Data Science Lab - 4

Academic year 2019-2020

Lecturer Falco J. Bargagli-Stoffi

IMT School for Advanced Studies Lucca & KU Leuven

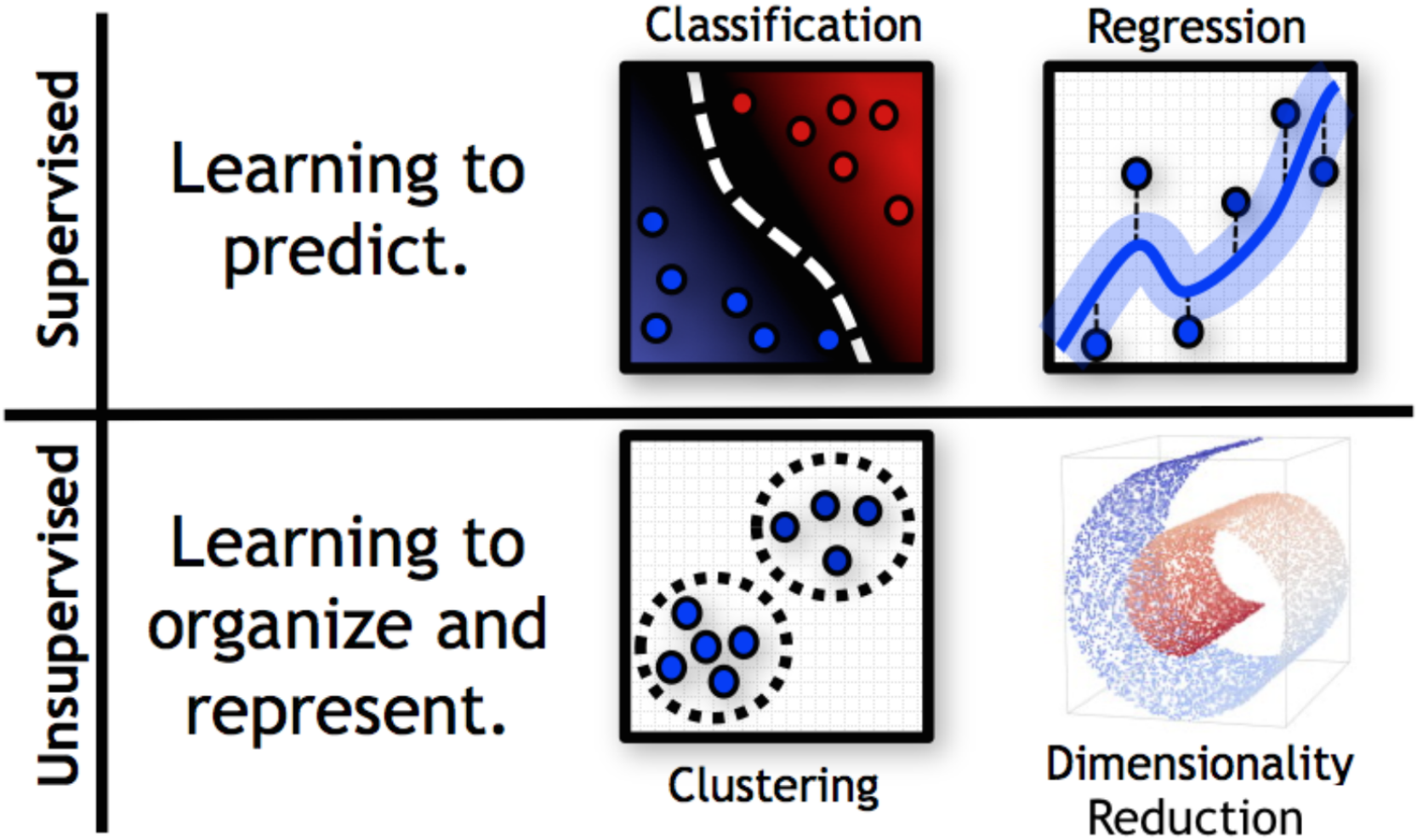
General Overview

ML is the sub-field of computer science that gives computers the ability to learn without being explicitly programmed, Arthur Samuel (1959)

- ML explores the study and construction of algorithms that can learn from and make predictions on data - such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions

- Main references in the Statistical Literature:
  1. Hastie, Tibshirani, Friedman *The Elements of Stat. Learning* (2001)
  2. Tibshirani, Hastie *An introduction to Statistical Learning* (2013)
  3. Efron, Hastie *Computer Age Statistical Inference* (2016)

Supervised vs Unsupervised ML



Overview

- Following Mullainathan and Speiss (2017 JEP) four main branches of applications:
  1. ML for causal inference (SL)
  2. ML for policy prediction (SL)
  3. ML to test theory (SL)
  4. ML for creation of new data sources (mostly UL)
- The focus will be on (1), (2)
- A brief overview on the some packages and functionalities for ML in `R` will be provided

Introduction

- **Supervised ML** algorithms are explicitly built for $\hat{y}$ (rather than the more familiar econometric compartment of $\hat{\beta}$ estimation)
- Picking a good **prediction function** is usually done in two steps:
  1. Pick the best **in-sample loss-minimizing function**

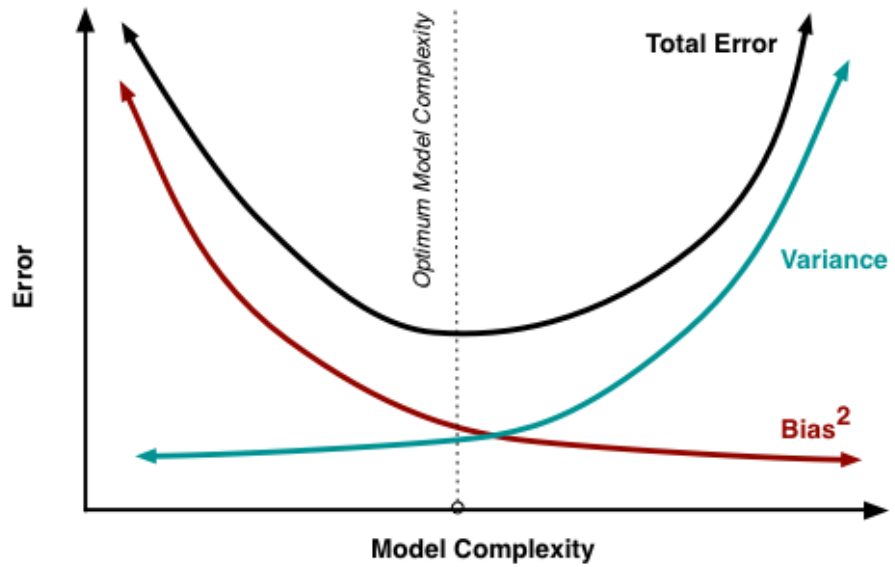
$$argmin \sum_{i=1}^{N} L(f(x_i), y_i) \quad over \quad f \in F \qquad s.t. \qquad R(f) \leq c$$

  2. Estimate the **optimal level of complexity** using **empirical tuning**

Why is ML tailored for prediction?

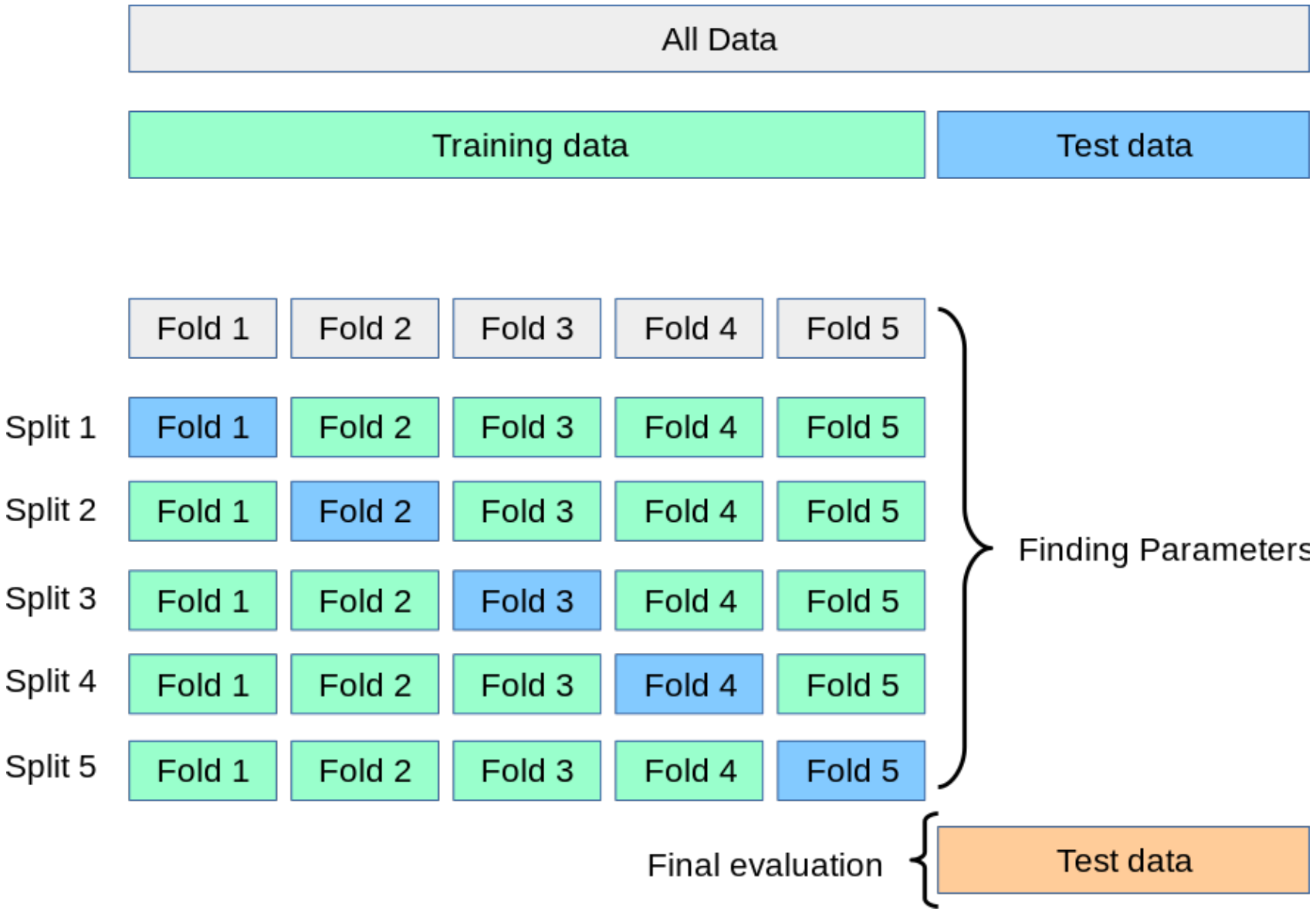
- Take a generic loss function such as the MSE of prediction:

$$E_{\mathcal{D}}[(y - \hat{f}(x))^2] = \underbrace{E_{\mathcal{D}}\left[\left(E_{\mathcal{D}}[\hat{y_0}] - \hat{f}(x)\right)^2\right]}_{\text{Variance}} + \underbrace{\left(E_{\mathcal{D}}[\hat{y_0}] - y\right)^2}_{\text{Bias}^2}$$



- By fixing the bias to be zero, the OLS regression rules out the possibility of this trade-off
- Impossibly to tune in a data-driven way the model with unbiased methods

How is model complexity chosen?



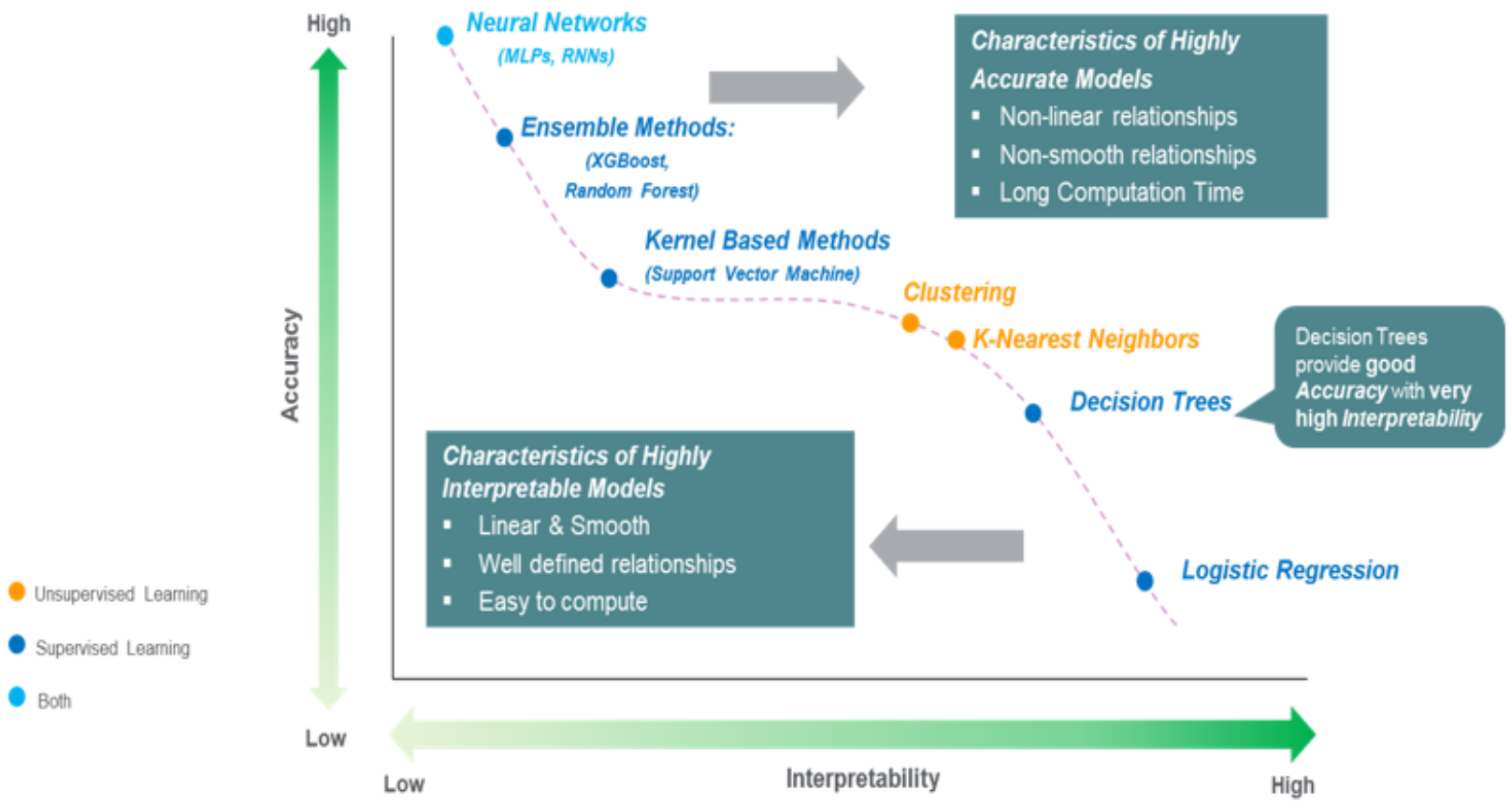
Accuracy measures for discrete outcomes

- Imagine the following scenario: 82 positive outcomes (e.g. high financial literacy score) and 18 negative outcomes (e.g. low financial literacy)

		Observed Outcome		
		Positive	**Negative**	
Predicted Outcome	**Positive**	80 (True positive)	17 (False positive)	Positive predicted value (PPV, Precision): 80/97= 82.5%
	Negative	2 (False negative)	1 (True negative)	Negative predicted value (NPV): 1/3= 33.3%
		True positive rate (TPR, Recall, Sensitivity) 80/82= 97.6%	True negative rate (TNR, Specificity): 1/18= 5.6%	Accuracy (ACC): 81/100= 81% Balanced Accuracy (BACC): (TPR+TNR)/2= 51.6%

$$F1 - score = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

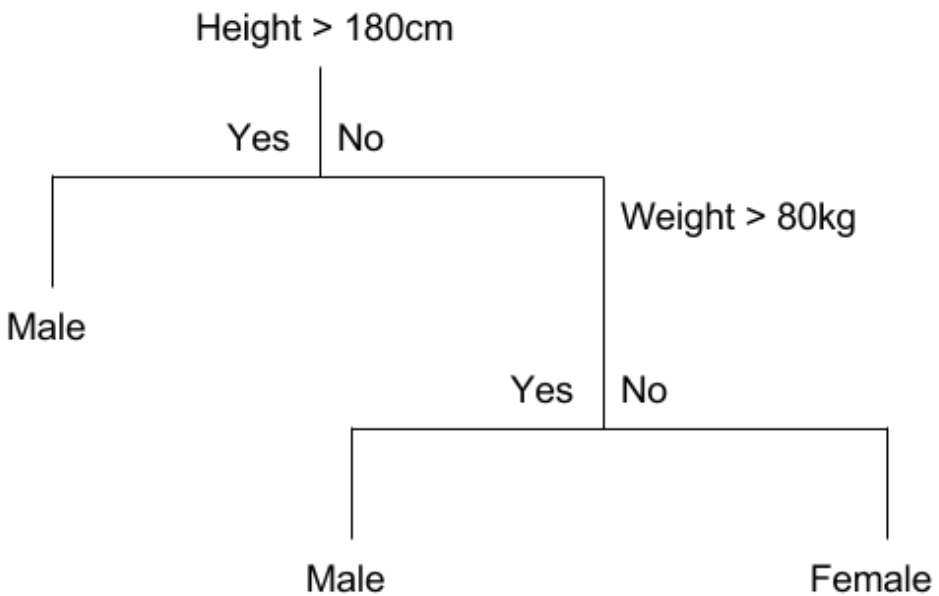
Interpretability vs Accuracy



Decision Trees

Definition 1 (Decision Tree)

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)



Classification and Regression Trees

Definition 2 (CART)

The CART methodology, introduced by Breiman, Friedman, Olshen and Stone in 1984 is an algorithm for construction of binary trees, or trees where each node is splitted in only two branches

- *Classification tree* analysis is when the predicted outcome is the class to which the data belongs
- *Regression tree* analysis is when the predicted outcome can be considered a real number

CART is the basis for other algorithms that generate more complex trees. It is divided into two phases:

1. Generation of the tree
2. Pruning of the tree

1. Generation of a Regression tree

Generation of a tree:

1. Splitting of the predictor space (set of possible values for $X_1, X_2, ..., X_p$) into J distinct and non-overlapping regions, $R_1, R_2, ..., R_J$
2. Predict Y conditional on realization of X_j in each region R_j using the sample mean in that region

The construction of the regions $R_1, R_2, ..., R_J$ (high-dimensional rectangles) proceeds by finding boxes $R_1, R_2, ..., R_J$ that minimize the MSE given by:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

where $-\hat{y}_{R_j}$ is the mean response for the training obs withing the j-th box

Binary splitting

1. Computationally infeasible to consider every possible partition of feature space

2. *Top-down* approach for the *recursive binary splitting*
   a. Select a predictor X_j and a cut point s s.t.:

$$R_1(j,s) = \{X|X_j \leq s\}\, and\, R_2(j,s) = \{X|X_j > s\}$$

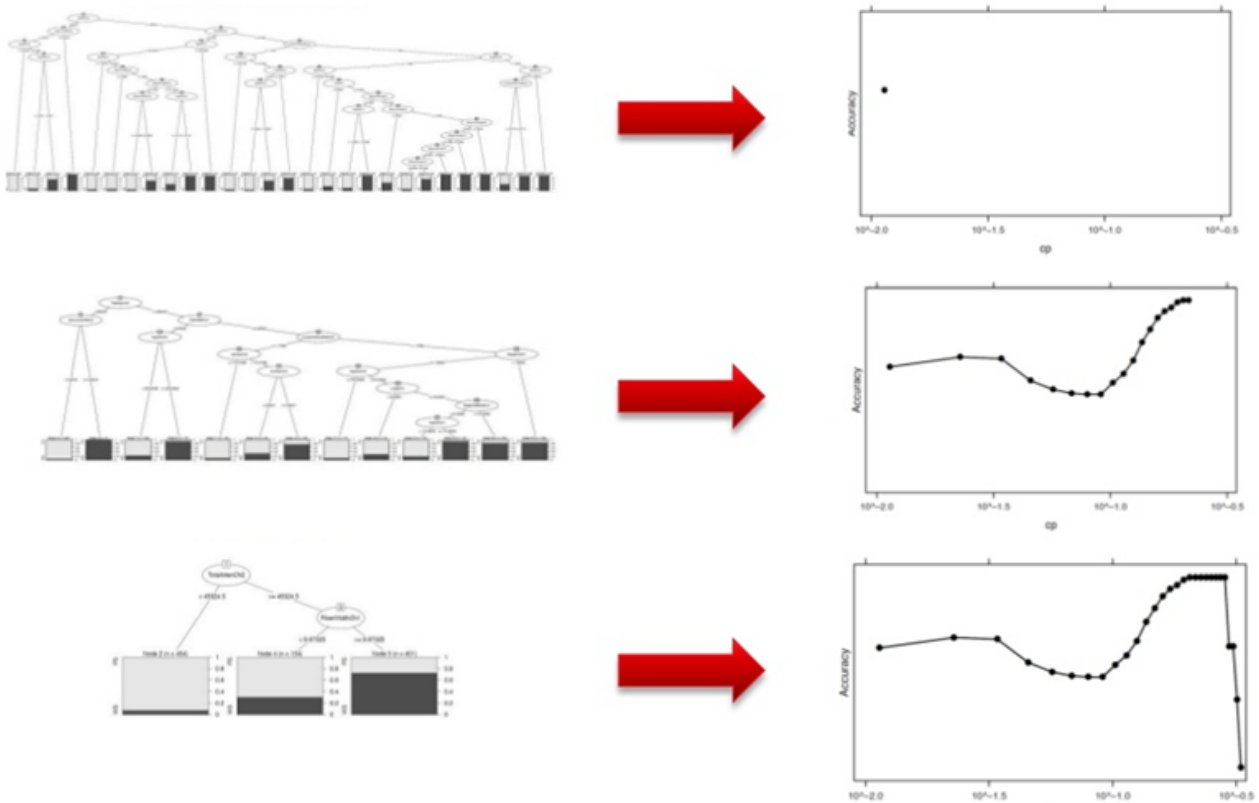
   minimizes:

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

   b. Repeat the process onto the two previously identified regions, so to minimize the MSE more
   c. Do it for all predictors and then choose the predictor and cut-point such that the resulting tree has the lowest MSE

2. Pruning of the tree (1)

- Too complex trees lead to data overfitting



2. Pruning of the tree (2)

- Two ways out:
  1. Split until the decrease in the MSE exceeds some threshold
  2. Grow a very large tree $\mathbb{T}$ and then prune it back to obtain a sub-tree

- This second strategy is implemented by minimizing:

$$\sum_{m=1}^{|\mathbb{T}|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |\mathbb{T}|$$

where $|\mathbb{T}|$ indicates the number of nodes of the tree $\mathbb{T}$, R_m is the rectangle corresponding to the m-th terminal node and α is a non-negative tuning parameter chosen by Cross-Validation

Classification Trees

- Classification Trees are similar to Regression Trees except they are used to predict a qualitative response
- The focus is not only on the class prediction corresponding to a particular terminal node region, but also in the class proportions among the training observations that fall into that region
- The main difference is that instead of minimizing the MSE it is used the Classification Error Rate

$$MSE \rightarrow CER$$

- CER is the fraction of training obs. in a region that do not belong to the most common class

$$CER = 1 - \max_k(\hat{p}_{m,k})$$

where $\hat{p}_{m,k}$ represents the proportion of training obs. in the m-th region that are from the k-th class

Impurity measure: Entropy and Information Gain

- Entropy and Information Gain
  - Definition: degree of disorder of our dataset Ω: if we define by F_1 and F_2 the fraction of observations Ω classified with "1" and "2", the entropy of the entire system S is defined as the following function $H(S)$:

$$H(S) = -F_1 log F_1 - F_2 log F_2$$

  - Respect to the J subclasses entropy is defined as:

$$H(S) = -\sum_{j=1}^{J} F_j log F_j$$

- The concept of information gain is a formalization of the entropic gain obtained through a partition of the data:

$$G = H(S) - H(S, s) \text{ where } H(S, s) = H'(S)$$

Pros and Cons of CART

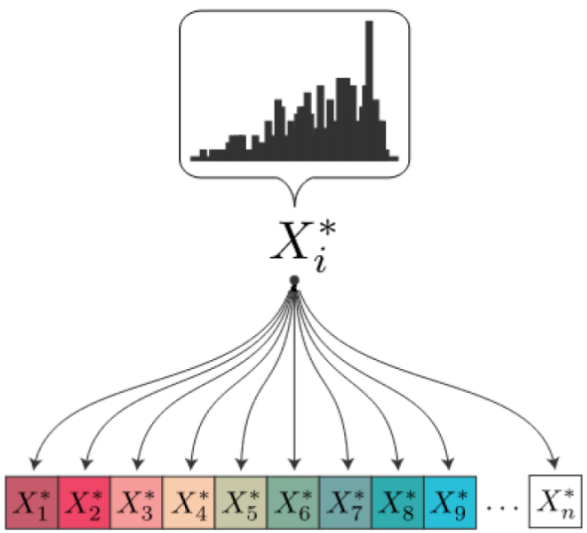
- Strengths and weaknesses of the CART methodology

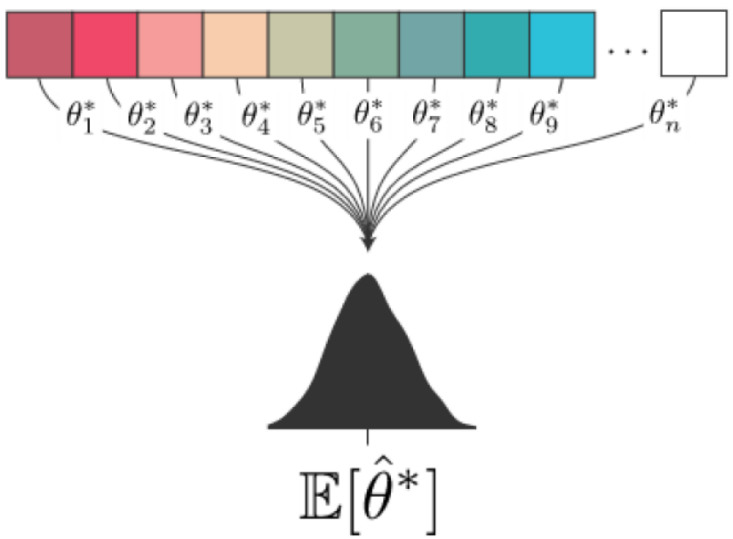
- Pros:
  1. CART results are invariant under monotone transformations of the independent variables;
  2. CART can use the data set with a complex structure have been developed to be able to detect the dominant structures of the data;
  3. CART are extremely robust to outliers;
  4. CART can use linear combinations of variables to make the split: no need to *discretize* continuous variables

- Cons:
  1. We don't use all the data (cross-validation);
  2. Every time our algorithm chooses a split, it chooses the best split in that exact moment (no bigger picture) $\rightarrow$ *greedy algorithm*

Random Forest

- RF: A Random Forest (Breiman, 2001) is a collection of fully grown CART. A Random Forest is a substantial transformation of the bagging method by introducing a collection of trees uncorrelated with each other.
  1. Bagging;
  2. Independence.





1. Bagging

- The techniques called bagging take shape **bootstrap** by the method. The term itself comes from $bootstrap\ aggregation$.
  1. Bradley Efron (1979)
  2. Sample **X** of dimension n
  3. Estimate the parameter θ by simulating **B** samples of the same abundance, obtained sampling by assuming **X** as if for the overpopulation of reference
- $\hat{f}(X) \mapsto B$ samples $X_1^*, ..., X_B^* \mapsto t(X_1^*), ..., t(X_B^*)$
- **Bagging estimator**

$$\hat{t}_{bag}(X) = \frac{1}{B} \sum_{b=1}^{B} t(X_b^*)$$

2. Independence

- If we develop the $bagged$ variance estimator $\hat{t}^B_{bag}(X)$:

$$\begin{aligned} Var(\hat{t}^B_{bag}(X)) &= Var\left(\frac{1}{B}\sum_{i=1}^{B} t(X_i)\right) \\ &= \sigma^2 \cdot \rho + \sigma^2 \frac{1-\rho}{B} \end{aligned}$$

.

- The idea behind the Random Forest is that we can significantly increase the benefits of $bagging$ through a reduction in the correlation of trees
- Random selection mechanism to select m variables between the p total splitting variables

Random Forest Algorithm for Regression and Classification

1. For b that goes from 1 to B:
   - Draw a sample $\mathbf{Z}^*$ of N units through the bootstrap method from our starting datasets Ω;
   - Grow a tree of the random forest T_b repeating, recursively, the following steps for each terminal node of the tree until you reach the minimum number of nodes n_{min}
     i. Select m randomly variables between the p available variables;
     ii. Choose the best combination of variables used for the split between the m variables;
     iii. Splitting the node into 2 children nodes.
2. Through the output of all the trees $\{T_b\}^B$ we can proceed as follows:
   - $Regression$: $\hat{t}_{rf}^B(X) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$;
   - $Classification$: we can think about prediction of the k-th class and the b-th tree of the random forest as $\hat{C}_{rf}^{B,K}(x)$. Where: $\hat{C}_{rf}^{B,K}(x) = majority\ vote\ \{\hat{C}_b(x)^{B,K}\}$

Pros and Cons of RF

- **Pros**:
  1. There is no need to rework or transform the data before building the model. Data must not be normalized and this approach is particularly robust to outliers;
  2. If we have a lot of input variables, we must not do any variable selection in a prior stage to construction of the model because it will be the same Random Forest to identify what are the most useful variables;
  3. Many trees are built through random mechanisms and therefore every tree is actually an independent model that does not bring the model to an overfitting.
- **Cons**:
  1. Strong data dependency
  2. Lower interpretability
  3. Higher computational costs
  4. Tendency to overfit

Bayesian Forests

- BART is Bayesian *sum-of-trees* model introduced by Chipman et al. (2010)
- BART accounts for potential shortcomings of RF (i.e., model overfitting)
- BART has shown excellent performance:
  1. In prediction settings: (Hern andez et al., 2018; Linero, 2018; Linero Yang, 2018; Murray, 2017; Starling et al., 2018)
  2. In causal inference: Hahn et al., 2017; Hill, 2011; Logan et al., 2019; Nethery et al., 2019, Bargagli-Stoffi et al., 2019)

Elements of Bayesian Inference

1. Sample space Y as the set of all the possible dataset where single dataset y consists in a subspace of the initial space

2. Θ is the set of all possible parameters from which we try to identify the parameter θ

3. the probability distribution of θ say $p(\theta)$ is the so called *prior distribution*

4. the probability distribution of y given θ, $p(y|\theta)$, represents the *sampling model*

5. the *posterior distribution* $p(\theta|y)$ represents the belief that θ will be our outcome given the observed dataset y.
   The Bayes' rule is a way to "connect" these elements:
$$P(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int_\theta p(y|\tilde{\theta})p(\tilde{\theta})d\tilde{\theta}}$$

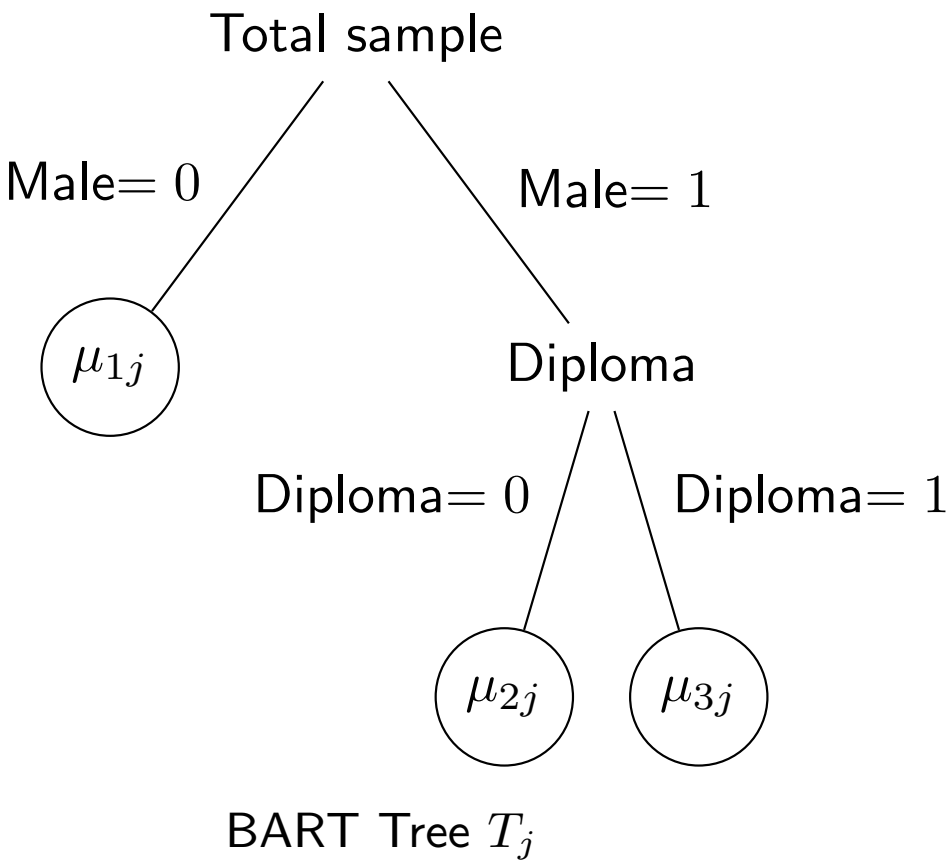
Bayesian Additive Regression Trees in a Nutshell (1)

- The BART model statement is:

$$\begin{aligned} y_i &= f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \\ f(x_i) &= \sum_{j=1}^{M} g(x, T_j, M_j) \end{aligned}$$

- BART prior is composed of priors on σ^2, terminal node values μ_{jl} and tree structures T_j
- The BART model is fit using an iterative MCMC model called *Bayesian backfitting*
- For μ_{lj} and T_j updates are straightforward since priors are conjugate

Bayesian Additive Regression Trees in a Nutshell (2)



- First prior on the probability that a node will split at depth k:

  $\beta(1+k)^{-\eta} \text{where} \beta \in (0,1), \eta \in [0,\infty)$

- Second prior on the probability distribution in the leaves:

  $$\mathcal{N}(0, \sigma_q^2) \text{ where } \sigma_q = \sigma_0/\sqrt{q},$$

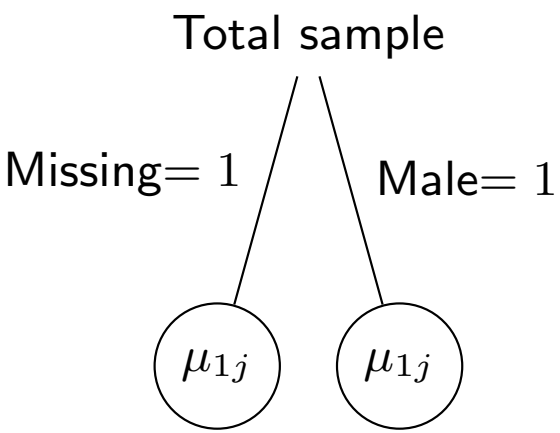
- Third prior on the error variance:

  $$\sigma^2 \sim Inv - Gamma(v/2, v\lambda/2)$$

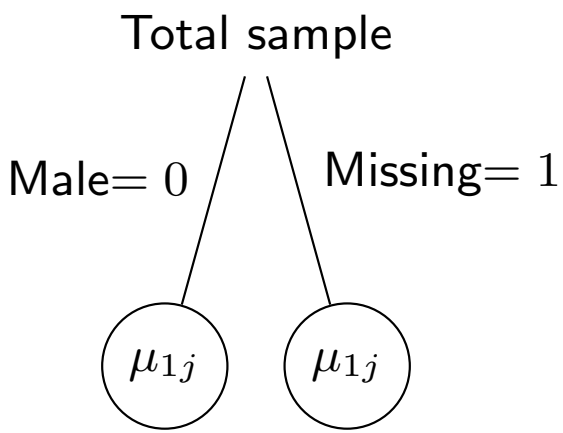
  where λ is chose to improve 90% of the times the RMSE of an OLS model

BART-MIA

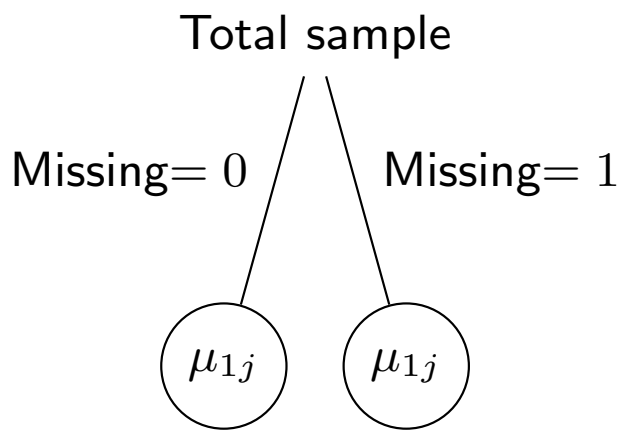
- To deal with missing values Kepelner and Bleich (2015) introduced a variation of BART to incorporate missing values in the splitting attributes



BART-MIA Tree T_1



BART-MIA Tree T_2



BART-MIA Tree T_3