CARTE ML Workshop

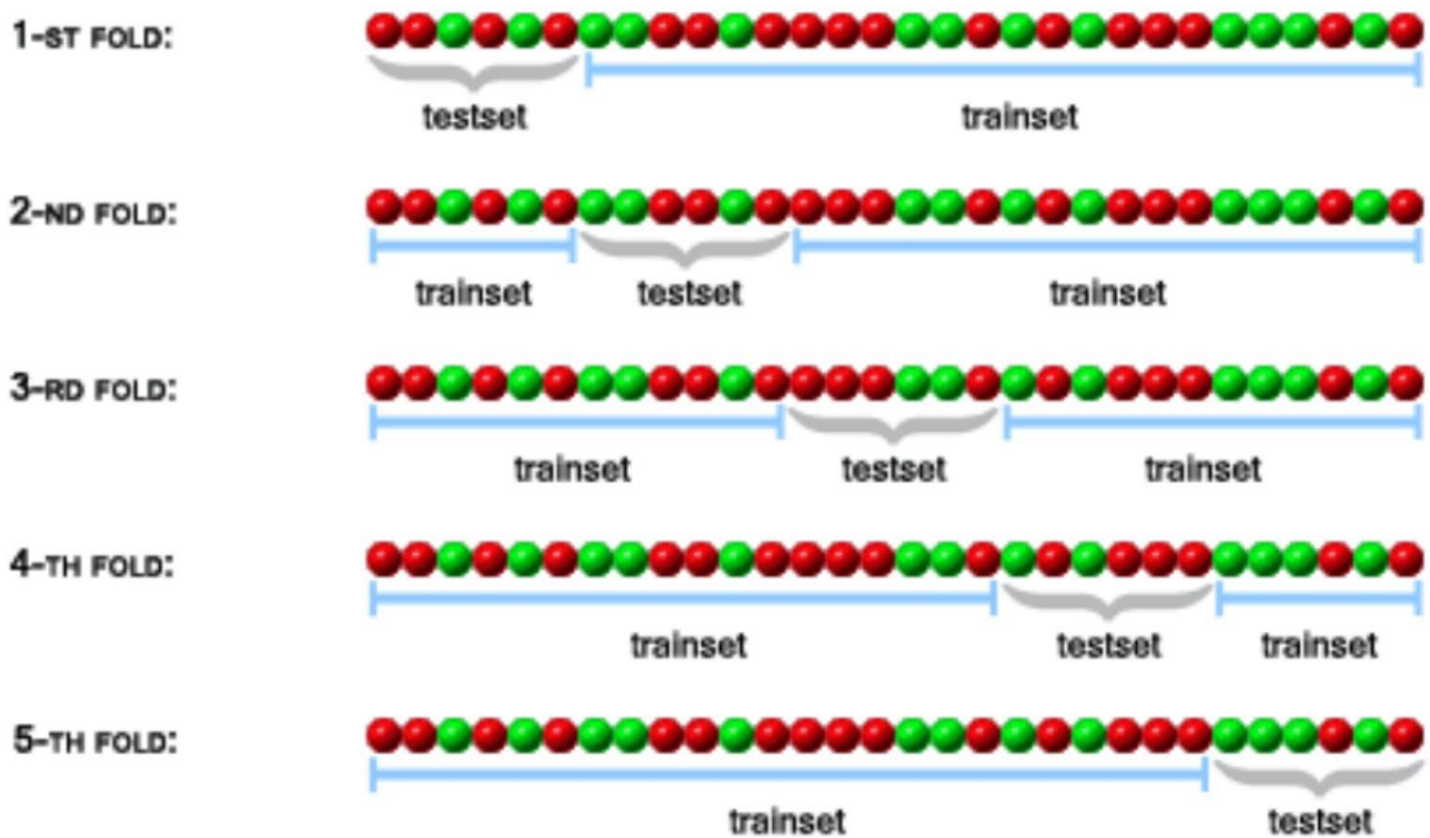
Ensemble Models

Cross validation

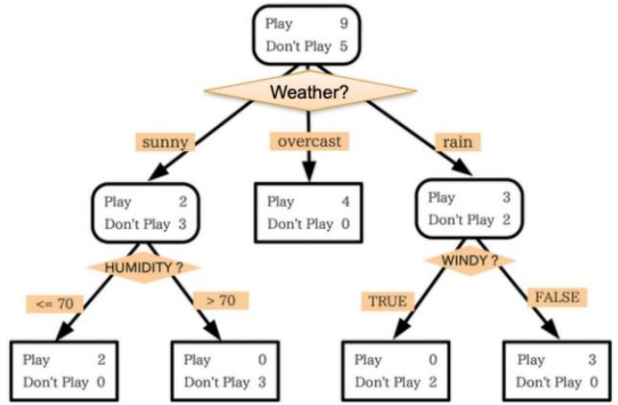
1. Divide your data into n parts
2. Hold 1 part as “test set” or “hold out set”
3. Train classifier on remaining n-1 parts “training set”
4. Compute test error on test set
5. Repeat above steps n times, once for each n-th part
6. Compute the average test error over all n folds
   (i.e., cross-validation test error)

Example: one run of *5-fold* cross validation

You should do a **few runs** and **compute the average**
(e.g., error rates if that's your evaluation metrics)



Model Combination



Any single model may have high variance in its predictions

- Decision tree: predictions can be brittle, small perturbation in data can lead to misclassification

Creating more stable models:

- Come up with a completely new model
- Combine (unstable) models intelligently!

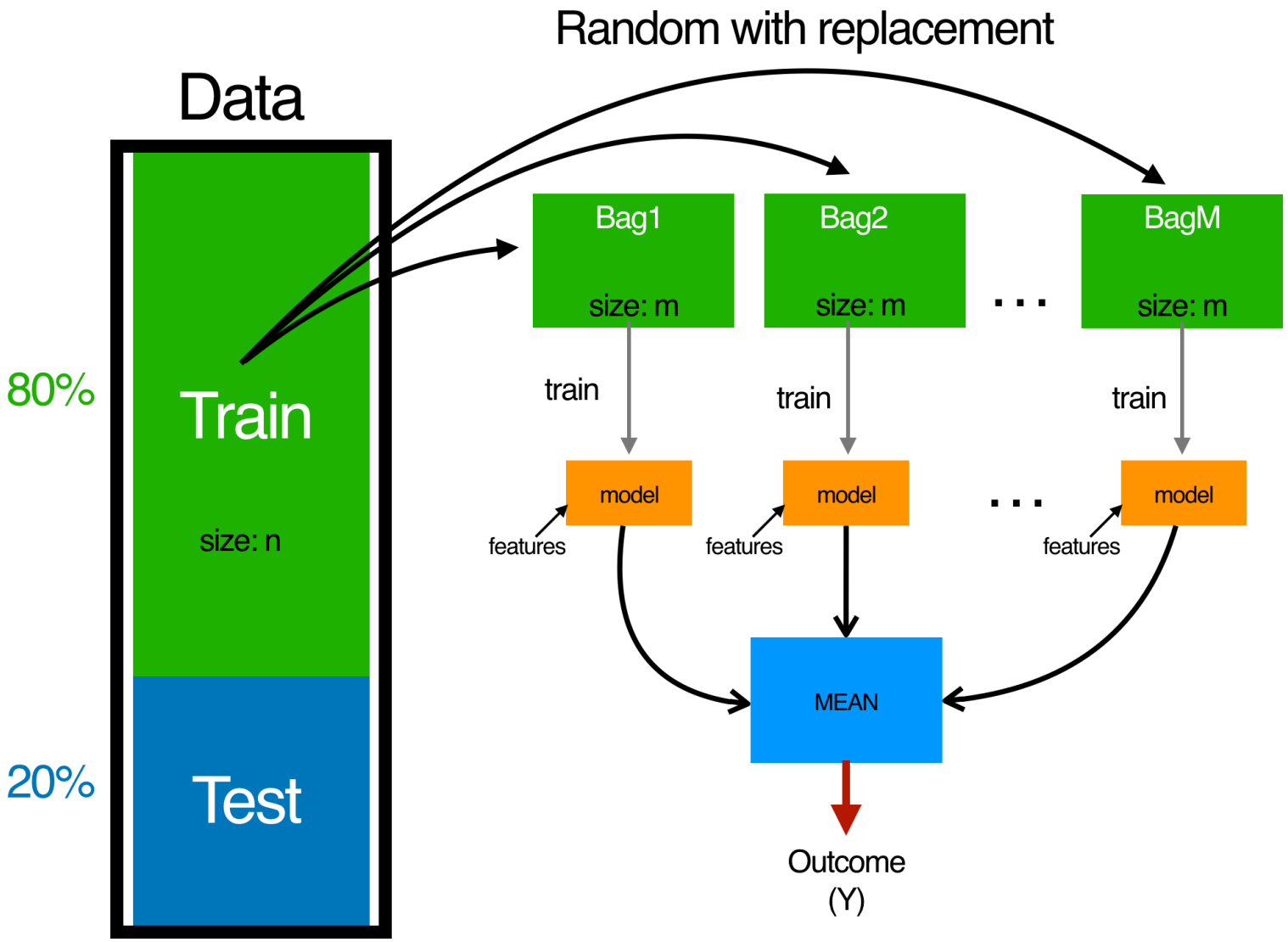
Bootstrap Sampling

Data: $S = \{(x_i, y_i)\}_{i = 1,\ldots,n}$

Bootstrap Sample: subsample of S, drawn with replacement

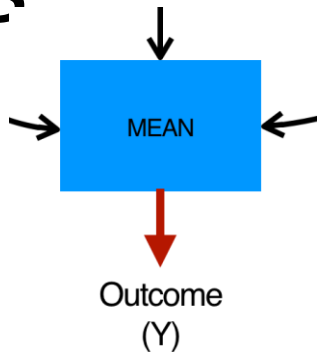
https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205

Bootstrap Aggregating (Bagging)



http://www.ub.edu/stat/docencia/EADB/bagging.html

Bagging

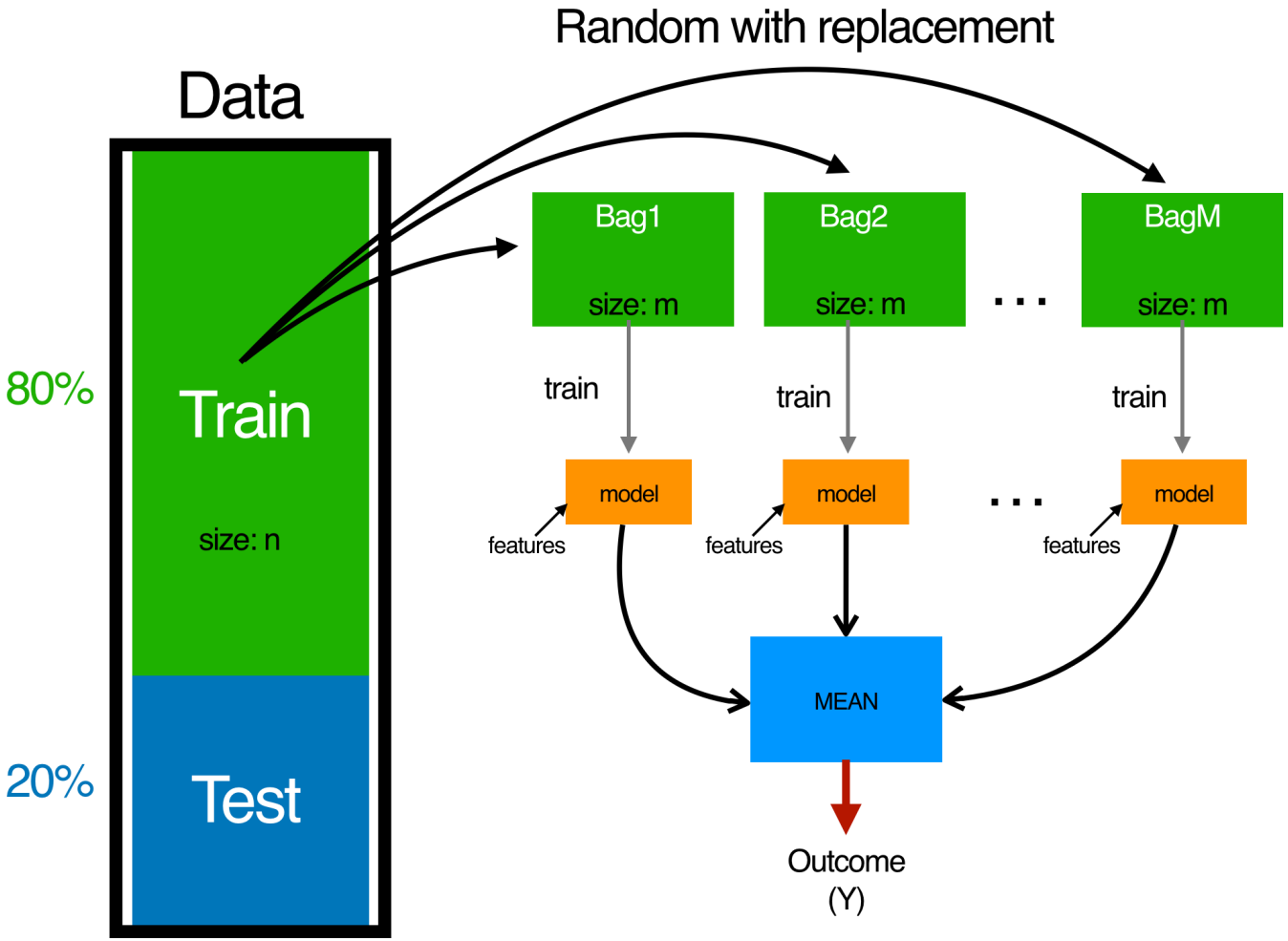


Aggregation:

- Majority vote!

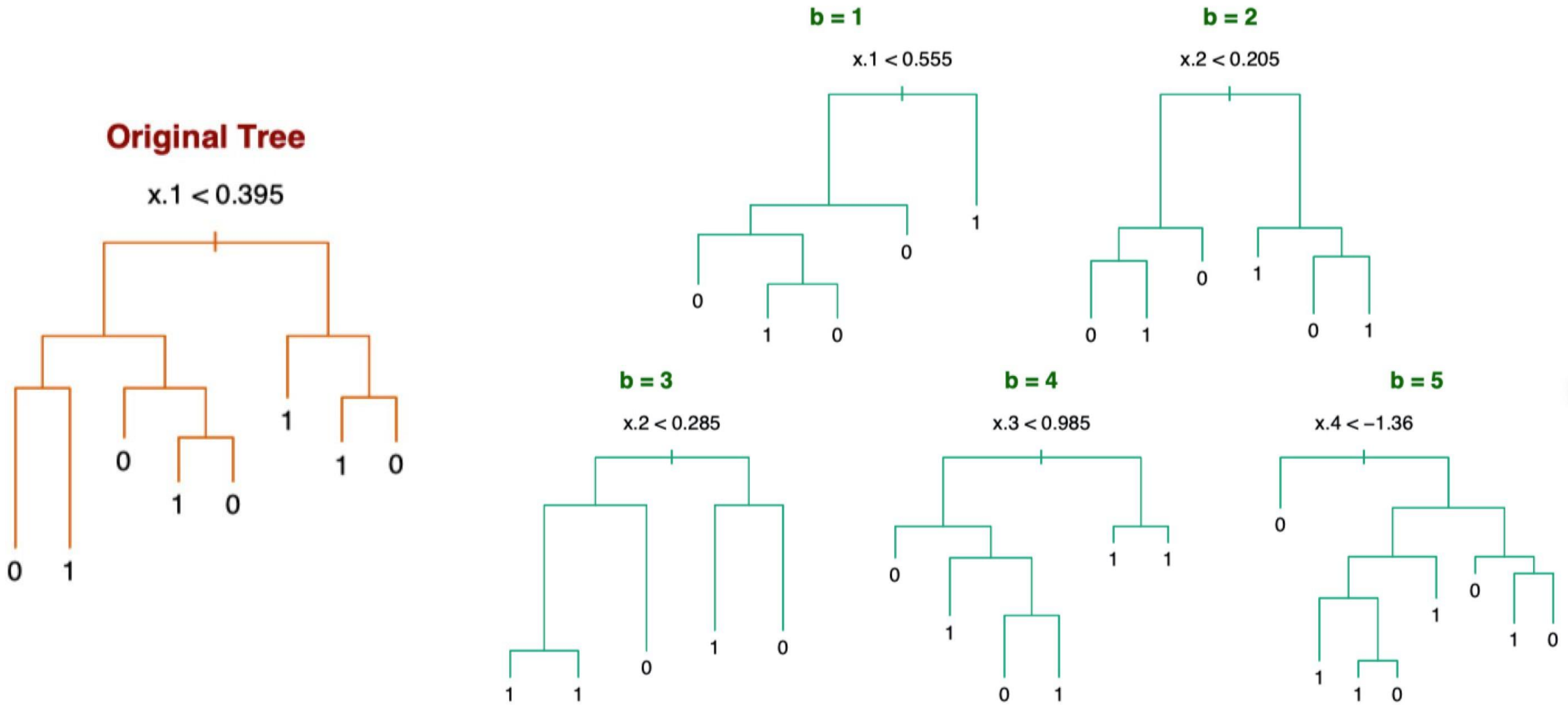
Benefits:

- Even if single model overfits, on average they will not



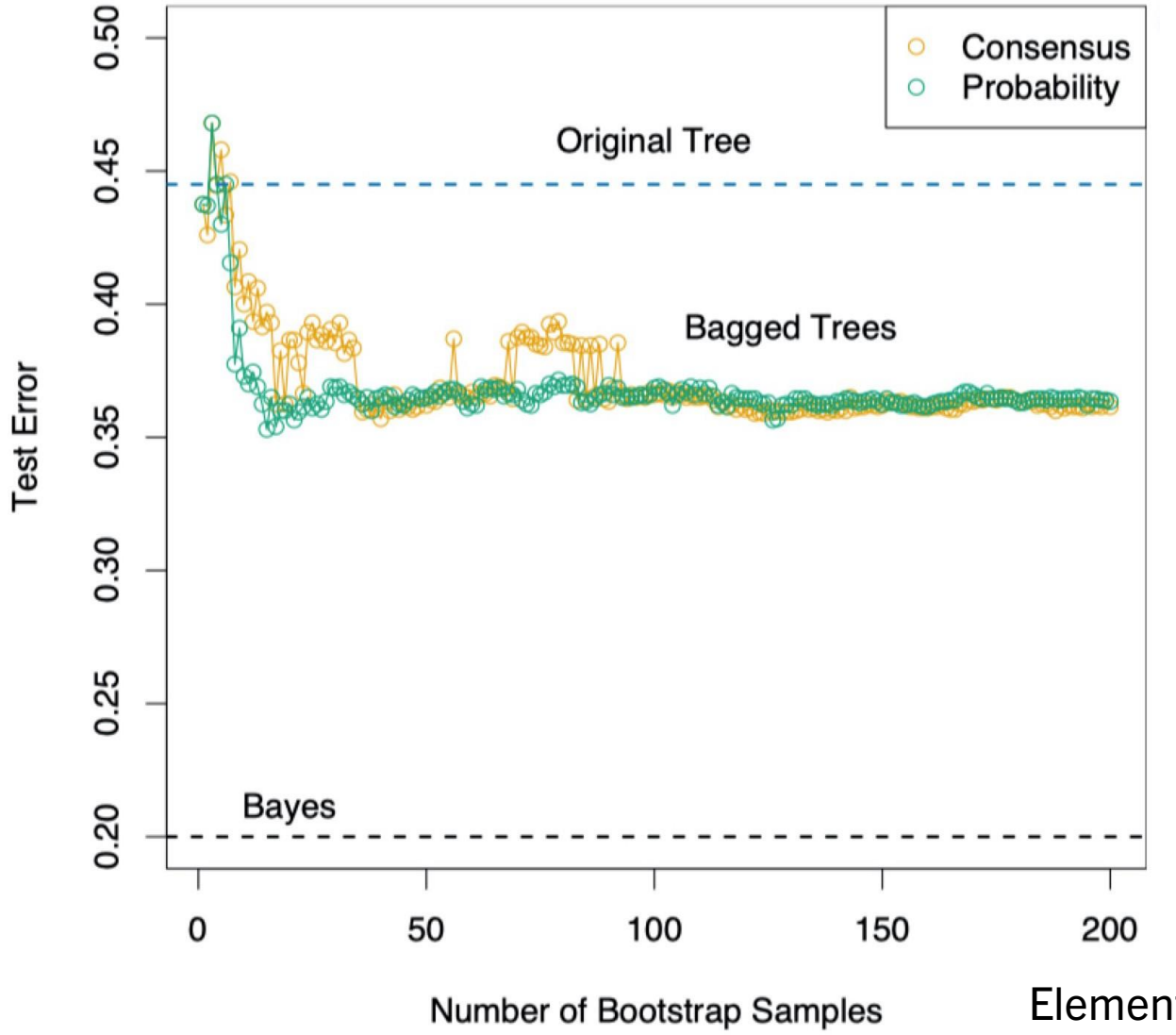
http://www.ub.edu/stat/docencia/EADB/bagging.html

Bagging with Decision Trees



Elements of Statistical Learning, Chapter 8.7

Predicting with bagged decision trees



Consensus: **proportion of trees that predict a given class**

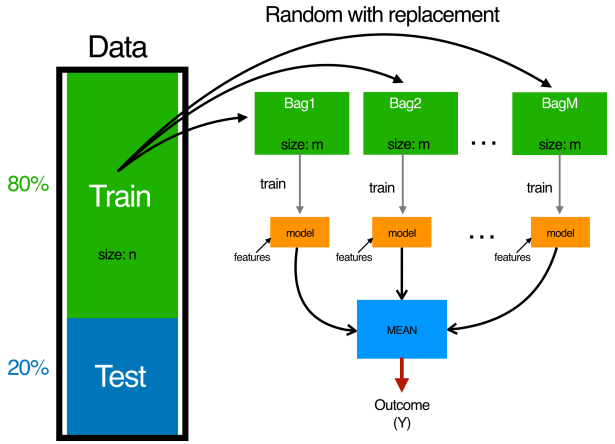
Probability:

- For b-th tree, look at the leaf corresponding to test sample
- What proportion of the training data had the same label as test sample?
- Average those proportions out

Prediction: in both cases, predict class with highest score

Elements of Statistical Learning, Chapter 8.7

When should use Bagging?



High-variance classifiers, e.g., decision trees.

Bagging reduces variance by providing an alternative approach to regularization.

Even if each of the learned classifiers are individually overfit, they are likely to overfit to different things.

Through voting, we can overcome a significant portion of this overfitting. In

practice, bagging tends to reduce variance and increase bias.

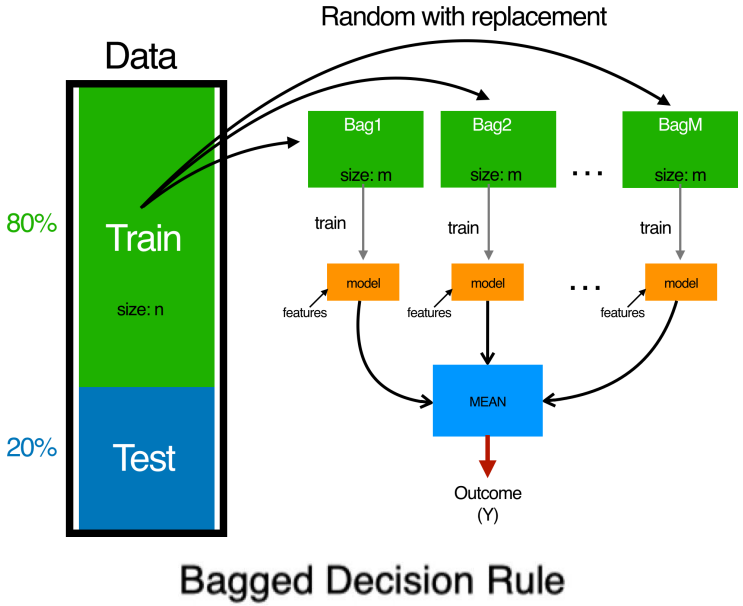
Final words on Bagging

Advantages

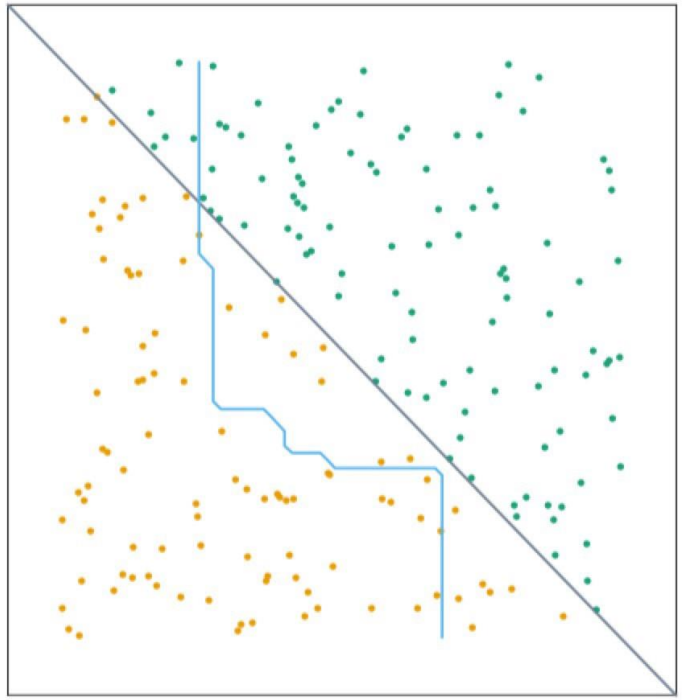
- Reduces model variance
- Trivial to implement and use

Caveats

- Destroys any interpretability in the base model
- May not capture simple patterns

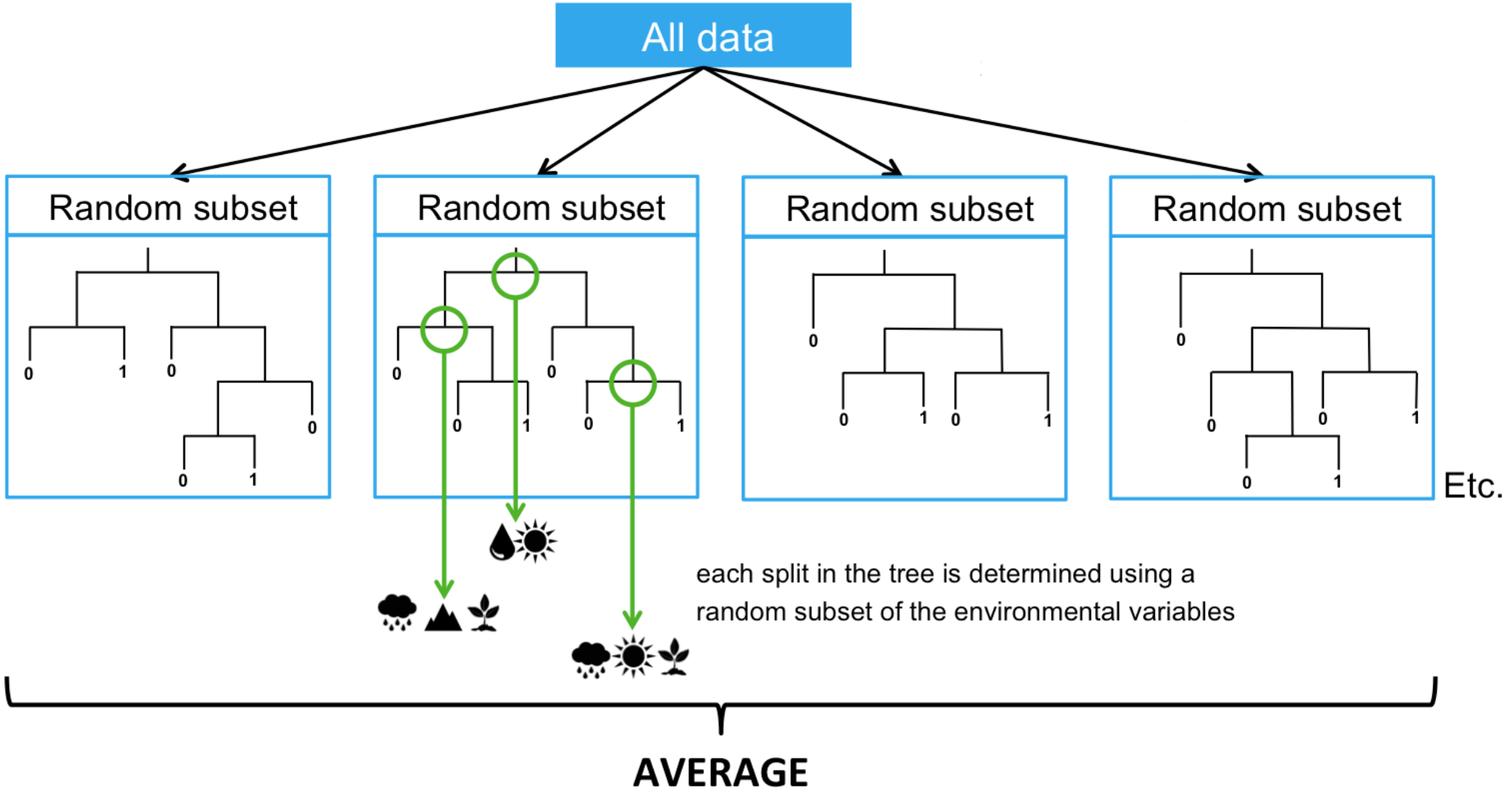


Bagged Decision Rule



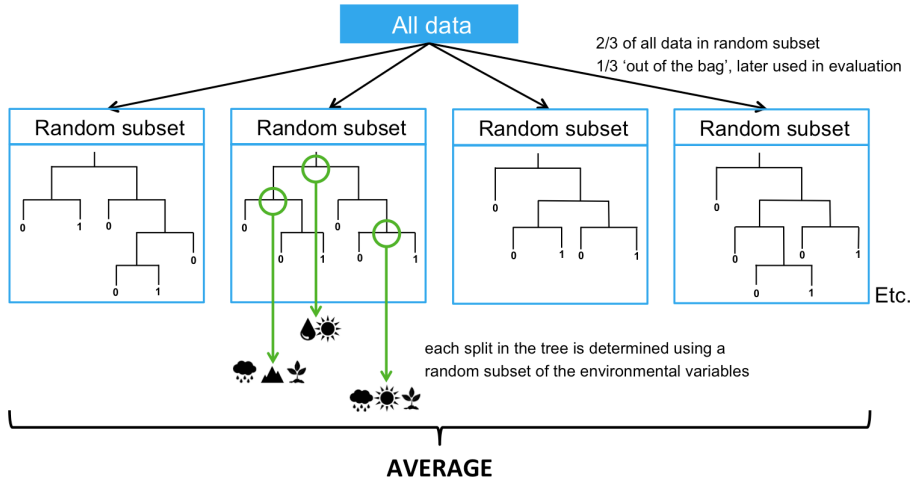
Random Forest

Similar to Bagging Decision Trees except:



https://support.bccvl.org.au/support/solutions/articles/6000083217-random-forest

Random Forest



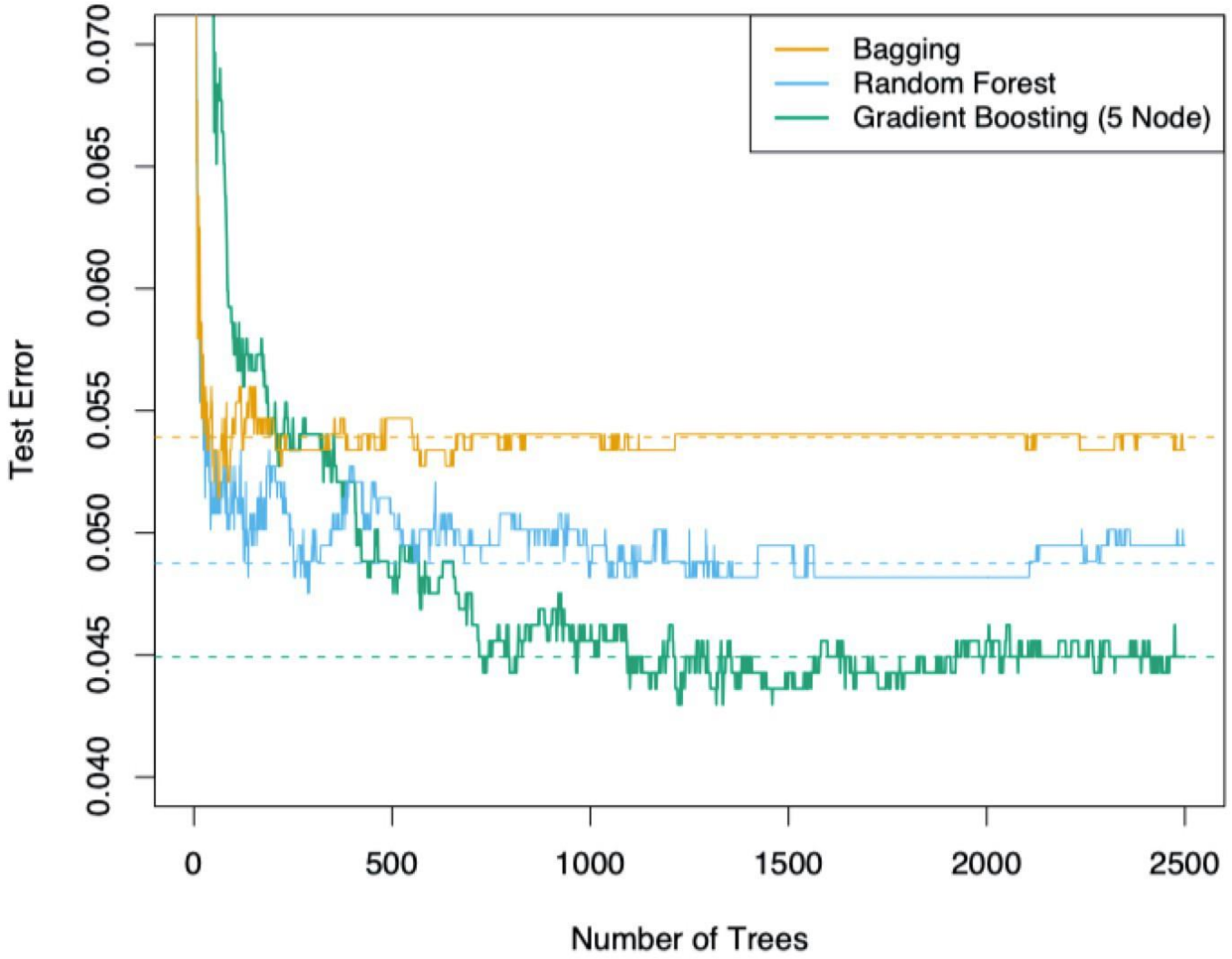
- One of the most popular models!
- Usually more effective than Bagging Decision Trees
- Typically, use 100s of trees (hyper-parameter to be tuned!)
- For data with d features, num. random features used for splitting:
  - Classification: $\sqrt{\boldsymbol{d}}$
  - Regression: $\boldsymbol{d/3}$

Example: Spam classification

Dataset of 4000 emails: y = (spam, not spam)

x = % of words equal to 'business', 'address', etc.;

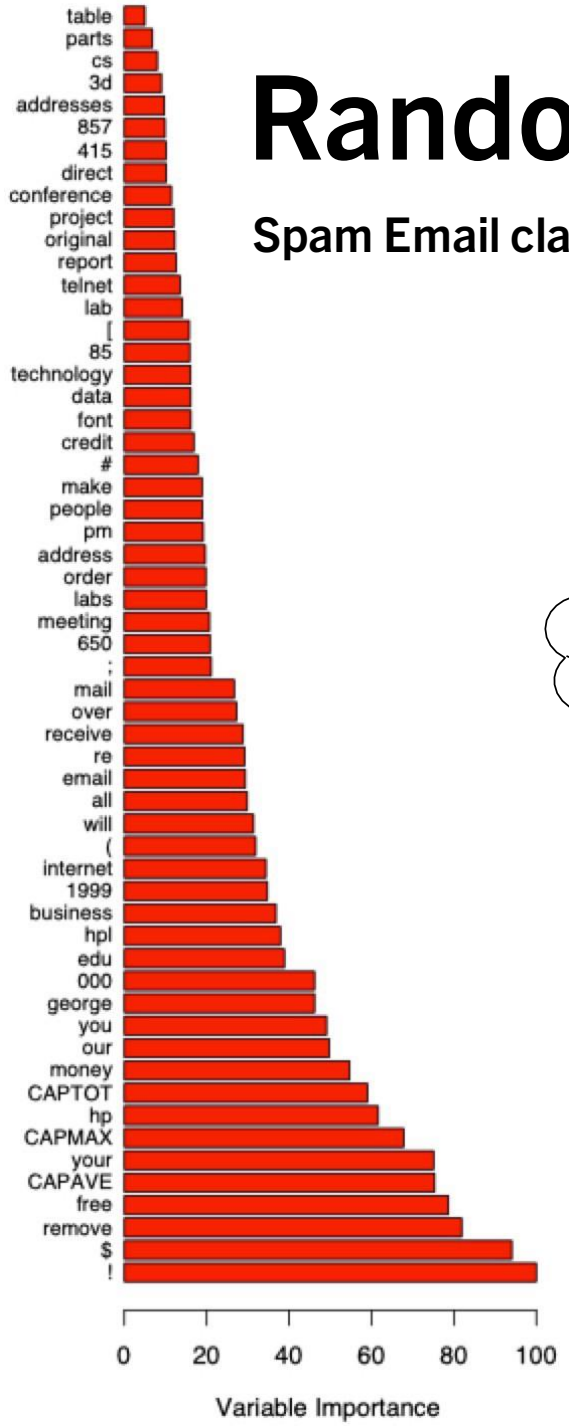
average length of uninterrupted sequences of capital letters, ...



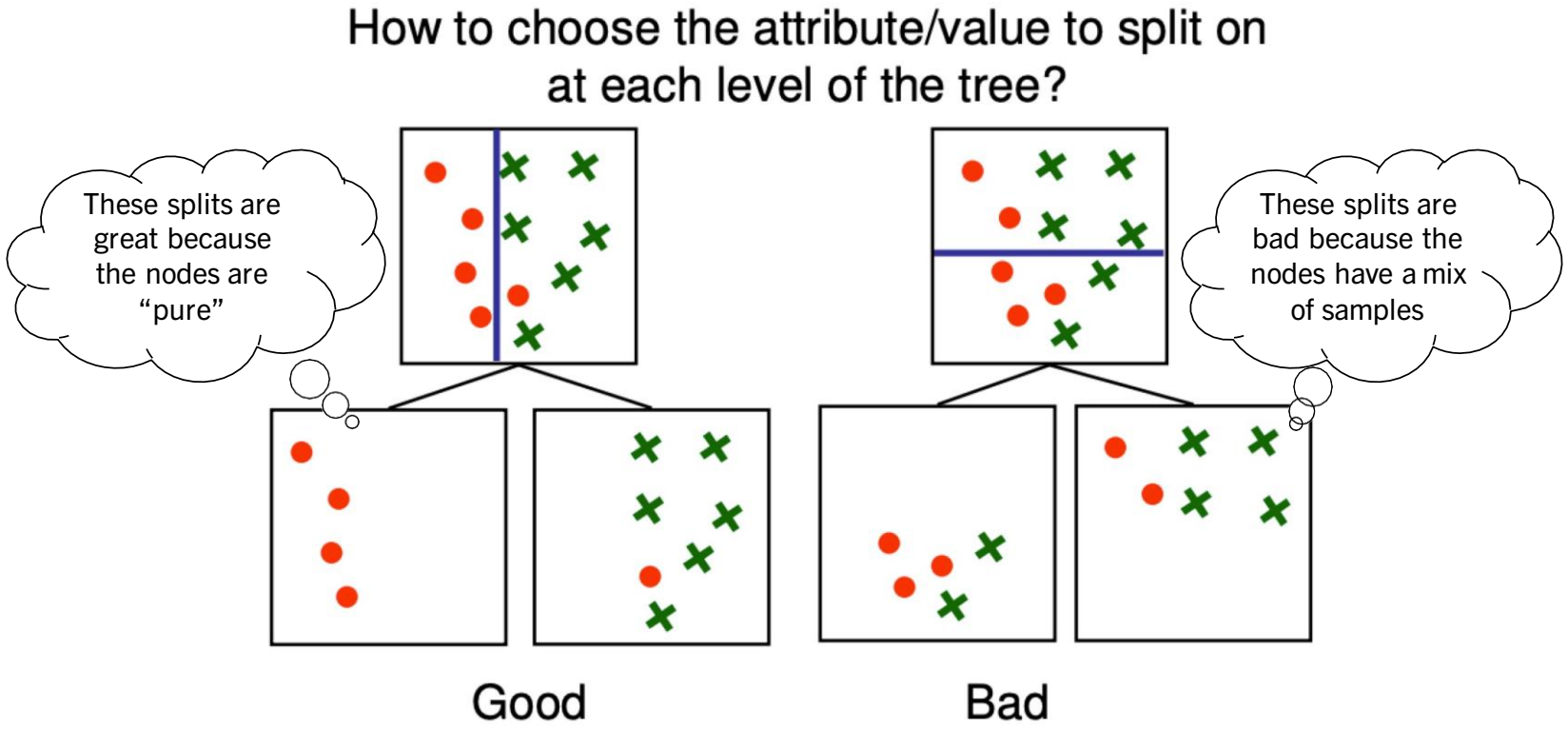
Ignore this for now

Random Forest: Feature Importance

Spam Email classification dataset



words



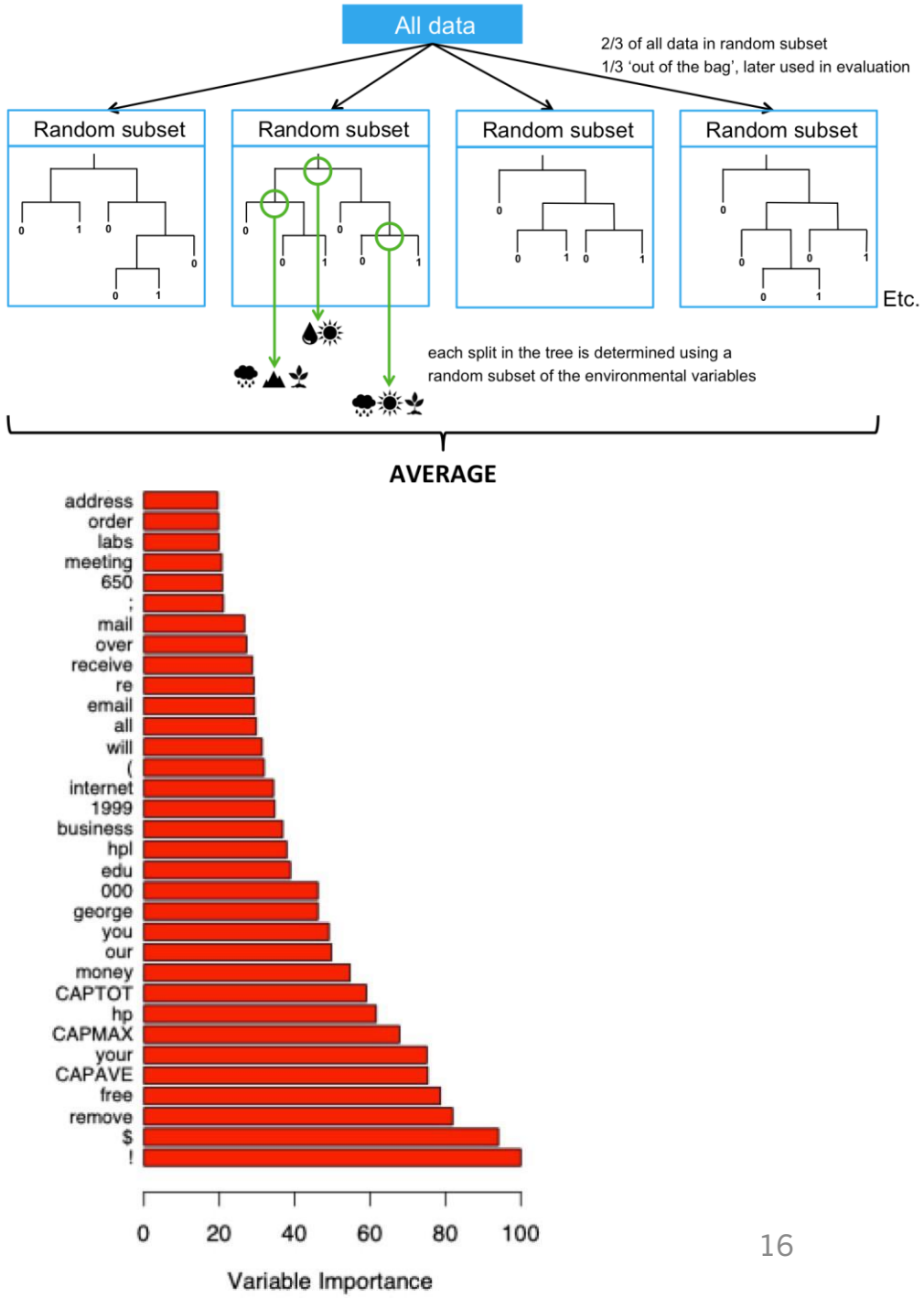
Final words on RF

Advantages

- Feed it data as is without preprocessing
- Fast training because of feature sampling!
- Easy parallel training and prediction

Caveats

- Not suitable for sparse data (why?)



Boosting

- goal: automatically categorize type of call requested by phone customer (Collect, CallingCard, PersonToPerson, etc.)
  - `yes I'd like to place a collect call long distance please` (Collect)
  - `operator I need to make a call but I need to bill it to my office` (ThirdNumber)
  - `yes I'd like to place a call on my master card please` (CallingCard)
  - `I just called a number in sioux city and I musta rang the wrong number because I got the wrong party and I would like to have that taken off of my bill` (BillingCredit)

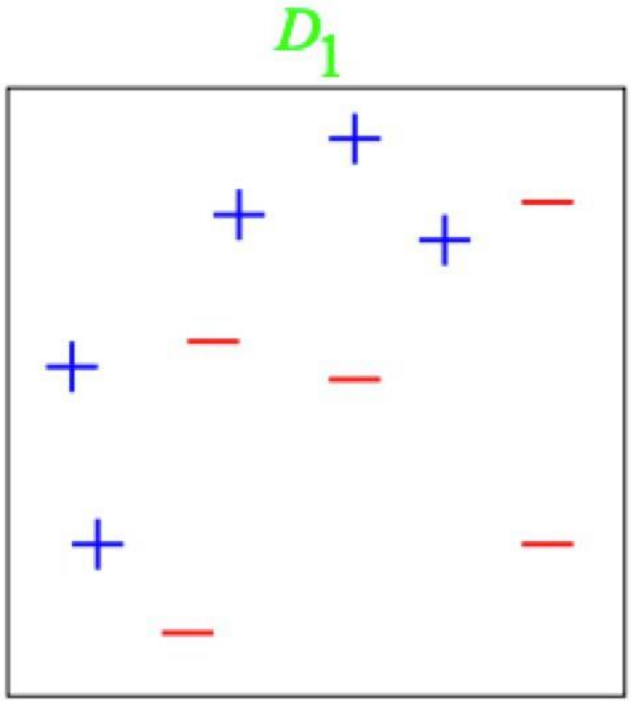
Boosting

- observation:
  - easy to find "rules of thumb" that are "often" correct
    - e.g.: "IF 'card' occurs in utterance THEN predict 'CallingCard' "
  - hard to find single highly accurate prediction rule

- goal: automatically categorize type of call requested by phone customer (Collect, CallingCard, PersonToPerson, etc.)
  - `yes I'd like to place a collect call long distance please` (Collect)
  - `operator I need to make a call but I need to bill it to my office` (ThirdNumber)
  - `yes I'd like to place a call on my master card please` (CallingCard)
  - `I just called a number in sioux city and I musta rang the wrong number because I got the wrong party and I would like to have that taken off of my bill` (BillingCredit)

Theory and Applications of Boosting by Schapire (NIPS 2007 tutorial)

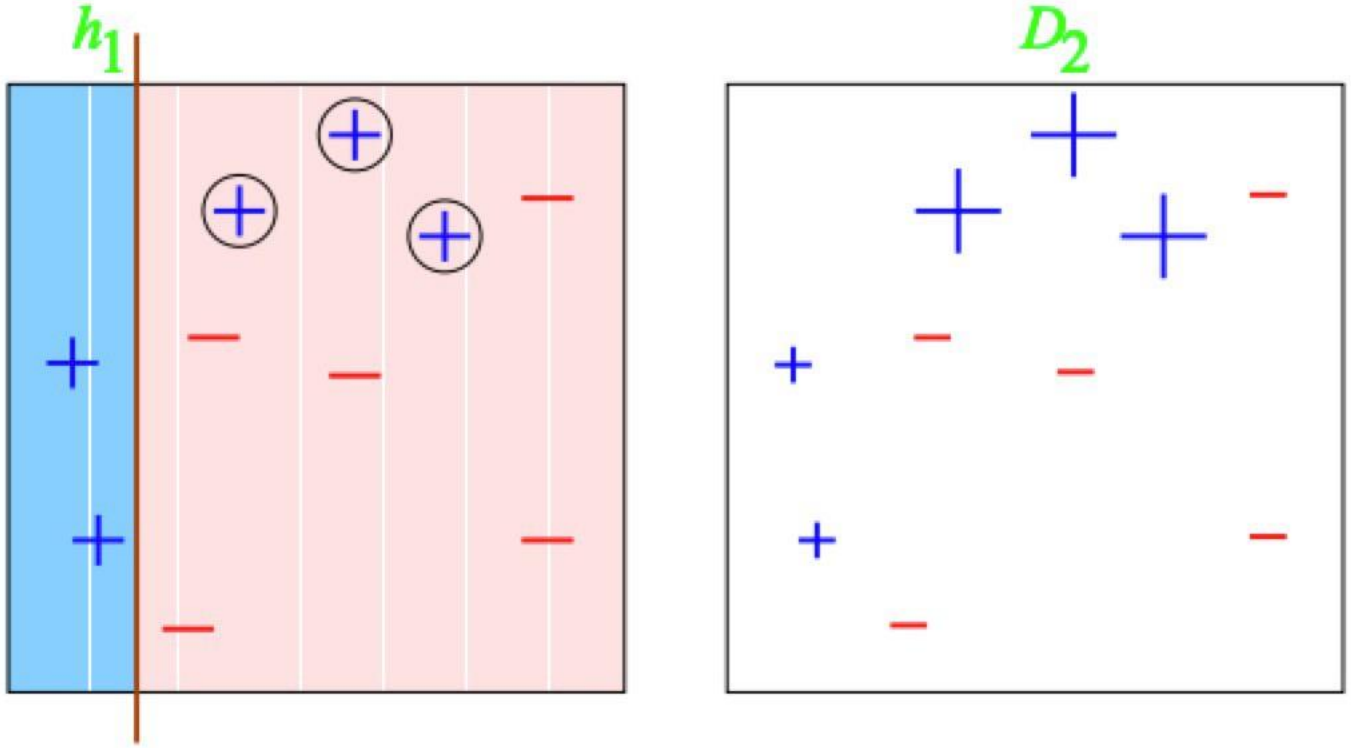
Boosting: A Toy Example



"rule of thumb" classifiers = vertical or horizontal half-planes

Boosting: A Toy Example

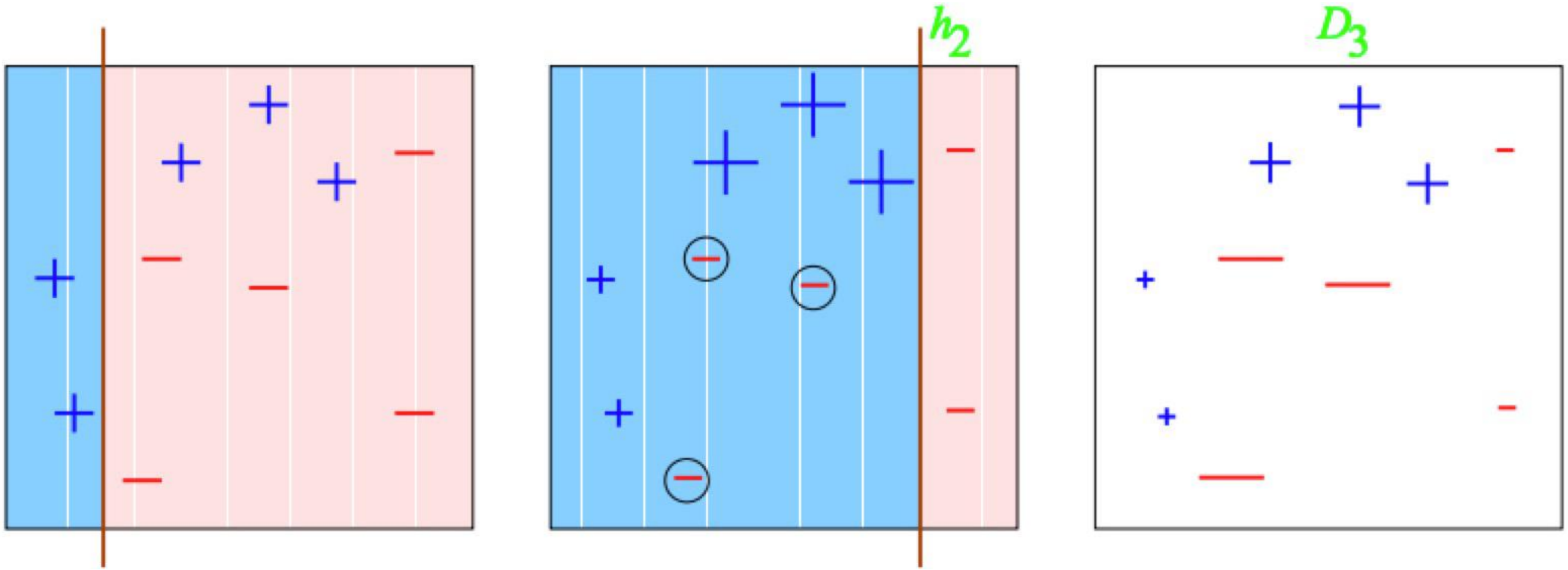
Size of the data point: penalty for misclassifying the point



“rule of thumb” classifiers = vertical or horizontal half-planes

Boosting: A Toy Example

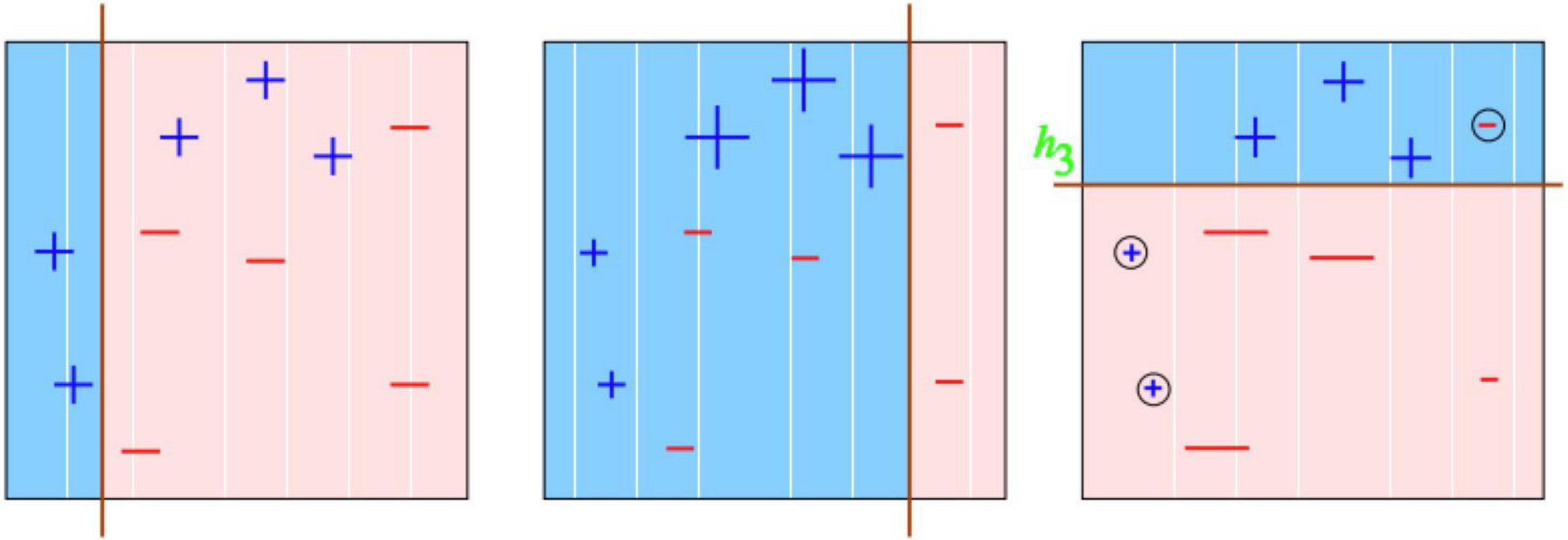
Size of the data point: penalty for misclassifying the point



"rule of thumb" classifiers = vertical or horizontal half-planes

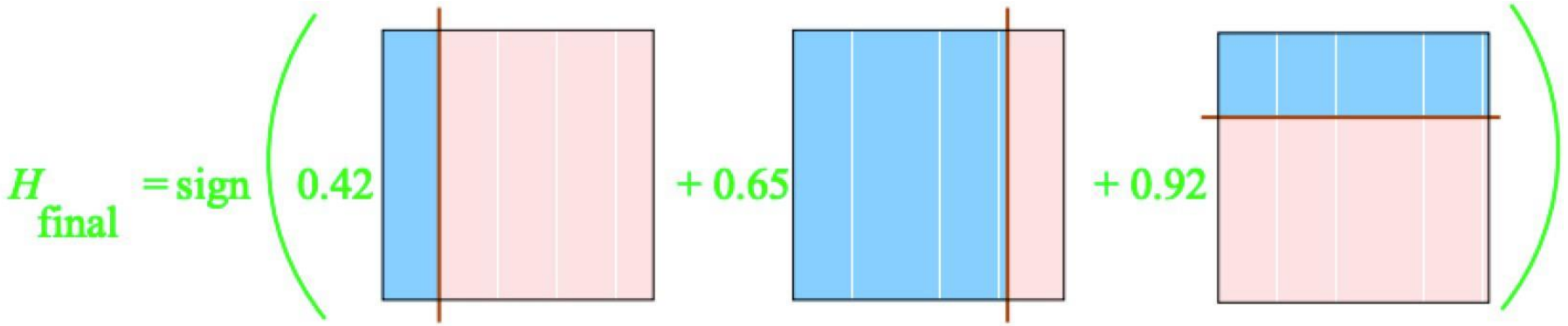
Boosting: A Toy Example

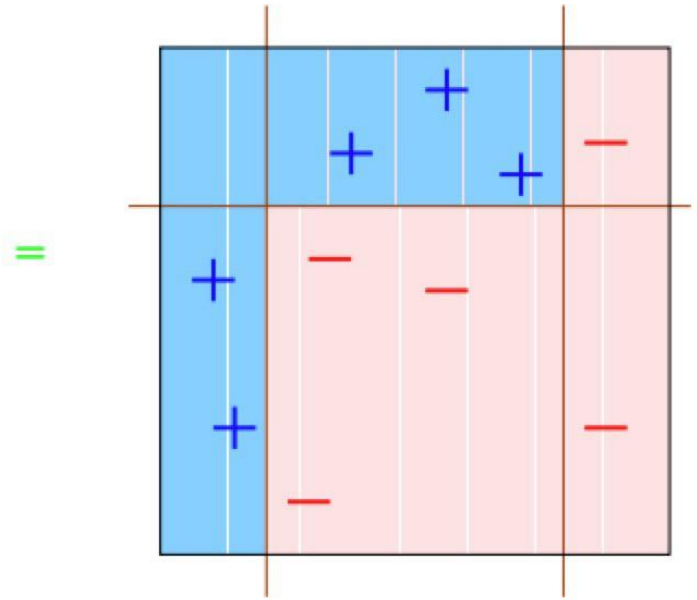
Size of the data point: weight, penalty for misclassifying the point

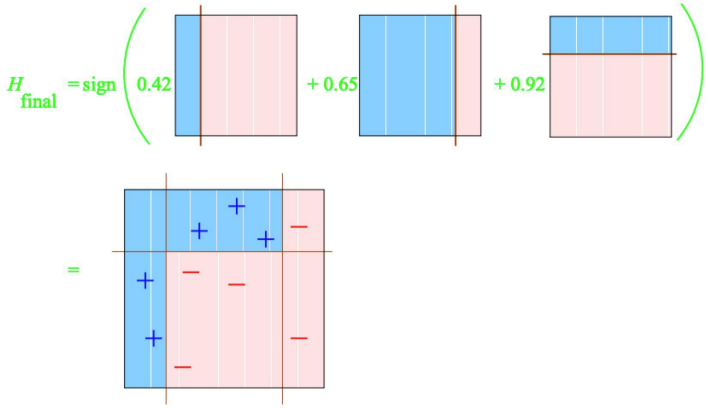


“rule of thumb” classifiers = vertical or horizontal half-planes

Boosting: A Toy Example







Boosting in a Nutshell

A general method of converting rough rules of thumb into highly accurate prediction rule.

Technically:

- assume given "weak" learning algorithm that can consistently find classifiers ("rules of thumb") at least slightly better than random, say, accuracy ≥ 55%
- given sufficient data, a boosting algorithm can provably construct single classifier with very high accuracy

Xgboost: A scalable tree boosting system

T Chen, C Guestrin - Proceedings of the 22nd acm sigkdd international …, 2016 - dl.acm.org

Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning …

☆ Cited by 3057 Related articles All 14 versions

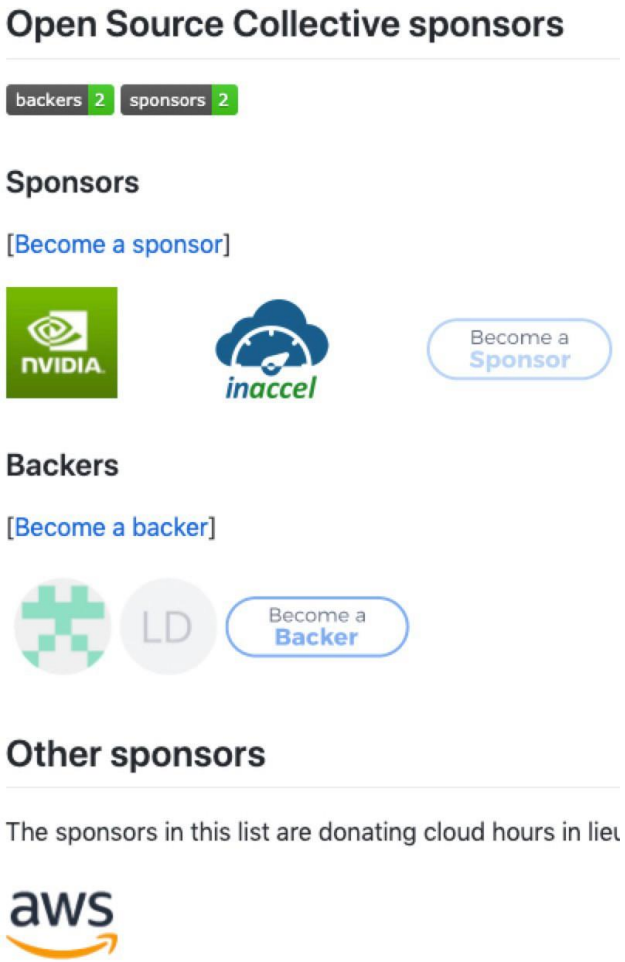
dmlc

XGBoost eXtreme Gradient Boosting

Very efficient, extensive tree boosting code by Tianqi Chen (UW)

A standard tool in data science applications

Easy to parallelize and run on billion-scale problems

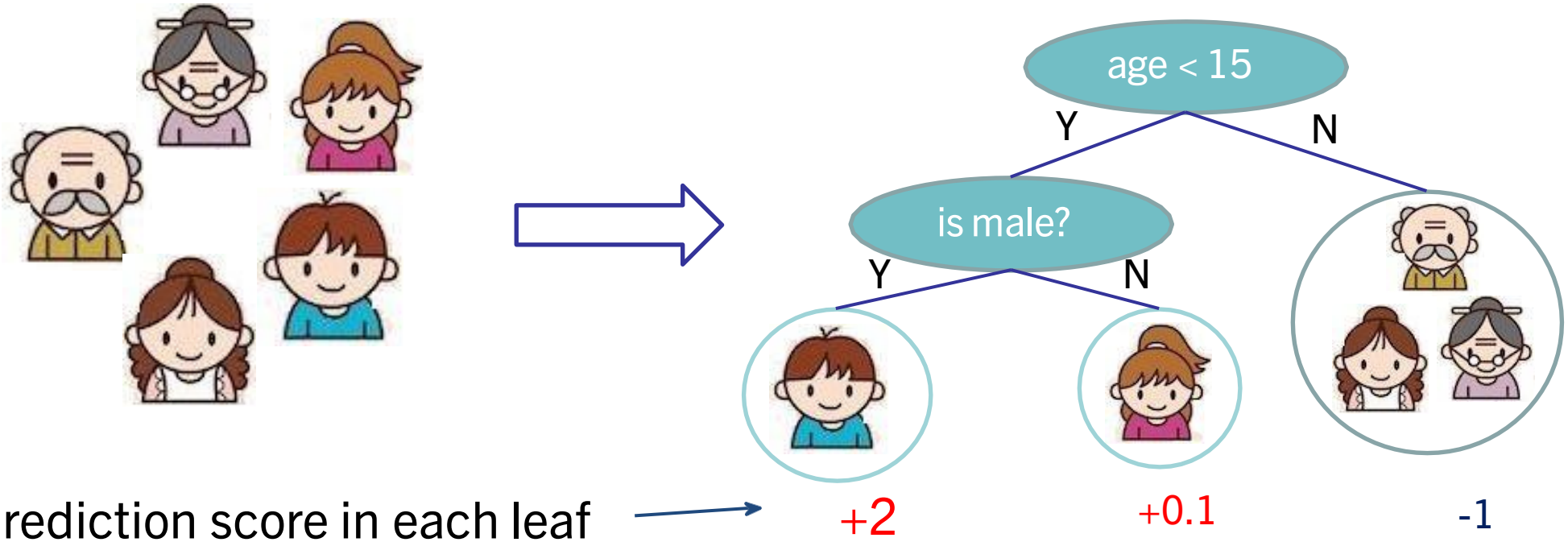


Regression Tree

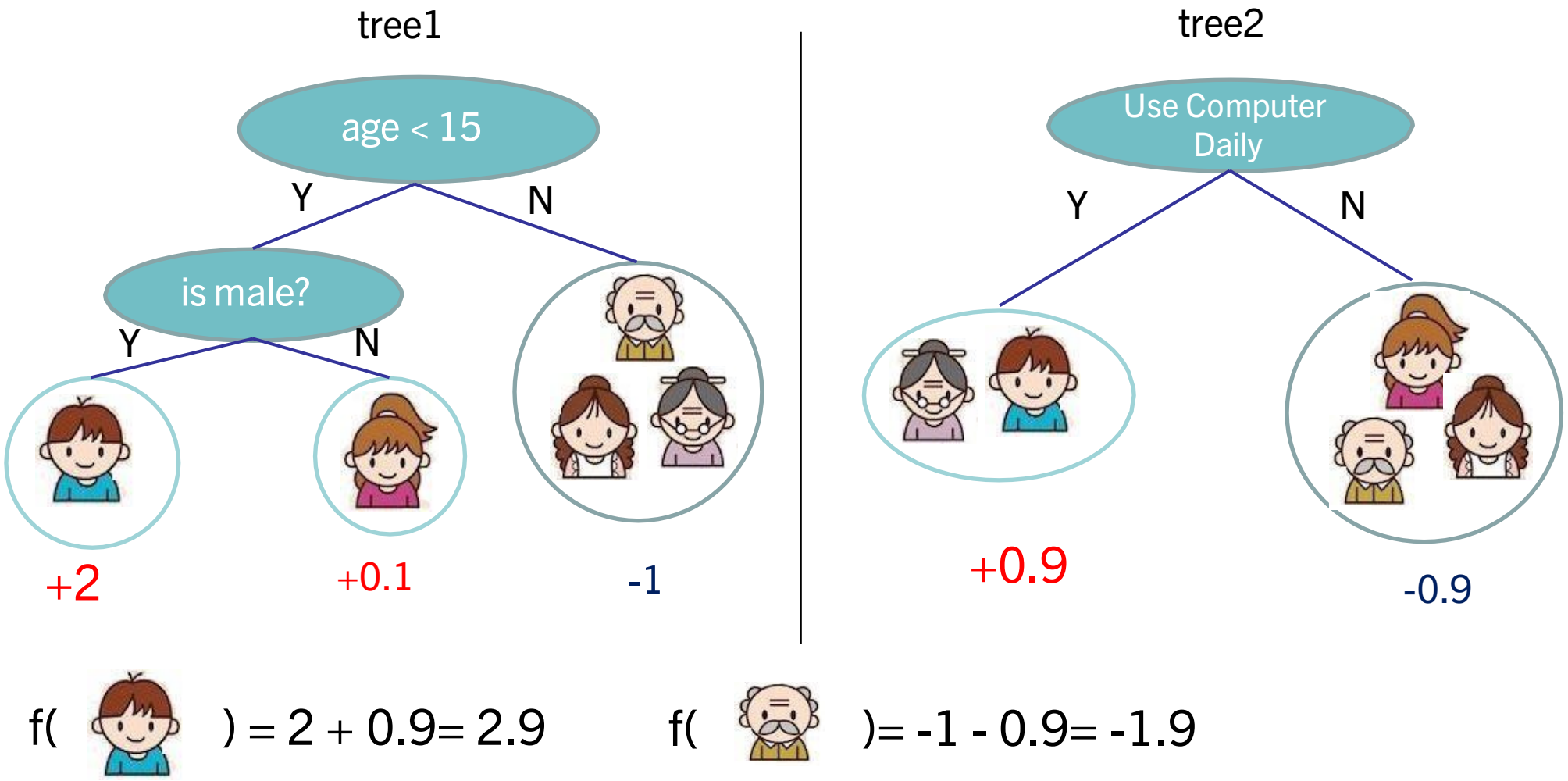
- Regression tree (also known as CART)
- This is what it would look like for a commercial system

Input: age, gender, occupation, …

Does the person like computer games



When Trees forms a Forest (Tree Ensembles)



f( ) = 2 + 0.9= 2.9

f( )= -1 - 0.9= -1.9

Trade off in Learning

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Training Loss measures how well model fit on training data

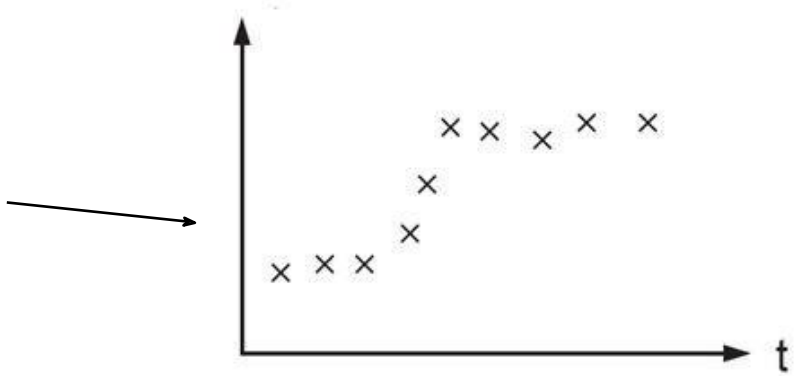
Regularization, measures complexity of trees

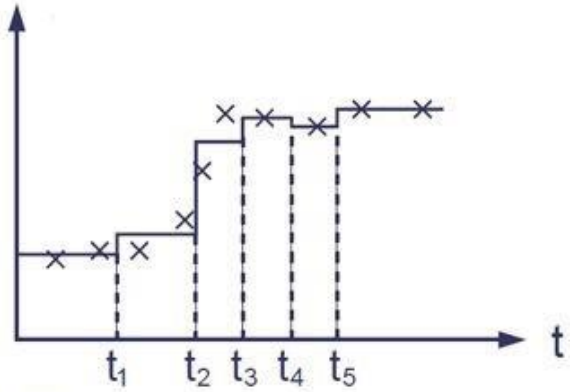
- Optimizing training loss encourages predictive models
  - Fitting well in training data at least get you close to training data which is hopefully close to the underlying distribution
- Optimizing regularization encourages simple models
  - Simpler models tends to have smaller variance in future predictions, making prediction stable

Why do we need regularization

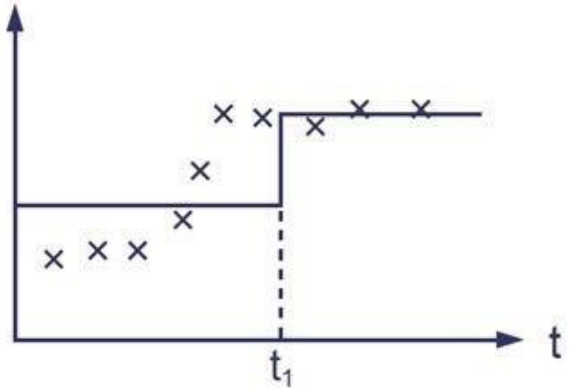
Consider the example of learning tree on a single variable t

Raw Data

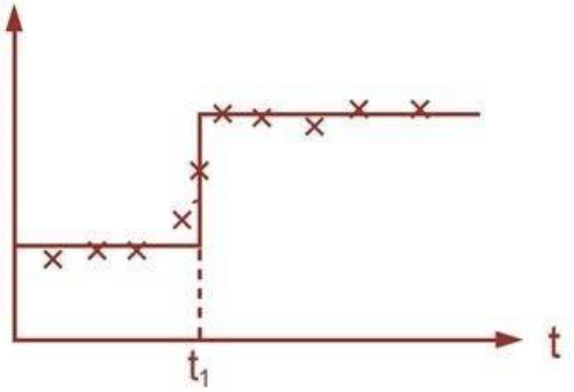




☒ Too many splits, $\Omega(f)$ is high



☒ Wrong split point, $L(f)$ is high



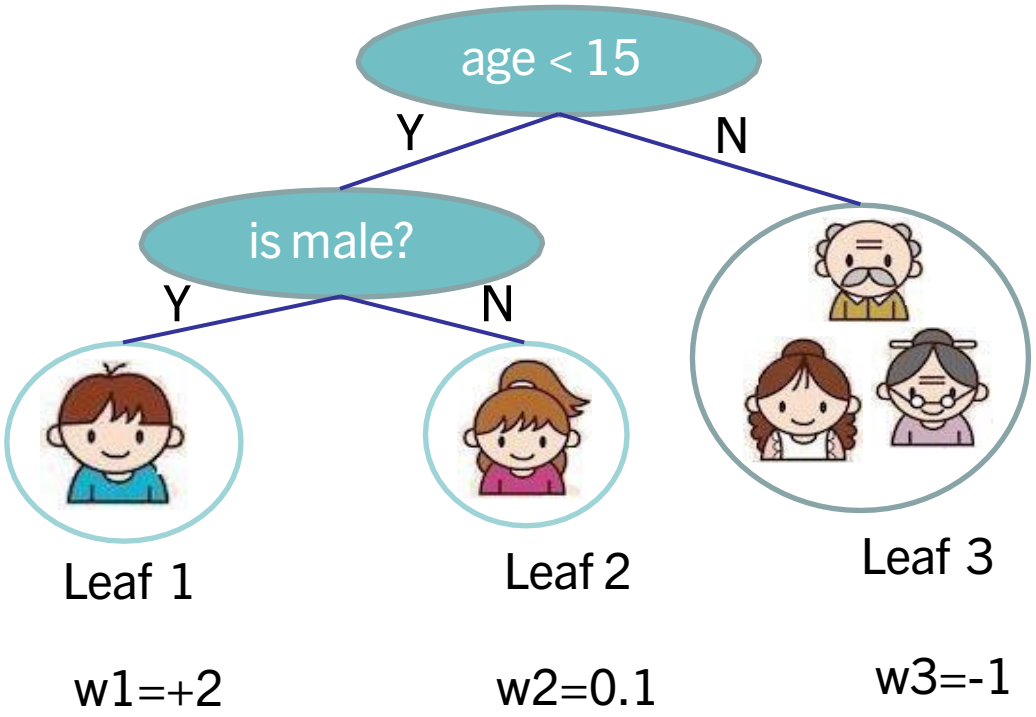
☑ Good balance of $\Omega(f)$ and $L(f)$

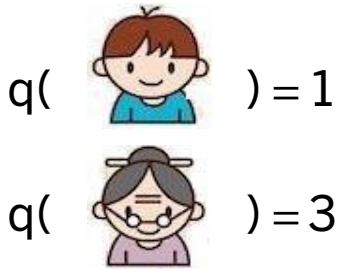
Define Complexity of a Tree

$$f_t(x) = w_{q(x)}, \quad w \in \mathbf{R}^T, q : \mathbf{R}^d \to \{1, 2, \cdots, T\}$$

The leaf weight of the tree

The structure of the tree



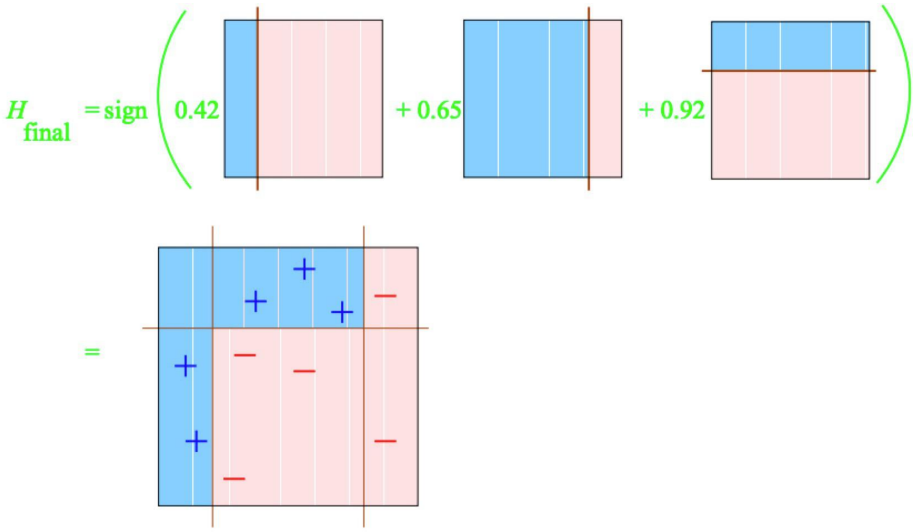


Final words on Boosting

Advantages

dmlc XGBoost eXtreme Gradient Boosting

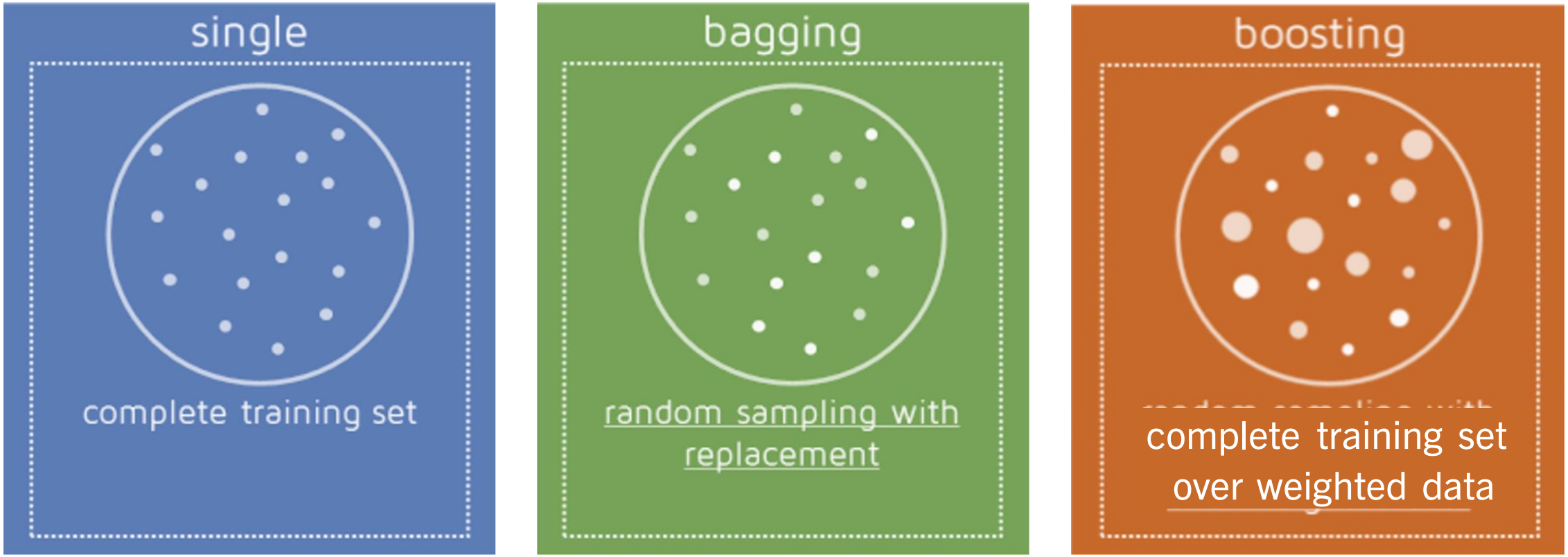
- Fast and simple
- Flexible: can work with any weak learner
- Handles various feature types
- Strong theoretical foundations



Caveats

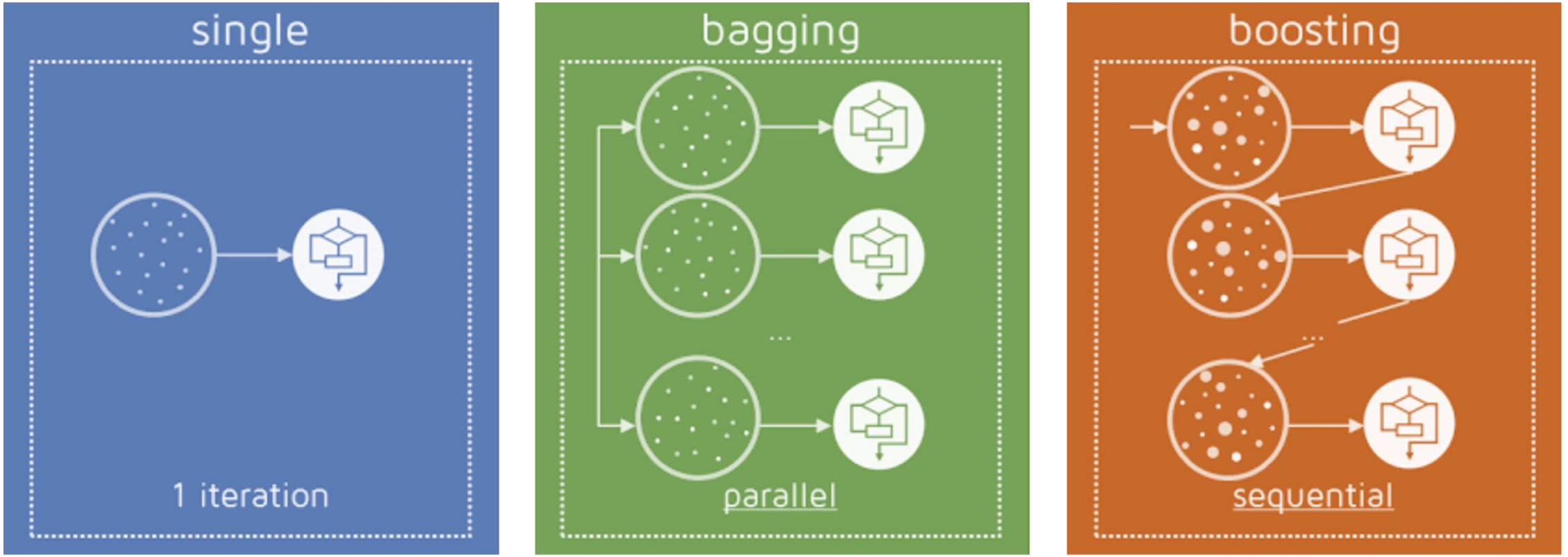
- Overfits if weak learner is too complex

Comparing Ensembles: Data



Based on https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Comparing Ensembles: Training



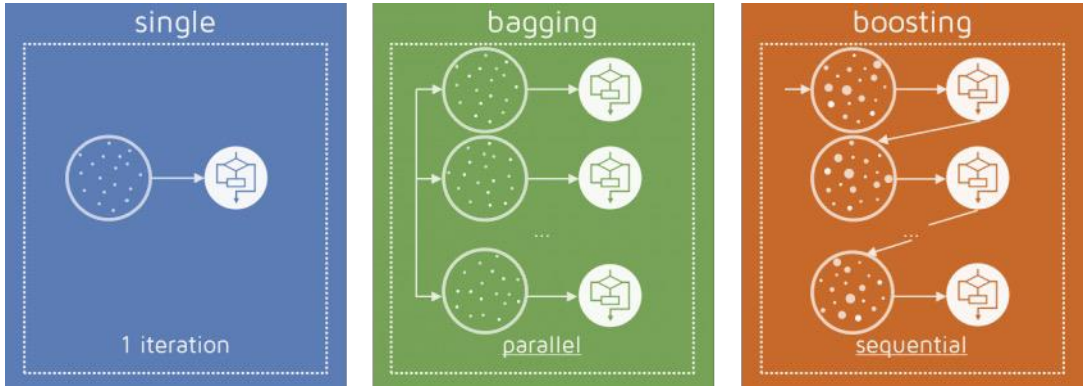
Based on https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

Boosting vs Bagging

- Bagging is typically faster, but may get a smaller error reduction (not by much).
- Bagging works well with “reasonable” classifiers.
- Boosting works with very simple classifiers.
  - E.g., Boostexter - text classification using decision stumps based on single words.
- Boosting may have a problem if a lot of the data is mislabeled, because it will focus on those examples a lot, leading to overfitting.

Slide from Edith Law at U Waterloo

Recap



- Ensemble methods are a must-try

- Bagging: average models trained on bootstrap samples
- Good bagging: Random Forests

- Boosting: sequence of models that correct for remaining mistakes
- Great boosting: xgboost