



Institut de Mathématiques de Toulouse, INSA Toulouse

Supervised Learning- Part II

Formation en machine Learning
EUR NanoX

Béatrice Laurent - Philippe Besse - Olivier Roustant

Methods studied in this course :

Part I

- Linear models for regression
- Linear models for classification

Part II

- Classification And Regression Trees, Bagging, Random Forests
- Neural networks, Introduction to deep learning

Outline

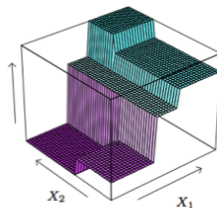
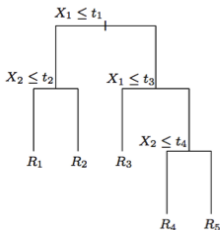
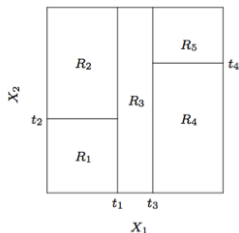
- Classification And Regression Trees (CART)
- Bagging, Random Forests

Classification And Regression Trees

Introduction

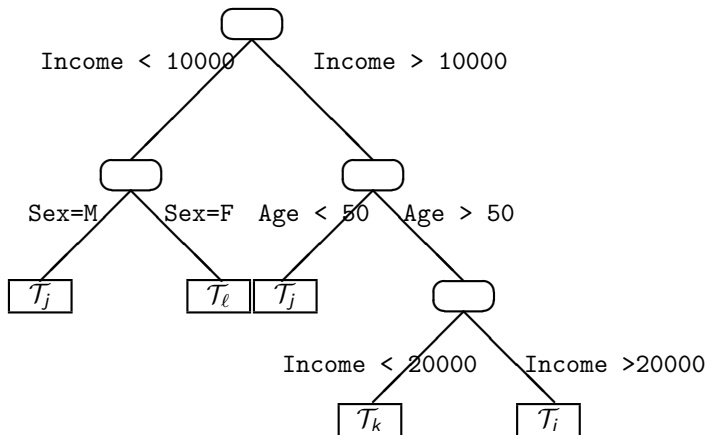
- Classification and regression trees (**CART**) : Breiman et al. (1984)
- X^j explanatory variables (quantitative or qualitative)
- Y qualitative with m levels $\{\mathcal{T}_\ell; \ell = 1 \dots, m\}$: **classification tree**
- Y quantitative : **regression tree**
- **Objective** : construction of a binary **decision tree** easy to interpret
- No assumption on the model : non parametric procedure.

Example of binary regression tree



Source : Hastie, Tibshirani, Friedman (2019), "The elements of statistical learning"

Example of binary classification tree



Principles for constructing a tree

- Recursive binary split
 - Split a region in two, then split subregions in two, then ...
- Splits are defined by one variable
 - Very easy numerically : d optimizations in 1-dimensions
- Clustering idea
 - Find a split that give the most homogeneous groups

Constructing regression trees

For a given region (node) κ with size $|\kappa|$, define the **heterogeneity** by :

$$D_{\kappa} = \sum_{i \in \kappa} (y_i - \bar{y}_{\kappa})^2 = |\kappa| \frac{1}{|\kappa|} \sum_{i \in \kappa} (y_i - \bar{y}_{\kappa})^2$$

Splitting procedure

For a variable x_j , and a split candidate t , define left and right subregions

$$\kappa_L(t, j) = \{x_j \leq t\}, \quad \kappa_R(t, j) = \{x_j > t\}.$$

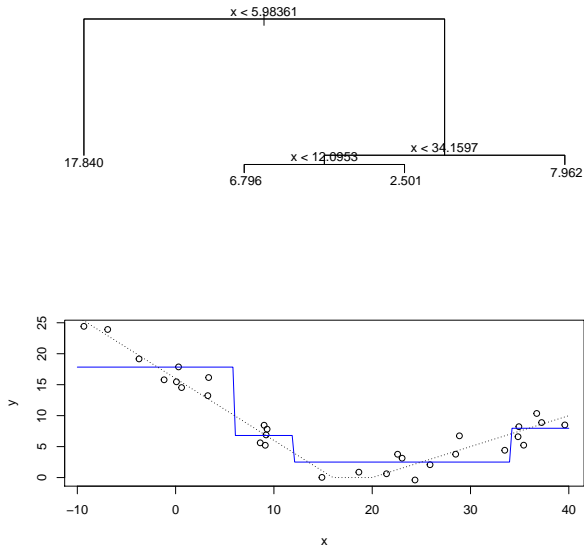
Find (j, t) in order **to minimize the intra-class variance**

$$J(j, t) = D_{\kappa_L(t, j)} + D_{\kappa_R(t, j)},$$

or equiv. **to maximize the decrease in heterogeneity (inter-class variance)**

$$D_{\kappa} - J(j, t)$$

Illustration in 1 dimension



Constructing classification trees

This is the same procedure, with specific notions of heterogeneity

Heterogeneity measures in classification

p_{κ}^{ℓ} : proportion of the class \mathcal{T}_{ℓ} of \mathcal{Y} in the node κ .

- Shannon Entropy

$$E_{\kappa} = - \sum_{\ell=1}^m p_{\kappa}^{\ell} \log(p_{\kappa}^{\ell}) \quad \Rightarrow \quad D_{\kappa} = -|\kappa| \sum_{\ell=1}^m p_{\kappa}^{\ell} \log(p_{\kappa}^{\ell})$$

Maximal in $(\frac{1}{m}, \dots, \frac{1}{m})$, minimal in $(1, 0, \dots, 0), \dots, (0, \dots, 0, 1)$
(by continuity, we assume that $0 \log(0) = 0$)

- Gini concentration : $D_{\kappa} = |\kappa| \sum_{\ell=1}^m p_{\kappa}^{\ell} (1 - p_{\kappa}^{\ell})$

Illustration with two classes ($m = 2$)

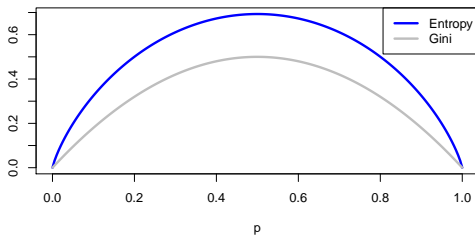
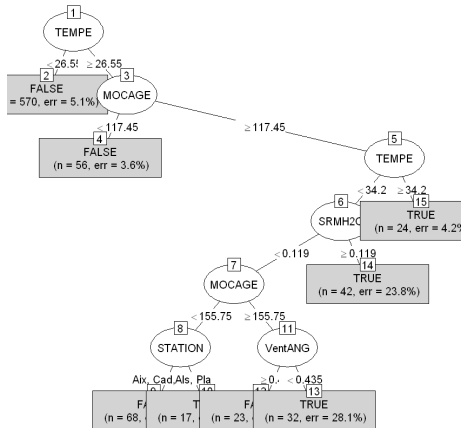


FIGURE – Heterogeneity criteria for classification. Both are minimal for $p = 0$ or $p = 1$, and maximal for $p = 1/2$.

Example for Ozone data



Ozone : Classification tree pruned by cross-validation

Stopping rule, pruning, optimal tree

- We need a tradeoff between maximal tree (overfits) and the constant tree (too rough)
- There exists a nice theory to find an optimal tree, minimizing prediction error penalized by complexity (number of leaves)
- When aggregating trees (random forest), simpler procedures are often preferred (see why after), e.g. fixing the number of leaves

Advantages

- **Trees** are easy to interpret
- **Efficient algorithms** to find the pruned trees
- Tolerant to **missing data**

⇒ Success of CART for practical applications

Warnings

- **Variable selection** : the selected tree only depends on few explanatory variables, trees are often (wrongly) interpreted as a variable selection procedure
- **High instability** of the trees : not robust to the learning sample, curse of dimensionality ..
- **Prediction accuracy** of a tree is often poor compared to other procedures

⇒ **Aggregation of trees : bagging, random forests**

Outline

- Classification And Regression Trees (CART)
- Bagging , Random Forests

Introduction

- Combination or **aggregation** of models (almost) without **overfitting**
- **Bagging** is for **bootstrap^(*) aggregating** : Breiman, 1996
- **Random forests** : Breiman, 2001
- Allows to aggregate any modelisation method
- **Efficient** methods : Fernandez-Delgado et al. (2014), *Kaggle*

(*) *bootstrap = sampling with replacement*

- **Bagging** is appropriate for unstable algorithms, with small bias and high variance (CART)

Bagging - Principle

Bootstrap AGGREGatING

- **Variance reduction** : by aggregating independent predictions
 - Aggregation : average (regression), majority vote (classification)
- **Bootstrap trick** : get new data from themselves by resampling!
 - Caution : new data remain (slightly) dependent on the initial ones

Bagging - Introductory example

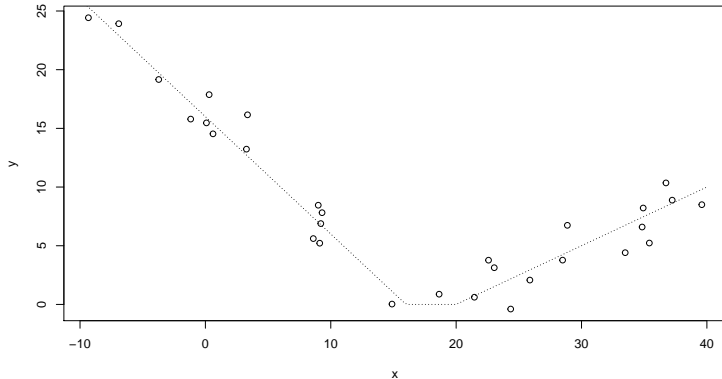


FIGURE – Original data

Bagging - Introductory example

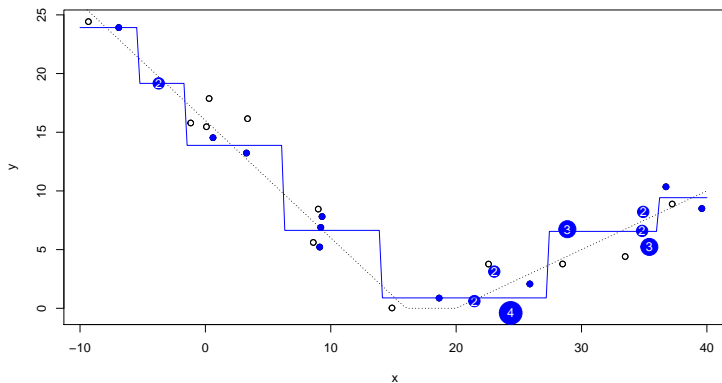


FIGURE – Bootstrap sample $n^{\circ}1$ (in blue), and corresp. prediction with tree. The point size is proportional to the number of replicates.

Bagging - Introductory example

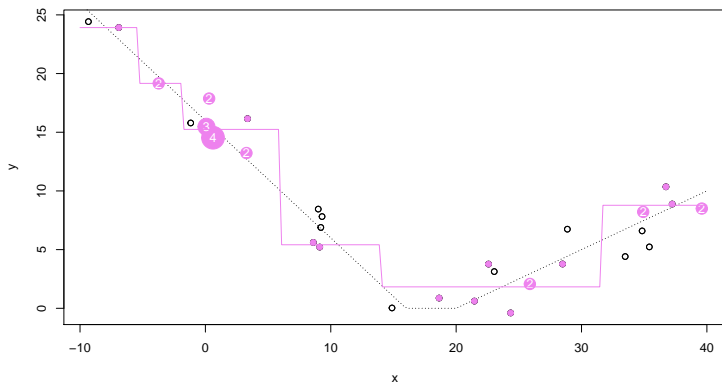


FIGURE – Bootstrap sample $n^{\circ}2$ (in violet), and corresp. prediction with tree. The point size is proportional to the number of replicates.

Bagging - Introductory example

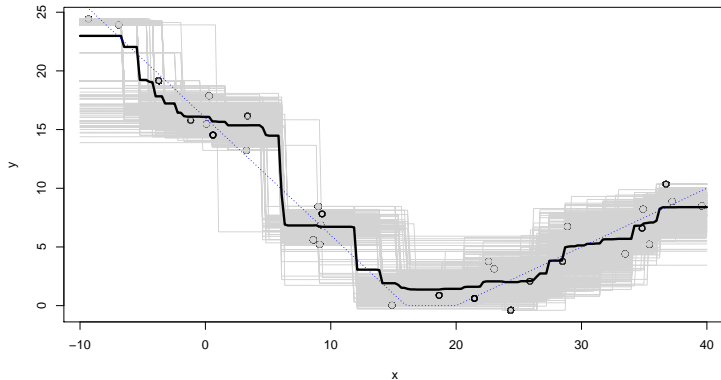


FIGURE – 500 bootstrap samples (grey), corresp. predictions with tree, and their average (bold line).

Bagging - Pause

Physical experiment !

Experiment yourself the bootstrap procedure by resampling “by hand”

Question : Choose a number between 1 and N (number of participants). What is the probability that your number does not appear in the bootstrap sample ?

Bagging - Out-Of-Bag data

Out-Of-Bag (OOB) data

For each bootstrap sample :

- Let U_1^*, \dots, U_N^* be random variables representing the bootstrapped indices. The probability that a given data z_i is not chosen is :

$$\mathbb{P}\left(z_{U_1^*} \neq z_i, \dots, z_{U_N^*} \neq z_i\right) = \left(1 - \frac{1}{N}\right)^N \xrightarrow{N \rightarrow +\infty} e^{-1} \approx 0.367$$

- The non-chosen data are called **Out-Of-Bag (OOB)**. They can be used **as a test set inside the bootstrap loop**

The **OOB error** is obtained by averaging prediction errors over OOB data

Bagging - Out-Of-Bag data

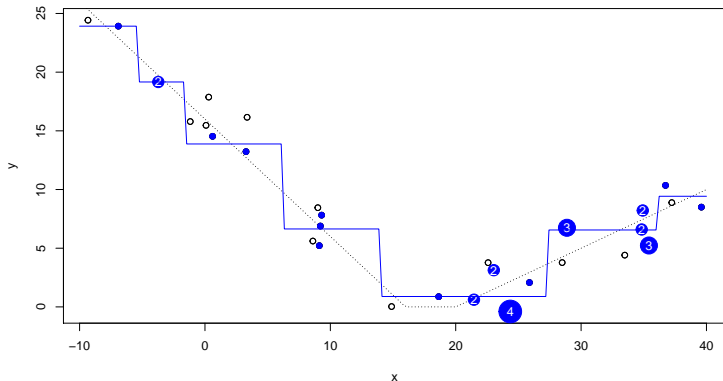


FIGURE – Residuals for the OOB bootstrap sample n^o1 (red bars).

Bagging - Out-Of-Bag data

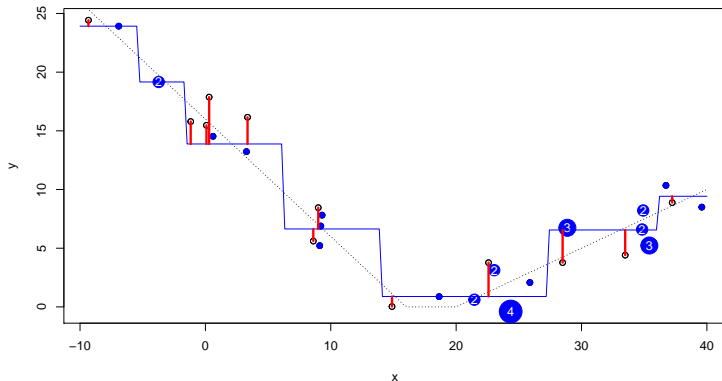


FIGURE – Residuals for the OOB bootstrap sample n^o1 (red bars).

Bagging - Out-Of-Bag data

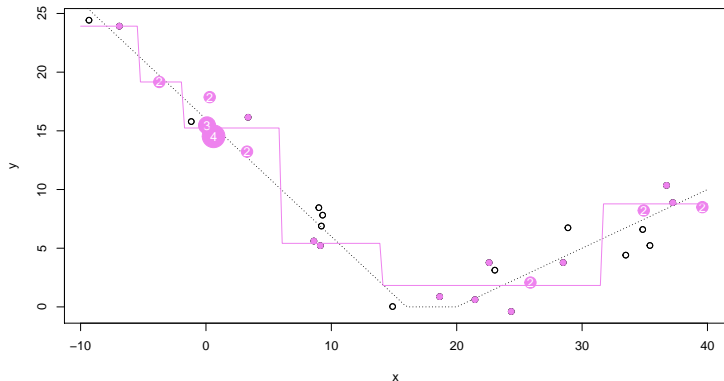


FIGURE – Residuals for the OOB bootstrap sample n^{o2} (red bars).

Bagging - Out-Of-Bag data

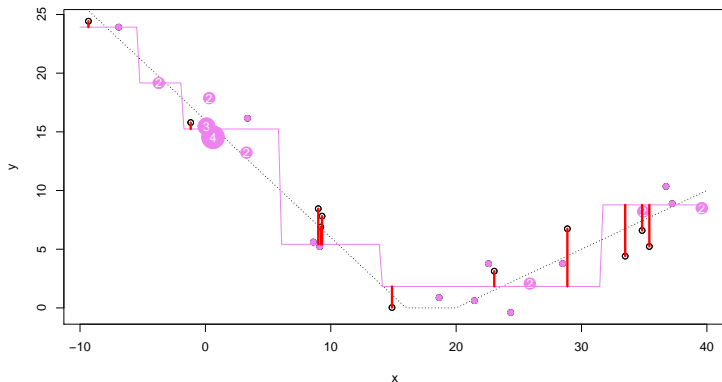


FIGURE – Residuals for the OOB bootstrap sample n^o2 (red bars).

Bagging - Theory

Framework and notations

- Output : Y , a quantitative or qualitative variable to explain
- Inputs : X^1, \dots, X^p , explanatory variables
- Model : $f(\mathbf{x})$, function of $\mathbf{x} = \{x^1, \dots, x^p\} \in \mathbb{R}^p$
- Learning sample : $\mathbf{z} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, with distribution F
- A predictor : $\hat{f}_{\mathbf{z}}$, associated to \mathbf{z} , with $f(.) = \mathbb{E}_F(\hat{f}_{\mathbf{z}})$
- Bootstrap samples : $\{\mathbf{z}_b\}_{b=1, B}$
- Aggregated predictor :
 - Y quantitative : $\hat{f}_B(.) = \frac{1}{B} \sum_{b=1}^B \hat{f}_{\mathbf{z}_b}(.)$ (mean)
 - Y qualitative : $\hat{f}_B(.) = \arg \max_j \text{card} \left\{ b \mid \hat{f}_{\mathbf{z}_b}(.) = j \right\}$ (majority vote)

Bagging - Theory

Variance reduction quantification

- The B bootstrap samples are built on the same learning sample \mathbf{z}
 \Rightarrow the estimators $\hat{f}_{z_b}(\mathbf{x}_0)$ are **not independent**
- Regression case : If $\text{Corr}(\hat{f}_{z_b}(\mathbf{x}_0), \hat{f}_{z_{b'}}(\mathbf{x}_0)) = \rho(\mathbf{x}_0)$,

$$\begin{aligned} E(\hat{f}_B(\mathbf{x}_0)) &= f(\mathbf{x}_0) \\ \text{Var}(\hat{f}_B(\mathbf{x}_0)) &= \rho(\mathbf{x}_0)\text{Var}(\hat{f}_b(\mathbf{x}_0)) + \underbrace{\frac{(1 - \rho(\mathbf{x}_0))}{B}\text{Var}(\hat{f}_b(\mathbf{x}_0))}_{\rightarrow 0 \text{ as } B \rightarrow \infty} \end{aligned}$$

- Importance to find **low correlated predictors** $(\hat{f}_b(\mathbf{x}_0))_{1 \leq b \leq B}$.
 \Rightarrow **Random forests**

Random forest - Principle

The three ingredients of random forest

- **Variance reduction** : by aggregating independent predictions
 - Aggregation : average (regression), majority vote (classification)
- **Data resampling** : get new data from themselves by resampling!
 - Caution : new data remain (slightly) dependent on the initial ones
- **Variable resampling** : reduces correlation between resampled data
 - The number of resampled variables must be tuned properly

Random forest = $\underbrace{\text{data resampling} + \text{aggregation}}_{\text{bagging}} + \text{variable resampling}$

Random forest

Algorithm

- Let \mathbf{x}_0 the point where we want to predict, \mathbf{z} a learning sample
- For $b = 1$ to B , do :
 - Generate a bootstrap sample \mathbf{z}_b^*
 - Estimate a tree with randomization of the variables :
At each node, resample $m < p$ variables to build the subdivision
- Aggregate predictors (average or majority vote)

Random forest

Variance reduction quantification

Consider the regression case. For a large number of bootstrap samples,

$$\text{Var} \left(\hat{f}_B(\mathbf{x}_0) \right) \approx \underbrace{\rho(\mathbf{x}_0)}_{\text{small when } m \text{ small}} \times \underbrace{\text{Var} \left(\hat{f}_b(\mathbf{x}_0) \right)}_{\text{small when } m \text{ large}}$$

⇒ Tradeoff required to choose m !

Random forest

Random forest : utilisation

- **Pruning** : tree with q leaves, or complete tree,
 - Reducing variance by computing the optimal tree is time-consuming
- **Random selection** of m predictors : default values
 - $m = \frac{p}{3}$ for regression
 - $m = \sqrt{p}$ for classification
- Choice of tuning parameters (including m) by cross-validation

Interpretation - Variable importance

How can we quantify the importance of a variable X_i in random forest ?

Decrease in heterogeneity

Average the decrease of heterogeneity when X_i is chosen as a split.

- Mean Decrease Accuracy
- Mean Decrease Gini

Permutation of variables

Compute the OOB error for the subsample of OOB data involving X_i . Compare with the OOB error when permuting at random the inputs (but keeping the output).

Random forest

To go further

- Prediction intervals with ranger
- Anomaly detection with IsolationForest
- Imputation of missing data with missForest
- Survival analysis with survival forest
- ...

Example on ozone data

```
> library(randomForest)
> rf.reg <- randomForest(O3obs ~, data = datappr, xtest = datestr[, -2],
  ytest = datestr[, "O3obs"], ntree = 500, do.trace = 50, importance = TRUE)
```

Tree	Out -of	- bag	Test	set
	MSE	%Var(y)	MSE	%Var(y)
50	697.9	40.77	568.5	36.75
100	689.5	40.28	555.9	35.93
150	683.8	39.95	563.2	36.41
200	685.4	40.04	561	36.27
250	678.2	39.62	564.2	36.47
300	675.1	39.44	569.2	36.79
350	676.8	39.54	572.8	37.02
400	674.3	39.39	571.4	36.93
450	673.9	39.37	571.5	36.94
500	674.3	39.39	569.6	36.82

```
> round(importance(rf.reg), 2)
```

	%IncMSE	IncNodePurity
JOUR	1.98	11011.79
MOCAGE	41.46	388657.27
TEMPE	51.73	409018.57
STATION	21.73	75350.42
VentMOD	12.95	91387.20
VentANG	18.81	124908.37
SRMH2O	16.76	114463.05
LN02	7.73	84152.34
LNO	10.04	74387.32

Details (from R help file of function importance)

“The first measure [%IncMSE] is computed from permuting OOB data : For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences.

The second measure [IncNodePurity] is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares.”

rf.reg

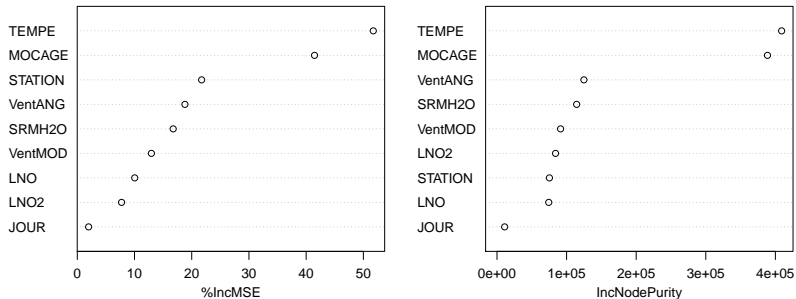


FIGURE – Variable importance plot, returned by the R function `importance`

References

- L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen (1984). *Classification and regression trees*. Chapman et Hall. CRC Press, Boca Raton.
- Giraud C. (2015) *Introduction to High-Dimensional Statistics* Vol. 139 of Monographs on Statistics and Applied Probability. CRC Press, Boca Raton, FL.
- Hastie, T. and Tibshirani, R. and Friedman, J, (2009), *The elements of statistical learning : data mining, inference, and prediction*, Springer.