Data Science Lab - 3

Academic year 2018-2019

Lecturer Falco J. Bargagli-Stoffi

IMT - School for Advanced Studies Lucca & KU Leuven

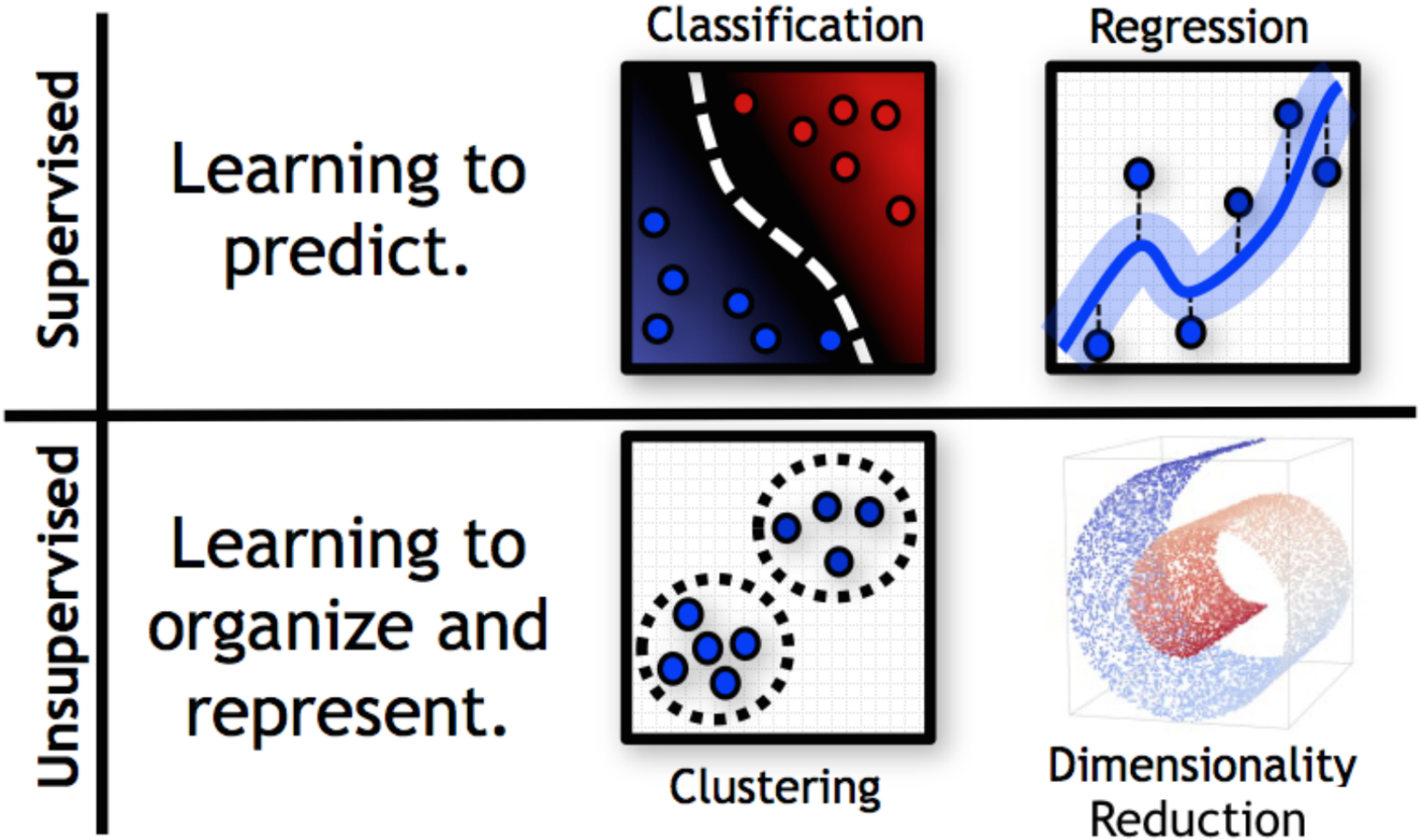
General Overview

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



Introduction

- A very good compendium of ML techniques from an applied econometric perspective is the recent paper by Mullainathan and Speiss (2017)
- The authors furnish a very useful insight:
  - **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**

Classes 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)

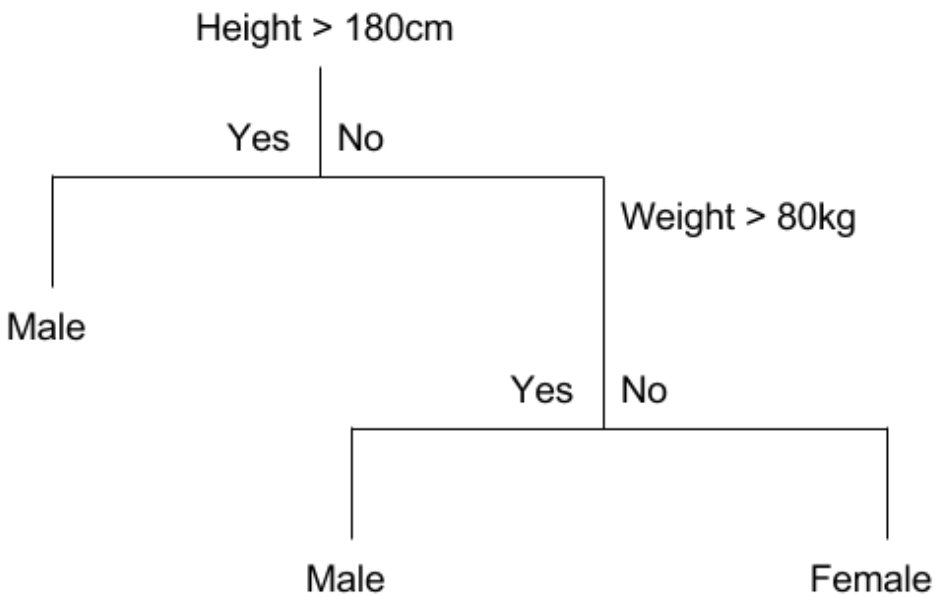
- The focus will be on (1), (2)

- A brief overview on the some packages and functionalities for ML in **R** will be provided

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.:

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

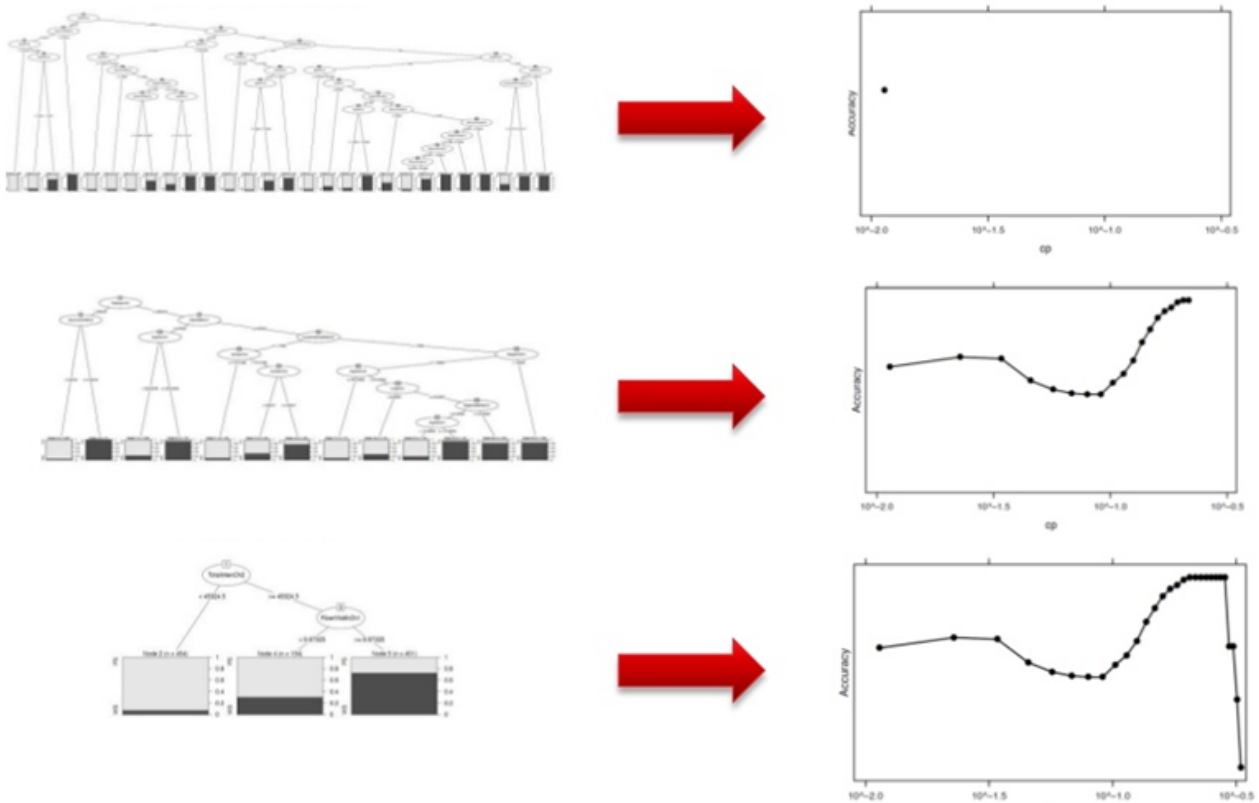
   minimizes:

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

   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

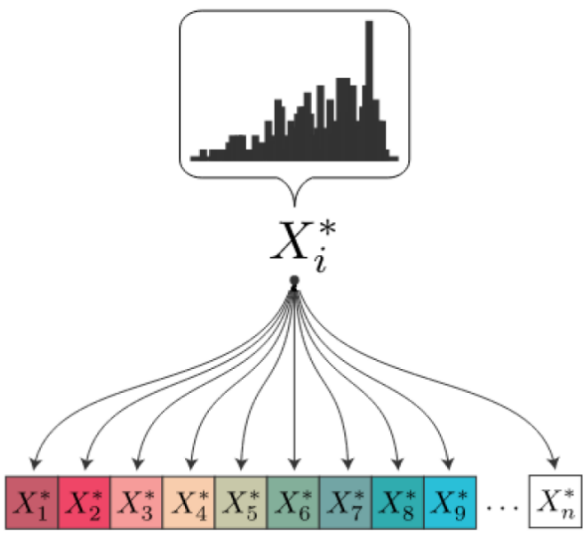
- Strengths and weaknesses of the CART methodology

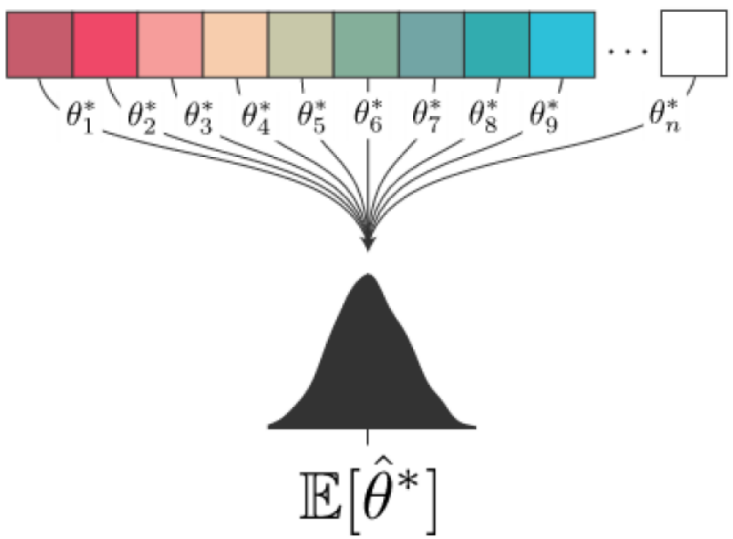
- Pros:
  1. CART results are invariant under monotone transformations of the independent variables;
  2. CART can use the data set with a complex structure have been developed to be able to detect the dominant structures of the data;
  3. CART are extremely robust to outliers;
  4. CART can use linear 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.





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