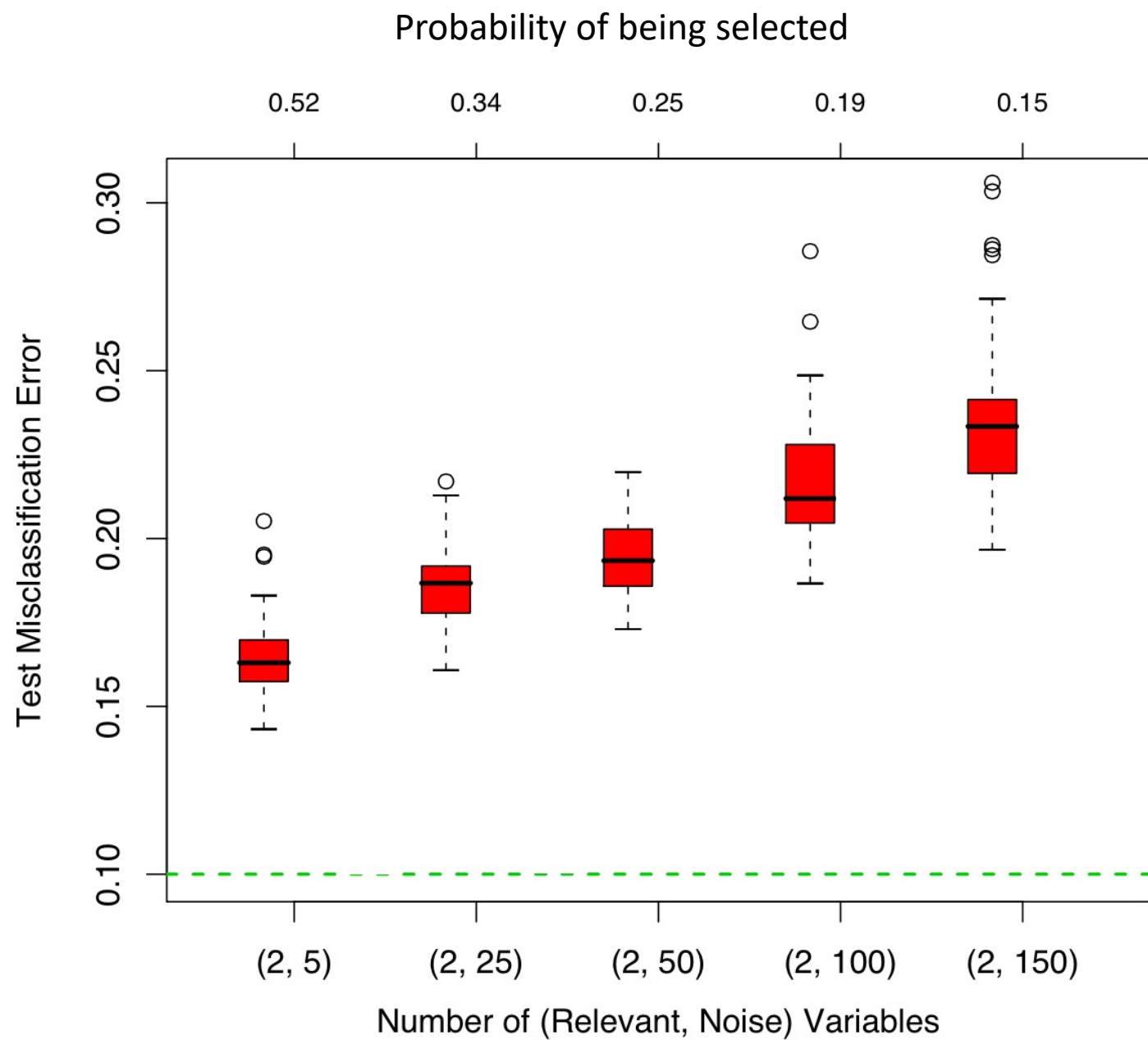
When the number of variables is large, but the fraction of relevant variables is small, random forests are likely to perform poorly when m is small

Why?

Because:

At each split the chance can be small that the relevant variables will be selected

For example, with 3 relevant and 100 not so relevant variables the probability of any of the relevant variables being selected at any split is ~ 0.25



Can RF overfit?

Random forests “cannot overfit” the data wrt to number of trees.

Why?

The number of trees, B does not mean increase in the flexibility of the model

Boosting

Boosting

Boosting is a general approach that can be applied to many statistical learning methods for regression or classification.

Bagging: Generate multiple trees from bootstrapped data and average the trees.

Recall bagging results in i.d. trees and not i.i.d.

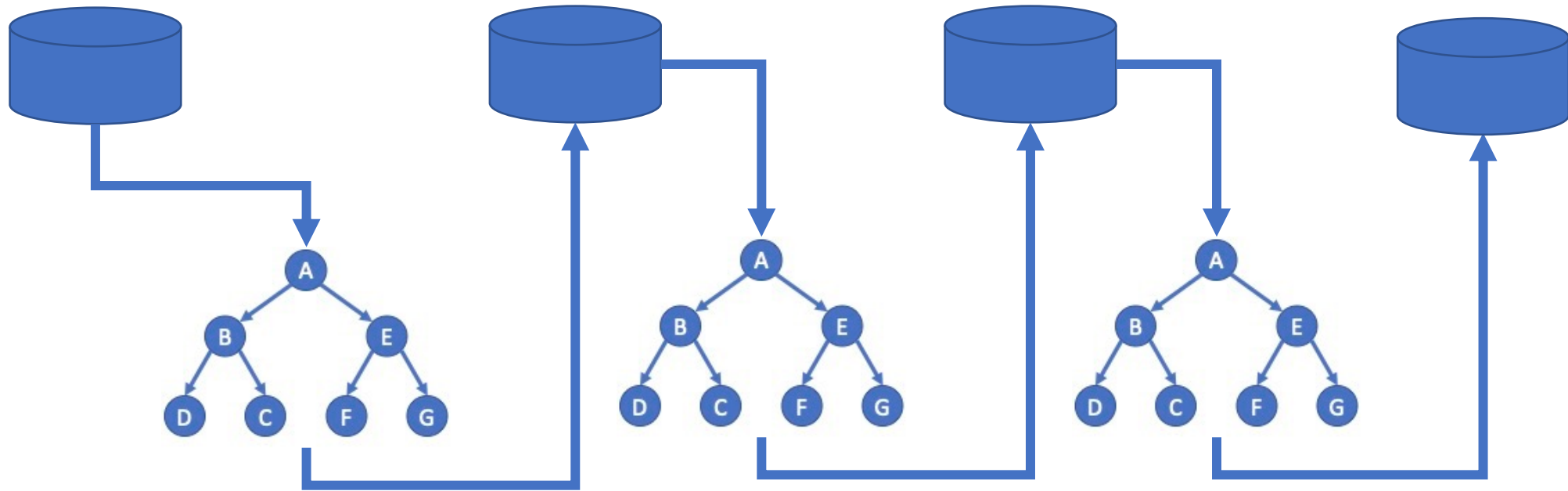
RF produces i.i.d (or more independent) trees by randomly selecting a subset of predictors at each step

Boosting

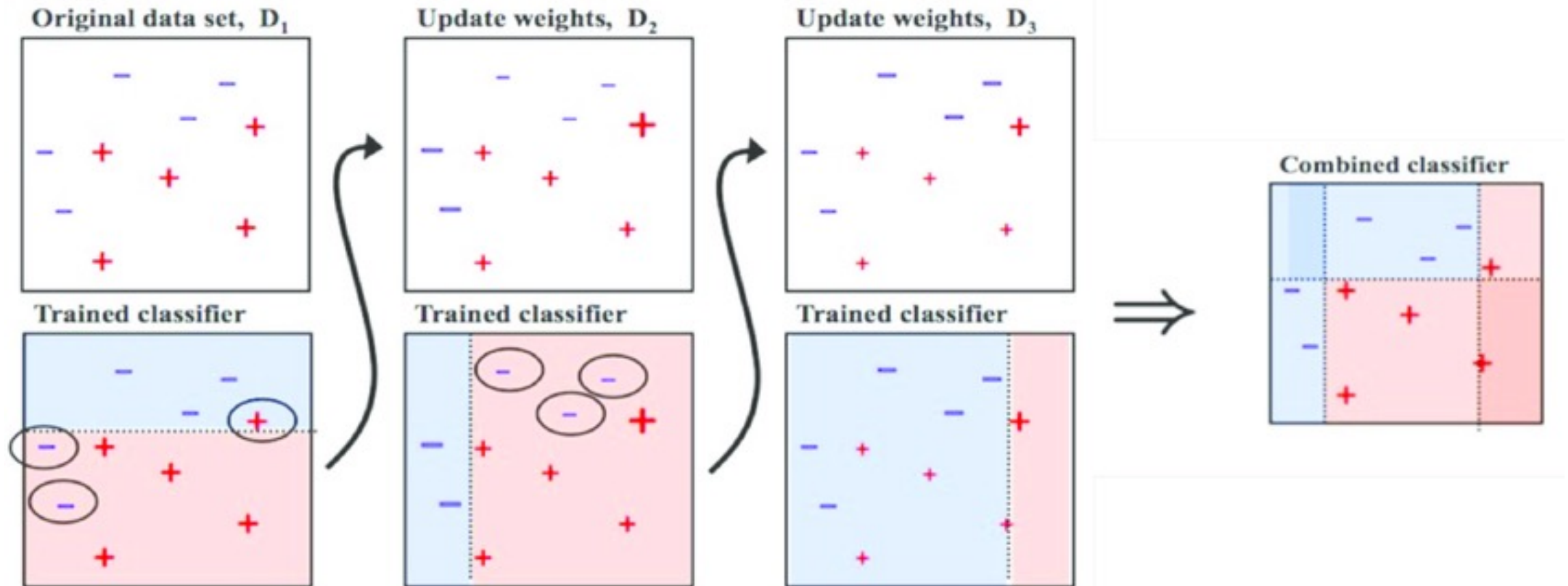
Boosting works very differently.

1. Boosting does not involve bootstrap sampling
2. Trees are grown sequentially: each tree is grown using information from previously grown trees
3. Like bagging, boosting involves combining a large number of decision trees, $T^1, \dots, T^B$

A gradient boosting model



Training of boosting model



Sequential fitting

Given the current model,

- we fit a decision tree to the **residuals** from the model. Response variable now is the residuals and not class label.
- We then add this new decision tree into the fitted function in order to update the residuals
- The learning rate has to be controlled

Boosting tuning parameters

- Number of trees : Adding excessive number of trees can lead to overfitting, so it is important to stop at the point where the loss value converges.
- Tree depth : Shorter trees are preferred over more complex trees. Limit the number of levels in a tree to 4-8.
- Number of observations per split : This imposes a minimum constraint on the amount of training data at a training node before a split can be considered

Boosting - Issues

- It is sensitive to outliers since every classifier is obliged to fix the errors in the predecessors and that can cause overfitting.
- It is almost impossible to scale up.
- It is almost impossible to justify the decision.
- It is computationally expensive.

For more information

- Friedman, J.H., 2017. ***The elements of statistical learning: Data mining, inference, and prediction.*** springer open.

Chapters (10 and 15).