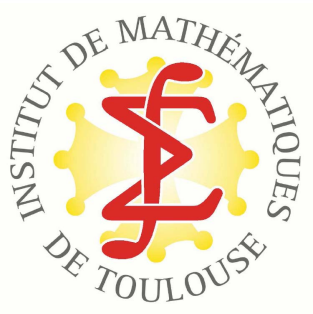


Institut de Mathématiques de Toulouse, INSA Toulouse

Supervised Learning- Part I
Linear models for Regression and Classification

ML Training for Data Science
October 2024

Béatrice Laurent - Mélisande Albert - Olivier Roustant -
Sébastien Gerchinovitz

Introduction

In the framework of **Supervised learning**, we have a **Learning sample** composed with observation data of the type **input/output** :

$$d_1^n = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

with $\boldsymbol{x}_i \in \mathbb{R}^p$, $y_i \in \mathcal{Y}$ for $i = 1 \ldots n$.

Objectives : From the learning sample, we want to

- **Estimate** the link between the input vector $\boldsymbol{x}$ (explanatory variables) and the output y (variable to explain) :

$$y = f(x^1, x^2, \ldots, x^p)$$

- **Predict** the output y associated to a new entry $\boldsymbol{x}$,
- **Select** the important explanatory variables among $x^1, \ldots, x^p$.

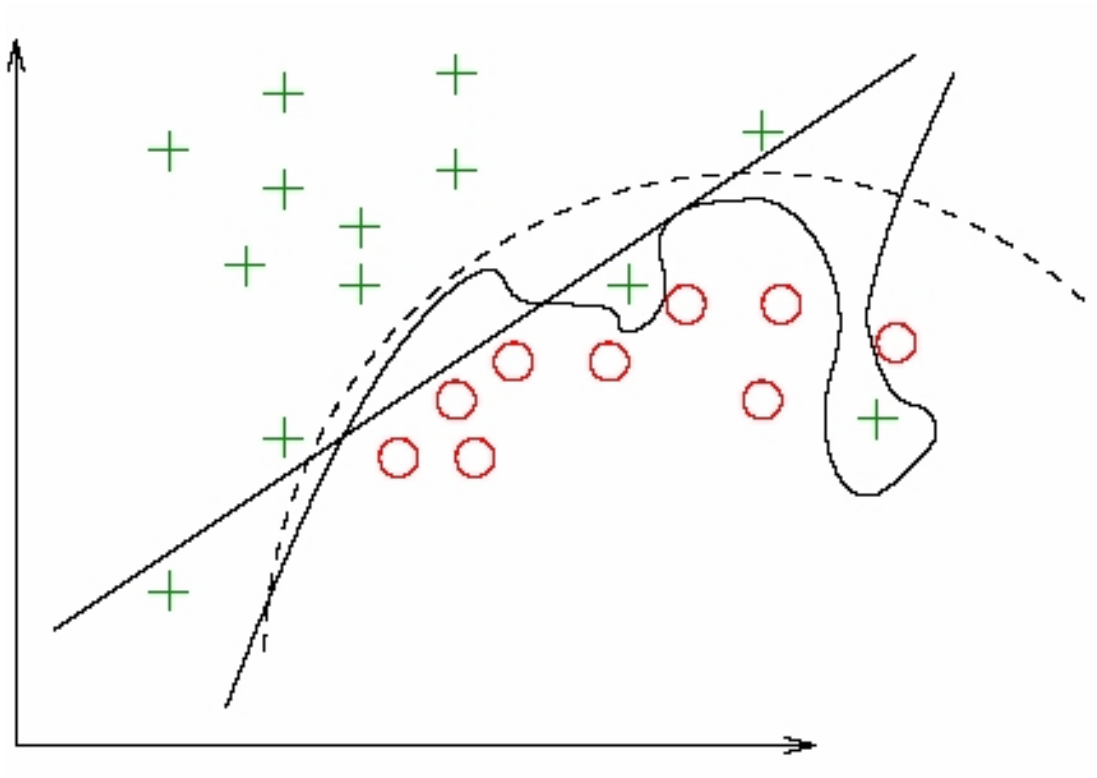
Introduction

quantitative output	qualitative output
$\mathcal{Y} \subset \mathbb{R}$	$\mathcal{Y}$ finite
$\downarrow$	$\downarrow$
real regression	**classification**

The explanatory variables $X^1, \dots X^p$ can be qualitatives or quantitatives

Choice of the model

- Importance of the principle of parcimony : "it is necessary to determine a model that provides an adequate representation of the data, with as few parameters as possible".
- Bias-variance trade-off



Supervised Classification : *Complexity of the models*

First step : *Data munging*

1. Extraction with or without survey
2. Exploration, visualization
3. Cleaning, transformation of the data, computation of new variables (*features*)
4. Management of missing data

Second step : *Learning*

1. Random Partition of the sample : learning, (validation), test
2. **For** each method that we consider :
   - Learning (estimation) depending on λ (tuning parameter)
   - Optimization of λ : validation set or cross-validation with the learning set
3. Comparison of the methods : prediction error on the test sample
4. Eventual Iteration (*Monte Carlo*)
5. Choice of the method (prevision *vs.* interpretability).
6. Estimation of the selected model with all the sample, exploitation

Possibly : Aggregation of several models

Question : Where to bring the effort ?

- *Data munging*
- Selection of the methods to compare
- Optimization of the parameters
- Optimal Combination of the models

Depending on :

- Goal, allotted time, computing resources
- Complexity of the underlying problem
- Structure and properties of the data

Usecase Ozone

Aim : Prediction of the ozone concentration for the next day at 5 PM (max. of the day) from a learning sample composed of the explanatory variables $X^1, \dots X^p$:

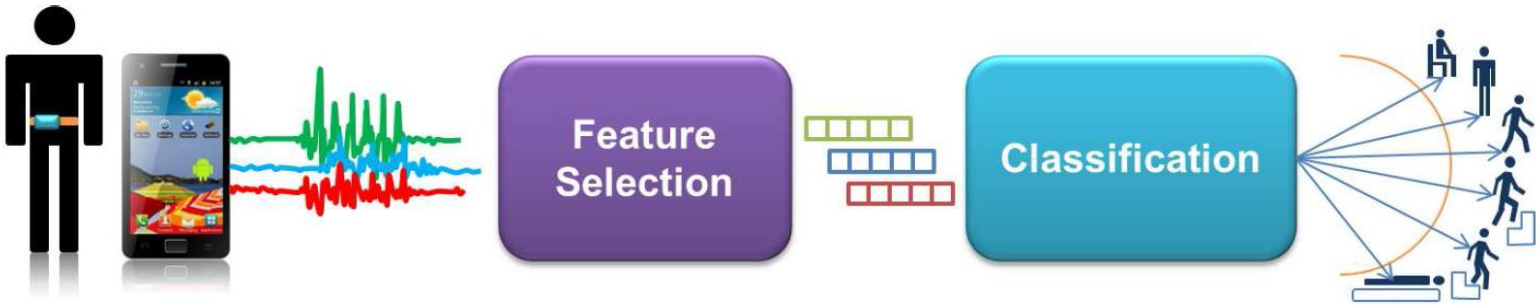
- MOCAGE (deterministic model of Meteo France),
- NO, NO_2,
- H_2O,
- Temperature,
- Wind speed and orientation
- Station,
- Type of day (holiday or not)

and the variable to explain :

- Y :Ozone concentration

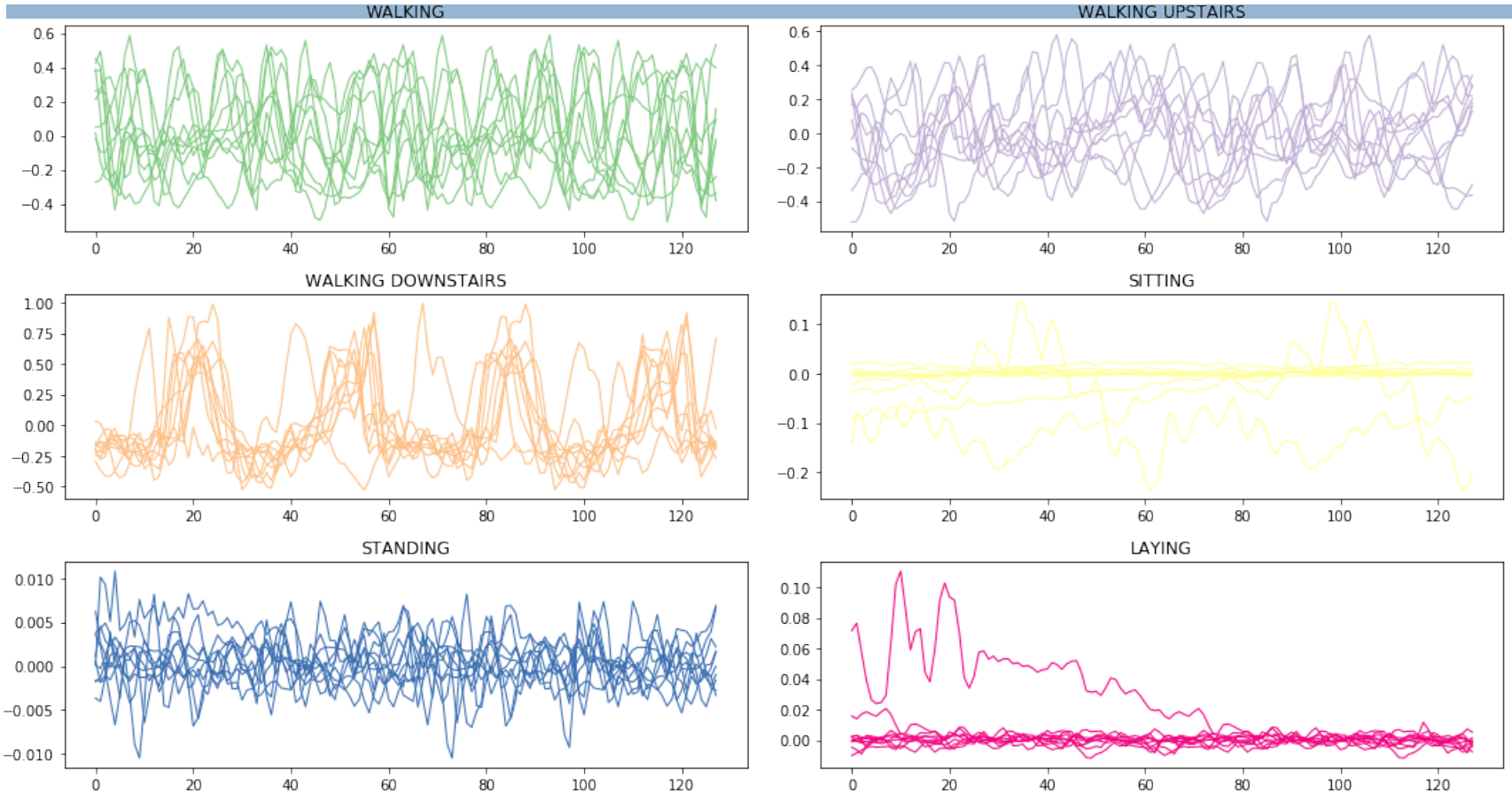
$\hookrightarrow$: **Statistical adaptation**

Usecase HAR



Human activity recognition HAR

- Public data available on *UCI repository*
- 9 signals per individual : The accelerations in x, y and z, those by subtracting the natural gravity and the angular accelerations in x, y, and z obtained from the gyroscope.
- Each signal contains $p = 128$ measures sampled at 64 htz during 2s.
- 7352 samples for learning and 2947 for testing.
- Objectives : Activity recognition (6 classes) standing, sitting, lying, walking, walking upstairs or walking downstairs.



Human activity recognition : acceleration in y by class

HAR First step : "features" variables obtained from signal processing

- $p = 561$ new variables (*features*)
  - Time domain : min, max, means, variances, correlations...
  - Frequency domain : largest, mean, energy per frequency band...

HAR ... to be continued

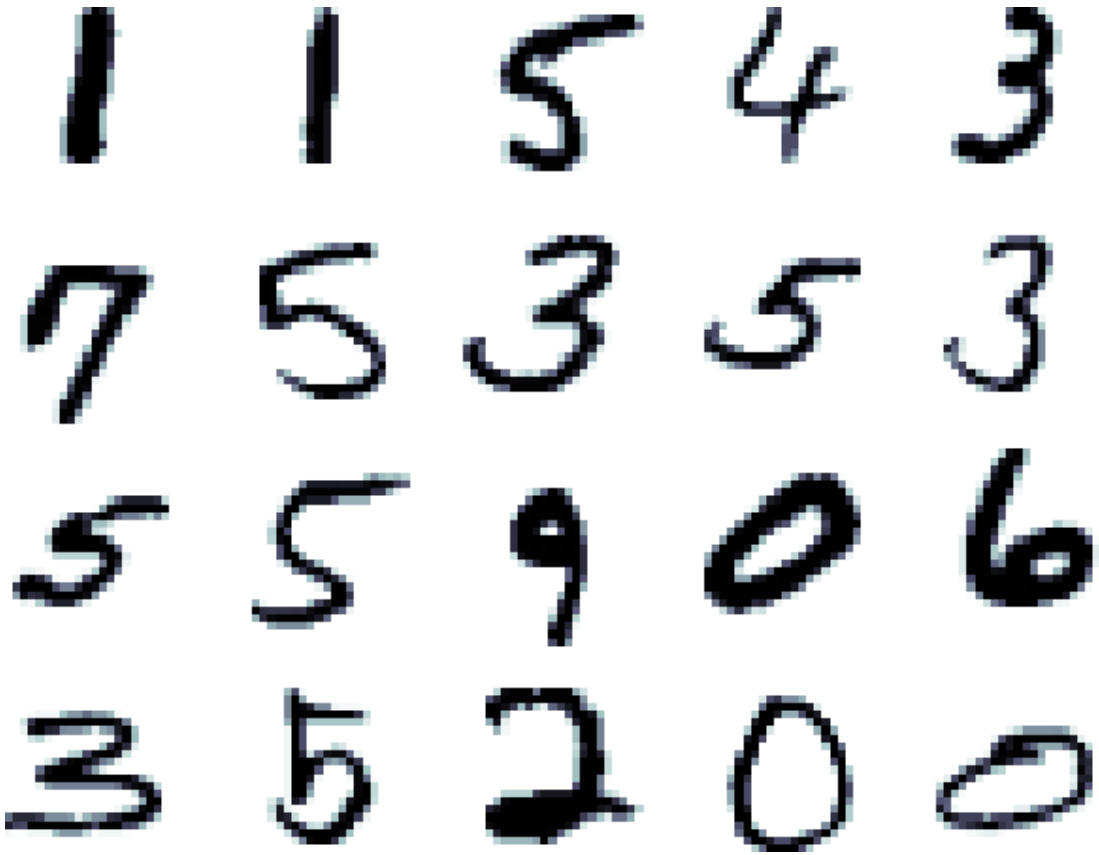
- raw signals and *deep learning*

Usecase MNIST

MNIST dataset

- Yann le Cun website
- 60 000 handwritten digits, $28 \times 28 = 784$ pixels
- Test : 10 000 images
- Classical methods (k-nn, Random Forests)
- Preprocessing : normalisation of the images
- Specific Distance with invariance properties
- Deep learning : *TensorFlow, Keras*

Usecase MNIST



MNIST : some examples of handwritten digits

Methods studied in this course :

Part I

- Estimation of a prediction error
- Linear model for regression, model selection, variable selection, Ridge regression, Lasso.
- Generalized linear models for classification : Logistic regression
- Support Vector Machine

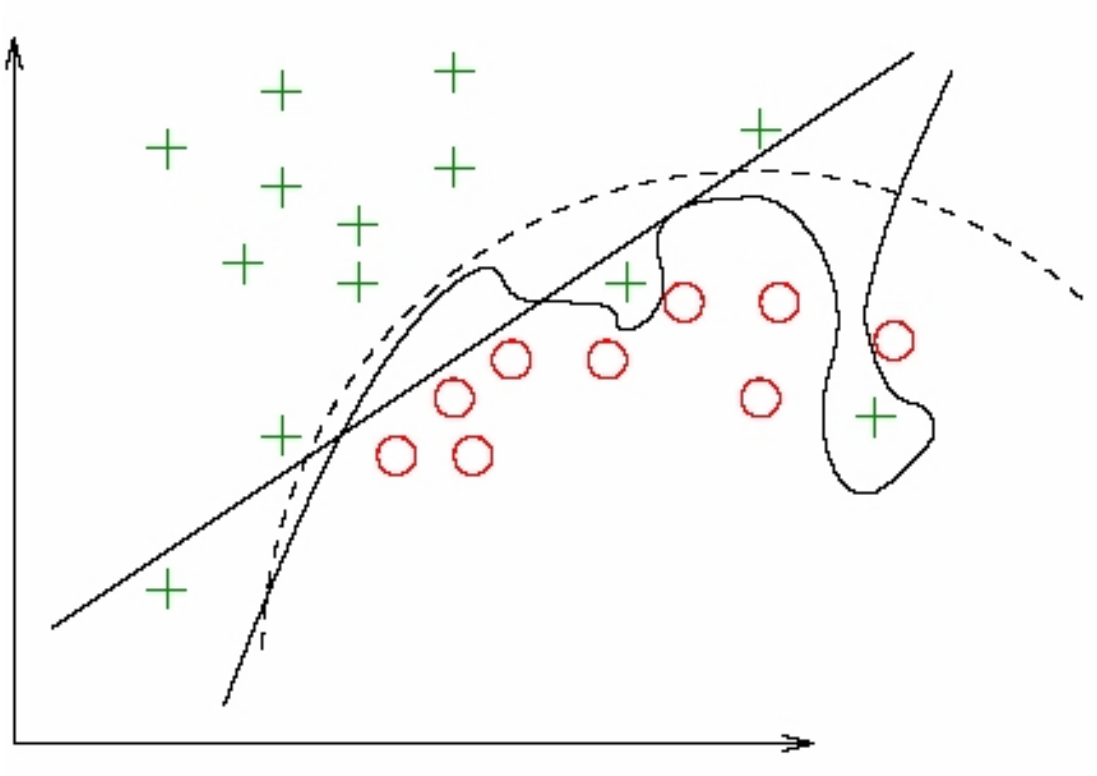
Part II

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

Part I-0 : Estimation of a prediction error

- The generalization performance of a learning procedure is related to its prediction capacity on a new data set, independent of the learning sample that was used to build the learning algorithm.
- Evaluating this performance is crucial to choose a learning method or model among several possible ones.
- It is also important to measure the quality of the ultimately chosen procedure.

Bias, variance, Model complexity



Supervised Classification : *Complexity of the models*

Loss functions

- In a regression framework, one generally considers the $\mathbb{L}_2$-loss :

$$\ell(Y, \hat{f}(\boldsymbol{X})) = (Y - \hat{f}(\boldsymbol{X}))^2,$$

  or the $\mathbb{L}_p$-loss ($p \geq 1$) :

$$\ell(Y, \hat{f}(\boldsymbol{X})) = |Y - \hat{f}(\boldsymbol{X})|^p.$$

- In supervised classification, generally one considers the 0-1 loss :

$$\ell(Y, \hat{f}(\boldsymbol{X})) = \mathbb{1}_{Y \neq \hat{f}(\boldsymbol{X})}.$$

Training error

- Given a loss function ℓ, the *training error* of a prediction rule $\hat{f}$ is defined by

$$\overline{err} = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, \hat{f}(\boldsymbol{X}_i)),$$

where $(\boldsymbol{X}_i, Y_i)_{1 \leq i \leq n}$ is the learning (or training) sample, used to train the algorithm $\hat{f}$.

Evaluation of the risk - generalization error

- Given a loss function ℓ, the *risk* - or *generalization error* - of a prediction rule $\hat{f}$ is defined by

$$R_P(\hat{f}) = \mathbb{E}_{(\boldsymbol{X},Y)\sim P}[\ell(Y, \hat{f}(\boldsymbol{X}))].$$

- Note that, in the above definition, $(\boldsymbol{X}, Y)$ is **independent of the training sample** $\boldsymbol{D}^n = (\boldsymbol{X}_i, Y_i)_{1\leq i\leq n}$ that was used to build the prediction rule $\hat{f}$.

Evaluation of the risk - generalization error

- It is important to estimate the generalization error of a learning algorithm $\hat{f}$: when the model becomes more and more complex, it is able to capture more complex underlying structures in the "true " model : the bias decreases, but at the same time, the estimation error increases, due to the increase of the variance.
- The "optimal" model is the one realizing the best compromise between the bias term and the variance term to give the smallest generalization error.
- The training error is not a good estimate of the generalization error : it decreases as the complexity of the model increases. Hence minimizing the training error leads to select the most complex model, this leads to **overfitting**.

Training error in a regression model

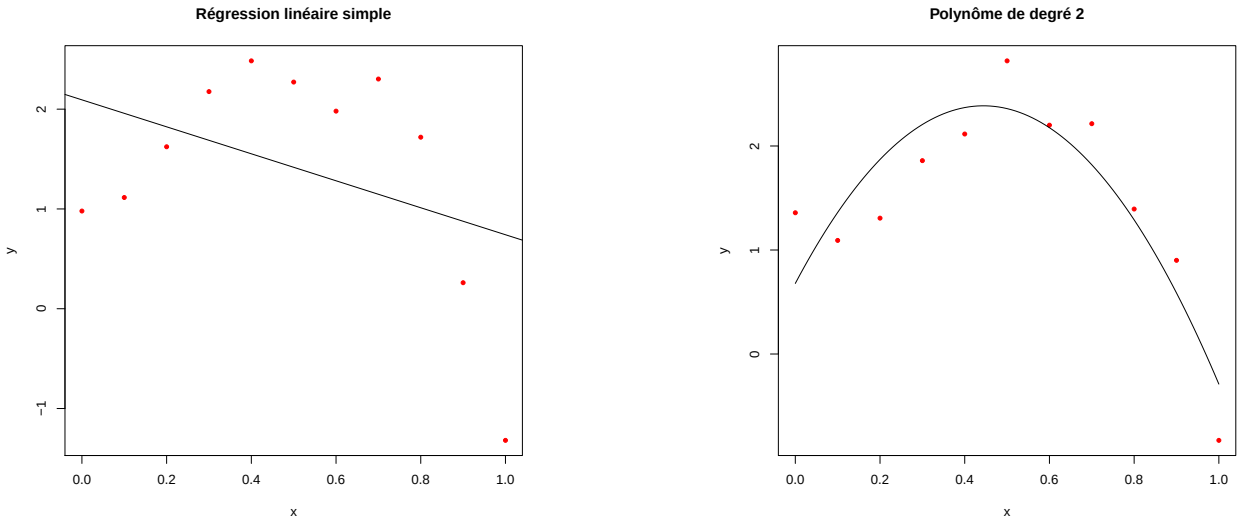


Figure – Polynomial regression : adjusted model, on the left : $y = \beta_0 + \beta_1 x + \epsilon$, on the right : $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$

In a regression model, with the $\mathbb{L}_2$ loss, the training error is equal to the

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \frac{SSR}{n}.$$

Training error in a regression mode

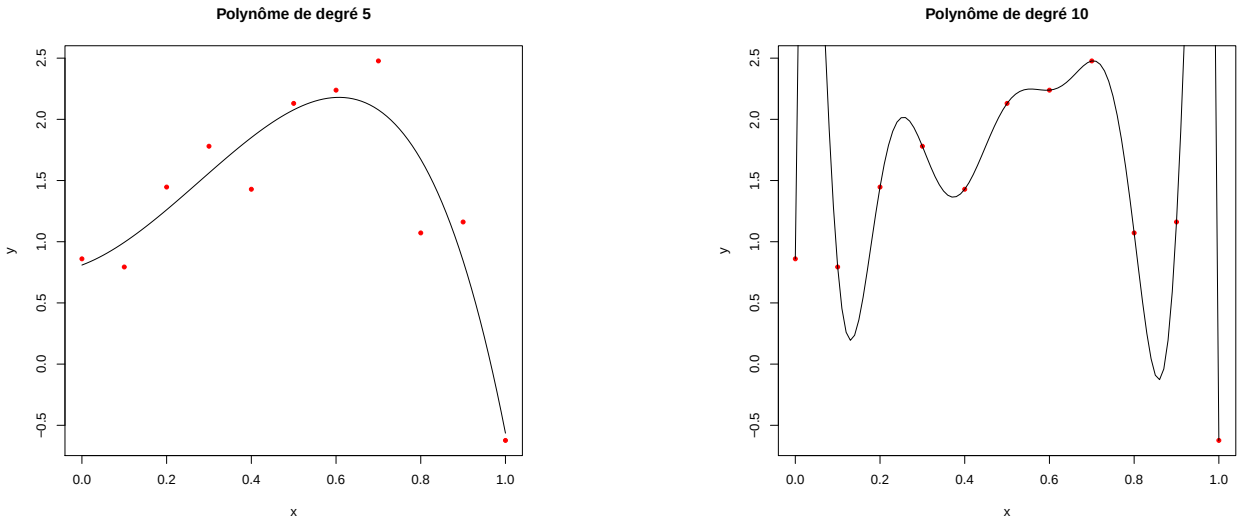


Figure – Polynomial regression : adjusted model, on the left : $y = \beta_0 + \beta_1 x + \ldots + \beta_5 x^5 + \epsilon$, on the right : $y = \beta_0 + \beta_1 x + \ldots + \beta_{10} x^{10} + \epsilon$.

The training error is equal to 0 for the polynomial of degree $n - 1$ (which has n coefficients) and passes through all the training points.

Training error and test error

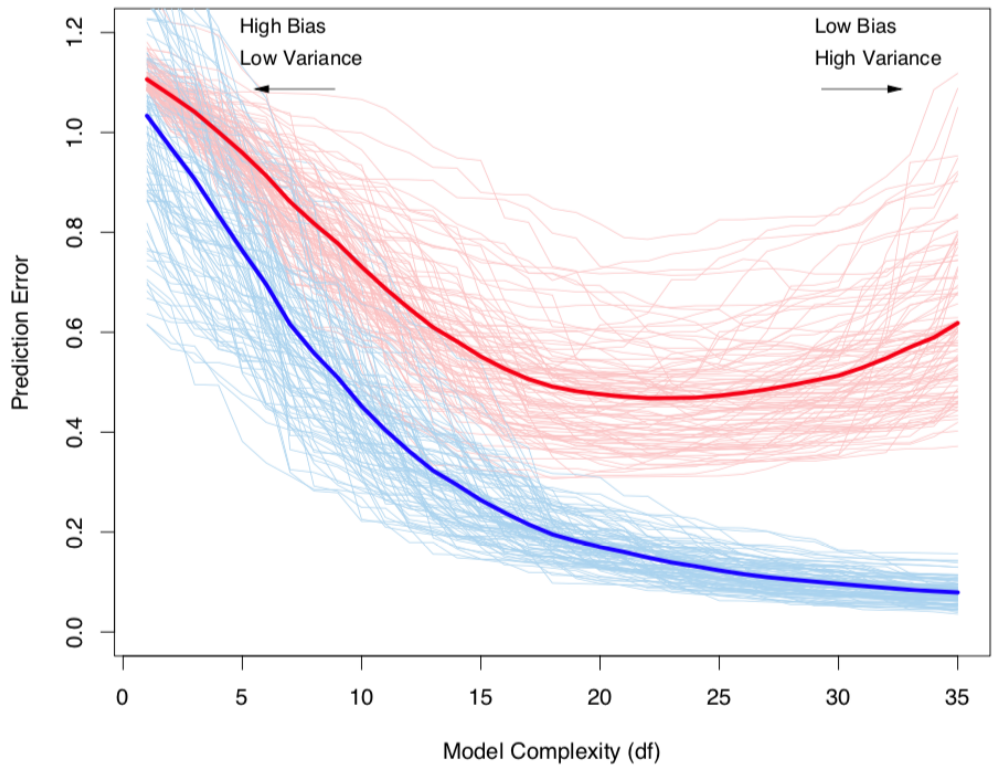


Figure – Behavior of training error (in blue) and test error (in red) as the complexity of the model increases. Source : "The elements of Statistical Learning", T. Hastie, R. Tibshirani, J. Friedman.

Evaluation of the risk - generalization error

- An accurate evaluation of the generalization error has two objectives :
  - **Model selection :** selecting, among a collection of models (or prediction rules), the one with the smallest risk, realizing the best bias/variance trade-off.
  - **Model assessment :** Once the final model has been chosen, evaluating its generalization error on a new data set.

Evaluation of the risk - generalization error

- If we have enough data, the recommended approach is to divide randomly the dataset in three parts : the learning sample, the validation sample, and the test sample.
  - **The learning sample** is used to train the models (generally by minimizing the training error).
  - **The validation sample** is used for model selection : we estimate the generalization error of each model with the validation sample and we select the model with the smallest generalization error.
  - **The test sample** is used to evaluate the risk of the final selected model.
- It is generally recommended to take 50% of the data for the learning sample, 25% of the data for the validation sample and 25% of the data for the test sample.

Cross-validation

- Often, taking only 50% of the data set to train the models may lead to bad performances, especially if we do not have too much data.
- In this case, it is recommended to gather the learning and validation samples and to use K **fold cross-validation** to estimate the generalization error.
- We split randomly the data into K subsamples, with (almost) the same size. ($K = 10$ generally).
- Each of the K folds will be successively used as a test sample.
- When the fold k is the test sample, we train a model with the $K - 1$ other folds, and we evaluate the loss function of this model on each element the fold k.
- This is done for $k = 1, \ldots, K$, and we compute a global estimation of the generalization error.

Cross-validation

- More precisely, assume that we have a n-sample $(\boldsymbol{X}_i, Y_i)_{1 \leq i \leq n}$ and a collection of models $(\hat{f}_m, m \in \mathcal{M})$. We split the data into K folds.
- For $k \in \{1, \ldots, K\}$, let $\hat{f}_m^{(-k)}$ denote the model m trained with all the data, except the fold k.
- The cross-validation estimate of the generalization error of the model m is

$$CV(m) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in k} \ell(Y_i, \hat{f}_m^{(-k)}(\boldsymbol{X}_i)).$$

- $CV(m)$ estimates the generalization error of the model m and we select the model which minimizes $CV(m)$.
- The selected model is then fitted with all the data.

Cross-validation

- When the number of folds $K = n$, the method is called leave-one-out cross-validation.
- The computation time is very high for the leave-one-out method.
- The choice of $K = 5$ or 10 is often recommended.

Strategy for the pratical part (Ozone data)

- For the prediction of the Ozone concentration we will compare several algorithms :
  - Linear models with and without penalization, with and without quadratic terms and interactions between variables
  - Support Vector Machines
  - Regression trees
  - Random Forests
  - Neural networks
- Most of these algorithms have parameters to tune : model selection inner loop to optimize each algorithm.
- We have to select the best of these optimized algorithms : model selection outer loop.

Strategy for the pratical part (Ozone data)

- The first step of the modelization consists in dividing the data set into a training set and a test set.
- The test set is reserved for model assessment of all the optimized algorithms. This will be used for the model selection outer loop.
- For the optimization of each algorithm (model selection inner loop), we use a K-fold cross validation method.
- At the end, we can implement a Monte Carlo cross-validation method to estimate the whole distribution of the risk for each optimized algorithm.

In the next two sections, we will see two methods for supervised learning : the linear model for regression and the logistic model for classification.

Part I-1 :

- Linear model, model selection, variable selection, penalized criterion.
  - Linear model
  - Least square estimation
  - Confidence intervals and prediction intervals
  - Testing a submodel
  - Determination coefficient, Diagnosis on the residuals
  - Model selection, variable selection, penalized criterion

The Linear model

We have a quantitative variable Y *to explain* which is related with p variables $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^p$ called *explanatory variables*.

The data are obtained from the observation of a n sample of $\mathbb{R}^{(p+1)}$ vectors :

$$(x_i^1, \ldots, x_i^j, \ldots, x_i^p, y_i) \quad i = 1, \ldots, n.$$

We assume in a first time that $n > p + 1$.

In *the linear model*, the regression function $\mathbb{E}(\mathbf{Y}/\mathbf{X})$ is linear in the input variables $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^p$.

The Linear model

The linear model is defined by :

$$Y_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + \cdots + \beta_p X_i^p + \varepsilon_i \quad i = 1, 2, \ldots, n$$

with the following assumptions :

1. The random variables ε_i are independent and identically distributed (i.i.d.) ; they are independent of $(\boldsymbol{X}^1, \ldots, \boldsymbol{X}^p)$, $\mathbb{E}(\varepsilon_i) = 0$, $Var(\varepsilon_i) = \sigma^2$.
2. We have

   $E(\mathbf{Y}|\boldsymbol{X}^1, \ldots, \boldsymbol{X}^p) = \beta_0 + \beta_1 \boldsymbol{X}^1 + \beta_2 \boldsymbol{X}^2 + \cdots + \beta_p \boldsymbol{X}^p$ and $Var(\mathbf{Y}|\boldsymbol{X}^1, \ldots, \boldsymbol{X}^p) = \sigma^2$.
3. The unknown parameters $\beta_0, \ldots, \beta_p$ are supposed to be constant.
4. It is sometimes assumed that the errors are Gaussian : $\boldsymbol{\varepsilon} = [\varepsilon_1 \cdots \varepsilon_n]' \sim \mathcal{N}_n(0, \sigma^2 \mathbf{I_n})$. The variables ε_i are then i.i.d. $\mathcal{N}(0, \sigma^2)$.

The Linear model

- The explanatory variables are given in the matrix $\mathbf{X}(n \times (p+1))$.
- The regressors $\boldsymbol{X}^j$ can be quantitative variables, nonlinear transformation of quantitative variables (such as log, exp, square ..), interactions : $\boldsymbol{X}^j = \boldsymbol{X}^k.\boldsymbol{X}^l$.
- They can also correspond to qualitative variables : in this case the variables $\boldsymbol{X}^j$ are indicator variables coding the different levels of a factor.
- The response variable is given in the vector $\mathbf{Y}$.
- We set $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \cdots \ \beta_p]'$, which leads to the matricial formulation of the linear model :

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Example

We consider the **Ozone data set** .
The data frame has 1041 observations of the following components :

JOUR	type of the day ; public holiday(1) or not (0)
O3obs	Ozone concentration observed the next day at 17h., generally the maximum of the day
MOCAGE	Prediction of this pollution obtained by a deterministic model of fluid mechanics
TEMPE	Temperature forecast by MétéoFrance for the next day 17h
RMH2O	Moisture ratio
NO2	Nitrogen dioxide concentration
NO	Concentration of nitric oxide
STATION	Location of the observation : Aix-en-Provence, Rambouillet, Munchhausen, Cadarache and Plan de Cuques
VentMOD	Wind force
VentANG	Orientation of the wind.

- We denote by Y the variable (**O3obs**) to explain.
- We set $X^1, \ldots X^p$ for the explanatory variables (**MOCAGE** , **TEMPE**, **JOUR** ..). The variables are quantitative (**MOCAGE** , **TEMPE** , ...), or qualitative (**JOUR**, **STATION**).
- We consider the linear model :

$$Y_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + \ldots + \beta_p X_i^p + \varepsilon_i, \ 1 \leq i \leq n,$$

- For the qualitative variables, we consider indicator functions of the different levels of the factor, and introduce some constraints for identifiability. By default, in R, the smallest value of the factor are set in the reference.
  This is an analysis of covariance model (mixing quantitative and qualitative variables).

Least square estimation

- The unknown parameters of the model are the vector β and σ^2.
- β is estimated by minimizing the residuals sum of square.
- We minimise with respect to the parameter $\beta \in \mathbb{R}^{p+1}$ the criterion :

$$\begin{aligned}\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_i^1 - \cdots - \beta_p X_i^p)^2 &= \|\mathbf{Y} - \mathbf{X}\beta\|^2 \\ &= (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta) \\ &= \mathbf{Y}'\mathbf{Y} - 2\beta'\mathbf{X}'\mathbf{Y} + \beta'\mathbf{X}'\mathbf{X}\beta.\end{aligned}$$

Lemma

Let $h : \beta \mapsto \beta' A \beta$ where A is a symmetric matrix.
Then $\bigtriangledown h(\beta) = 2A\beta$.
Let $g : \beta \mapsto \beta' z = z'\beta = \langle z, \beta \rangle$ where $z \in \mathbb{R}^p$.
Then $\bigtriangledown g(\beta) = z$.

Least square estimation

- Derivating the last equation, we obtain the *normal equations* :

$$2(\mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{X}\beta) = 0$$

- The solution is a minimizer of the criterion since the Hessian $2\mathbf{X}'\mathbf{X}$ is positive semi definite (the criterion is convex) .

Least square estimation

We make the additional assumption that the matrix $\mathbf{X}'\mathbf{X}$ is invertible. Under this assumption, the estimation of β is given by :

$$\widehat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

and the predicted values of $\mathbf{Y}$ are :

$$\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{H}\mathbf{Y}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is called the "*hat matrix*".

Geometrically, it corresponds to the matrix of orthogonal projection in $\mathbb{R}^n$ onto the subspace Vect($\mathbf{X}$) generated by the columns of $\mathbf{X}$.

Least square estimation

- If $\mathbf{X}'\mathbf{X}$ is not invertible, the application $\beta \mapsto \mathbf{X}\beta$ is not injective, hence the model is not identifiable and β is not uniquely defined.
- In this case, the predicted values $\widehat{\mathbf{Y}}$ are still defined as the projection of $\mathbf{Y}$ onto the space generated by the columns of $\mathbf{X}$.
- In practice, if $\mathbf{X}'\mathbf{X}$ is not invertible (which is necessarily the case in high dimension when $p > n$), we have to remove variables from the model or to consider other approches to reduce the dimension (*Ridge*, Lasso, PLS ...).

Least square estimation

- We define the vector of residuals as :

$$\mathbf{e} = \mathbf{Y} - \widehat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\widehat{\beta} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$$

- This is the orthogonal projection of $\mathbf{Y}$ onto the subspace $\text{Vect}(\mathbf{X})^{\perp}$ in $\mathbb{R}^n$.
- The variance σ^2 is estimated by

$$\widehat{\sigma}^2 = \frac{\|\mathbf{e}\|^2}{n-p-1} = \frac{\left\|\mathbf{Y} - \mathbf{X}\hat{\beta}\right\|^2}{n-p-1}.$$

Properties of the least square estimator

Theorem

— *Assuming that*

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

with $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 \mathbf{I_n})$, *we obtain that* $\widehat{\beta}$ *is a Gaussian vector :*

$$\widehat{\beta} \sim \mathcal{N}_{p+1}(\beta, \sigma^2 (X'X)^{-1}).$$

In particular, the components of $\widehat{\beta}$ *are Gaussian variables :*

$$\widehat{\beta}_j \sim \mathcal{N}(\beta_j, \sigma^2 (X'X)^{-1}_{j,j}).$$

$$\hat{\sigma}^2 \sim \frac{\sigma^2}{n-(p+1)} \chi^2_{(n-(p+1))}$$

and is independent of $\hat{\beta}$.

Confidence intervals

One can easily deduce from the first theorem that

$$\frac{\widehat{\beta}_j - \beta_j}{\sqrt{\hat{\sigma}^2 (X'X)^{-1}_{j,j}}} \sim \mathcal{T}_{(n-(p+1))}.$$

This allows to build confidence intervals and tests of significance for the parameters β_j.
The following interval is a 0.95 **confidence interval for** β_j :

$$\left[\widehat{\beta}_j - t_{n-(p+1),0.975}\sqrt{\hat{\sigma}^2 (X'X)^{-1}_{j,j}}, \widehat{\beta}_j + t_{n-(p+1),0.975}\sqrt{\hat{\sigma}^2 (X'X)^{-1}_{j,j}}\right].$$

In order to test $H_0 : \beta_j = 0$ contre $H_1 : \beta_j \neq 0$, we reject the null hypothesis at the level 5% if 0 does not belong to the previous confidence interval.

Test of significance

- We recall the linear model

$$Y_i = \beta_0 + \beta_1 X_i^1 + \beta_2 X_i^2 + \cdots + \beta_p X_i^p + \varepsilon_i \quad i = 1, 2, \ldots, n$$

- We want to test if the variable X^j is significant in the model or not, which is equivalent to test the nullity of the parameter β_j.
- We test $H_0 : \beta_j = 0$ against $H_1 : \beta_j \neq 0$.
- Under the hypothesis H_0,

$$T_j = \frac{\widehat{\beta}_j}{\sqrt{\hat{\sigma}^2 (X'X)^{-1}_{j,j}}} \sim \mathcal{T}_{(n-(p+1))}.$$

Test of significance

- The p-value of the test is defined as

$$\mathbb{P}_{H_0}(|T_j| > |T_j|_{obs}) = \mathbb{P}(|\mathcal{T}_{(n-(p+1))}| > |T_j|_{obs}),$$

  where $|T_j|_{obs}$ is the observed value for the variable $|T_j|$ with our data.
- If the p-value is very small, then it is unlikely that $|T_j|_{obs}$ is obtained from a Student distribution with $n-(p+1)$ degrees of freedom, hence we will reject the hypothesis H_0, and conclude that the variable X^j is significant.
- We fix some level α (generally 5%) for the test.
- If p-value $< \alpha$, we reject the nullity of β_j and conclude that the variable X^j is significant in the model.
- One easily prove that the probability to reject H_0 when it is true (i.e. to conclude that the variable X^j is significant when it is not) is less than the level α of the test.

Example

We consider the **Ozone data set** .
The data frame has 1041 observations of the following components :

JOUR	type of the day ; public holiday(1) or not (0)
O3obs	Ozone concentration observed the next day at 17h., generally the maximum of the day
MOCAGE	Prediction of this pollution obtained by a deterministic model of fluid mechanics
TEMPE	Temperature forecast by MétéoFrance for the next day 17h
RMH2O	Moisture ratio
NO2	Nitrogen dioxide concentration
NO	Concentration of nitric oxide
STATION	Location of the observation : Aix-en-Provence, Rambouillet, Munchhausen, Cadarache and Plan de Cuques
VentMOD	Wind force
VentANG	Orientation of the wind.

We first consider a simple linear regression model with the single variable $X =$ **MOCAGE**

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \ \ i = 1, \ldots, n.$$

For the least square estimation, we obtain the following results :

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	37.78887	3.42998	11.02	<2e-16 ***
MOCAGE	0.61006	0.02573	23.71	<2e-16 ***

Residual standard error : 33.04 on 1039 degrees of freedom

Multiple R-squared : 0.3511, Adjusted R-squared : 0.3505

F-statistic : 562.1 on 1 and 1039 DF, p-value : $<$ 2.2e-16

We consider here a linear regression model with all the variables :

$$Y_i = \beta_0 + \beta_1 X_i^1 + \ldots + \beta_p X_i^p + \varepsilon_i, \ \ i = 1, \ldots, n.$$

For the least square estimation, with the default constraints of R, we obtain the following results :

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-33.43948	6.98313	-4.789	1.93e-06 ****
JOUR1	0.46159	1.88646	0.245	0.806747
MOCAGE	0.37509	0.03694	10.153	$<$ 2e-16 ***
TEMPE	3.96507	0.22135	17.913	$<$ 2e-16 ***
...	...	...	...	...

Residual standard error : 27.83 on 1028 degrees of freedom

Multiple R-squared : 0.5445, Adjusted R-squared : 0.5391

F-statistic : 102.4 on 12 and 1028 DF, p-value : $<$ 2.2e-16

Prediction

As mentioned above, the vector of predicted values is

$$\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{H}\mathbf{Y}.$$

Based on the n previous observations, we may be interested with the prediction of the response of the model for a new point $\mathbf{X_0}' = (1, X_0{}^1, \ldots, X_0{}^p)$:

$$Y_0 = \beta_0 + \beta_1 X_0^1 + \beta_2 X_0^2 + \ldots + \beta_p X_0^p + \varepsilon_0,$$

where $\varepsilon_0 \sim \mathcal{N}(0, \sigma^2)$.
The predicted value is

$$\widehat{Y}_0 = \widehat{\beta}_0 + \widehat{\beta}_1 X_0{}^1 + \ldots \widehat{\beta}_p X_0{}^p = \mathbf{X_0}'\widehat{\beta}.$$

Prediction

- We derive from the previous theorem that

$$\mathbb{E}(\widehat{Y}_0) = \mathbf{X_0}'\beta = \beta_0 + \beta_1 X_0^1 + \beta_2 X_0^2 + \ldots + \beta_p X_0^p$$

and that $\widehat{Y}_0 \sim \mathcal{N}(\mathbf{X_0}'\beta, \sigma^2 \mathbf{X_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X_0})$. Let $t = t_{n-(p+1),0.975}$.

- **Confidence interval for the mean response $\mathbf{X_0}'\beta$** at the new observation point $\mathbf{X_0}$:

$$\left[\mathbf{X_0}'\widehat{\beta} - t\hat{\sigma}\sqrt{\mathbf{X_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X_0}}, \mathbf{X_0}'\widehat{\beta} + t\hat{\sigma}\sqrt{\mathbf{X_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X_0}}\right].$$

- **Prediction interval for the response Y_0** at the new observation point $\mathbf{X_0}$ is :

$$\left[\mathbf{X_0}'\widehat{\beta} - t\hat{\sigma}\sqrt{1 + \mathbf{X_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X_0}}, \mathbf{X_0}'\widehat{\beta} + t\hat{\sigma}\sqrt{1 + \mathbf{X_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X_0}}\right].$$

We consider here a simple linear regression model with the single variable $X =$ **MOCAGE**

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \ \ i = 1, \ldots, n.$$

For the least square estimation, we obtain the following results :

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	37.78887	3.42998	11.02	<2e-16 ***
MOCAGE	0.61006	0.02573	23.71	<2e-16 ***

Residual standard error : 33.04 on 1039 degrees of freedom

Multiple R-squared : 0.3511, Adjusted R-squared : 0.3505

F-statistic : 562.1 on 1 and 1039 DF, p-value : $<$ 2.2e-16

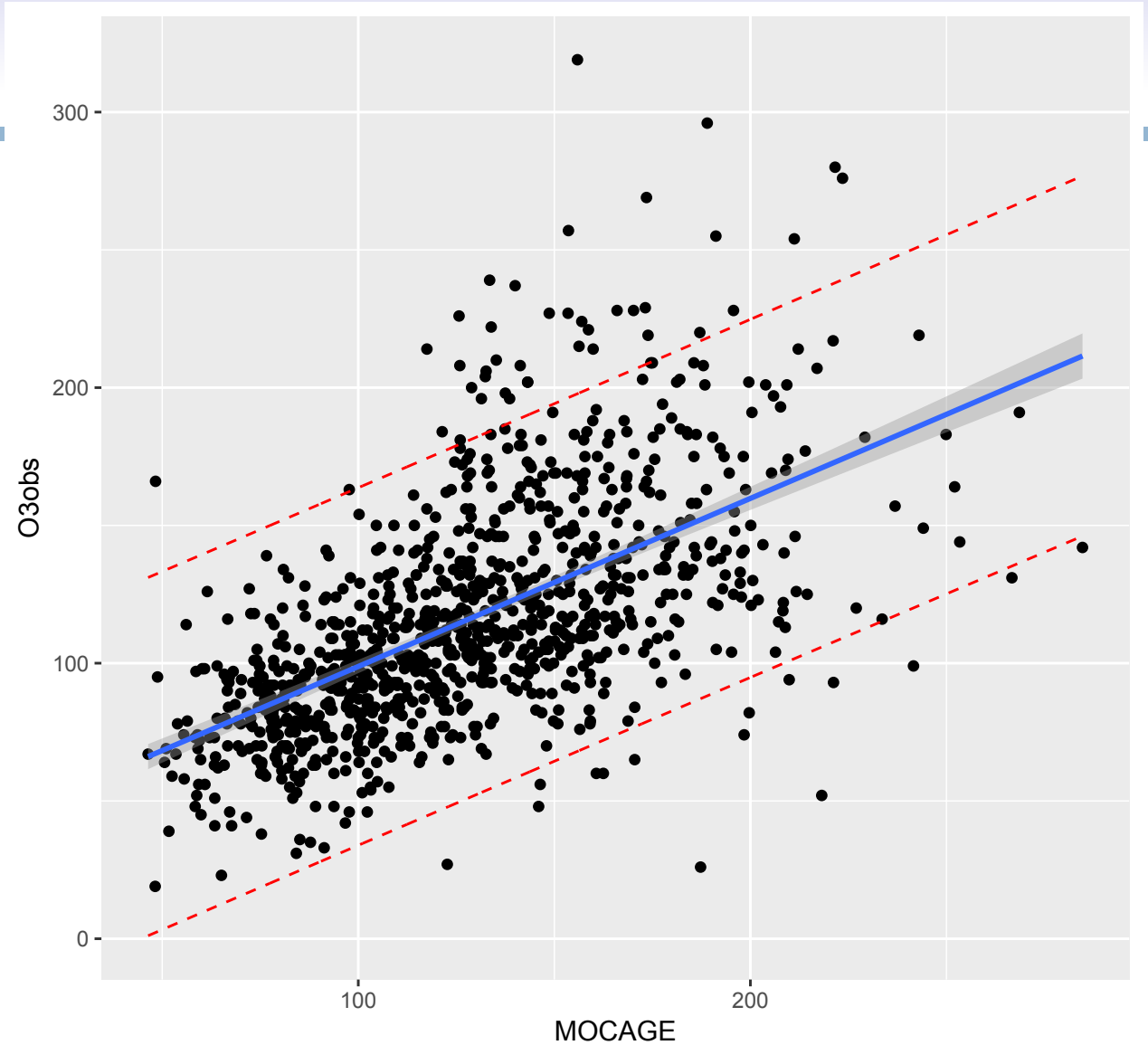


Figure – Simple linear regression model : confidence and prediction intervals

We consider here a linear regression model with all the variables :

$$Y_i = \beta_0 + \beta_1 X_i^{(1)} + \ldots + \beta_p X_i^{(p)} + \varepsilon_i, \ \ i = 1, \ldots, n.$$

For the least square estimation, with the default constraints of R, we obtain the following results :

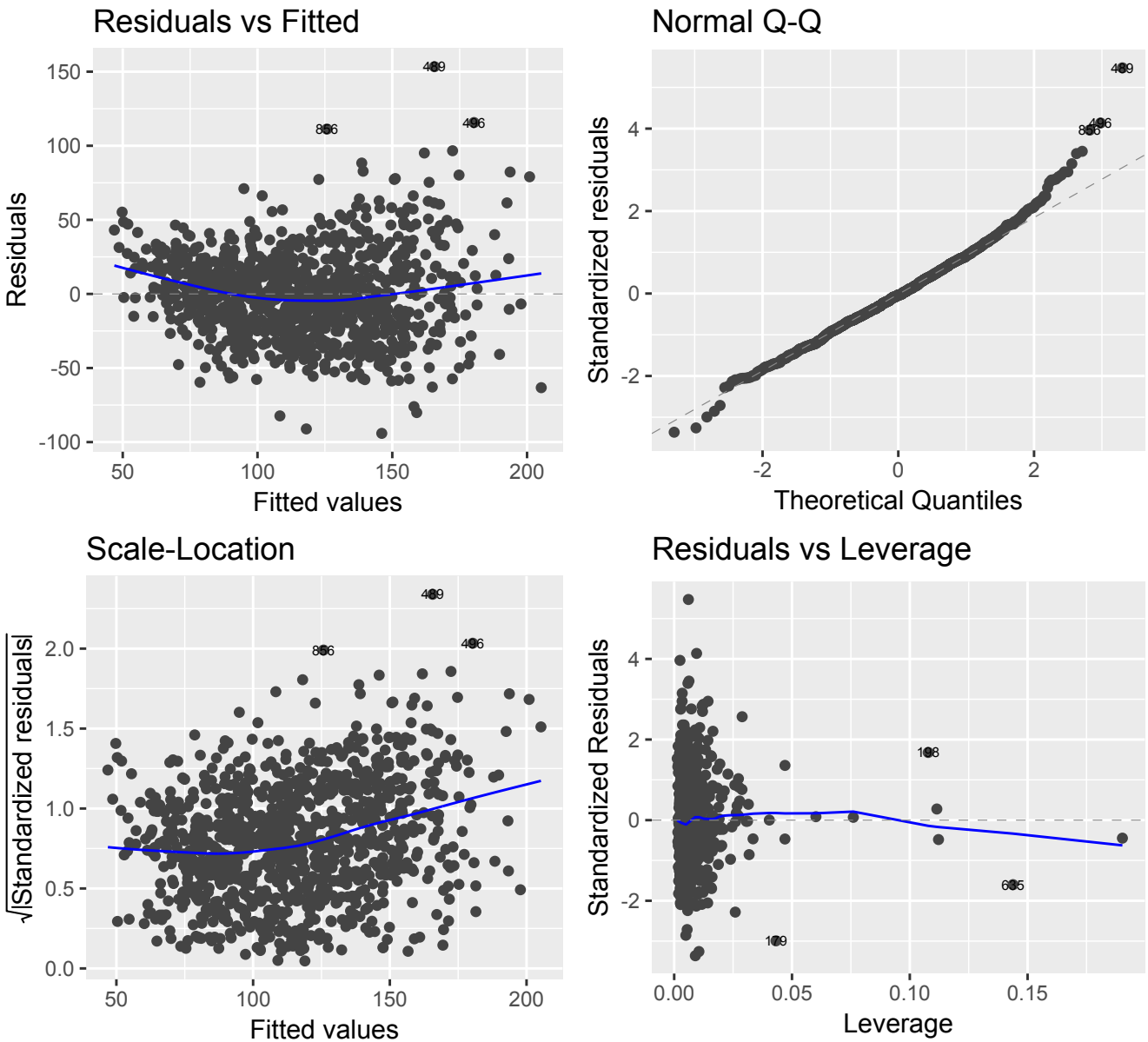
Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-33.43948	6.98313	-4.789	1.93e-06 ****
JOUR1	0.46159	1.88646	0.245	0.806747
MOCAGE	0.37509	0.03694	10.153	$<$ 2e-16 ***
TEMPE	3.96507	0.22135	17.913	$<$ 2e-16 ***
...	...	...	...	...

Residual standard error : 27.83 on 1028 degrees of freedom

Multiple R-squared : 0.5445, Adjusted R-squared : 0.5391

F-statistic : 102.4 on 12 and 1028 DF, p-value : $<$ 2.2e-16

Diagnosis on the residuals



Measures for goodness-of-fit

$$\mathrm{SST} = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 = \left\| \mathbf{Y} - \overline{\mathbf{Y}} \mathbb{1} \right\|^2,$$

$$\mathrm{SSE} = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 = \left\| \widehat{\mathbf{Y}} - \overline{\mathbf{Y}} \mathbb{1} \right\|^2,$$

$$\mathrm{SSR} = \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 = \left\| \mathbf{Y} - \widehat{\mathbf{Y}} \right\|^2 = \|\mathbf{e}\|^2.$$

$$\left\| \mathbf{Y} - \overline{\mathbf{Y}} \mathbb{1} \right\|^2 = \left\| \mathbf{Y} - \widehat{\mathbf{Y}} \right\|^2 + \left\| \widehat{\mathbf{Y}} - \overline{\mathbf{Y}} \mathbb{1} \right\|^2,$$

hence

$$\mathrm{SST} = \mathrm{SSR} + \mathrm{SSE}.$$

$$R^2 = \frac{\mathrm{SSE}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSR}}{\mathrm{SST}}.$$

Note that $0 \leq R^2 \leq 1$.

Determination coefficient and Model selection

The model is well adjusted to the n training data if the determination coefficient R^2 is close to 1.
Hence, the first hint is that a "good" model is a model for which R^2 is close to 1.
This is in fact not true.
Suppose that we have a training sample $(X_i, Y_i)_{1 \leq i \leq n}$ where $X_i \in [0, 1]$ and $Y_i \in \mathbb{R}$ and we adjust polynomials on these data :

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \ldots + \beta_k X_i^k + \varepsilon_i.$$

When k increases, the model is more and more complex, hence $\left\| \mathbf{Y} - \widehat{\mathbf{Y}} \right\|^2$ decreases, and R^2 increases as shown in Figures 5 and 6.

Determination coefficient and Model selection

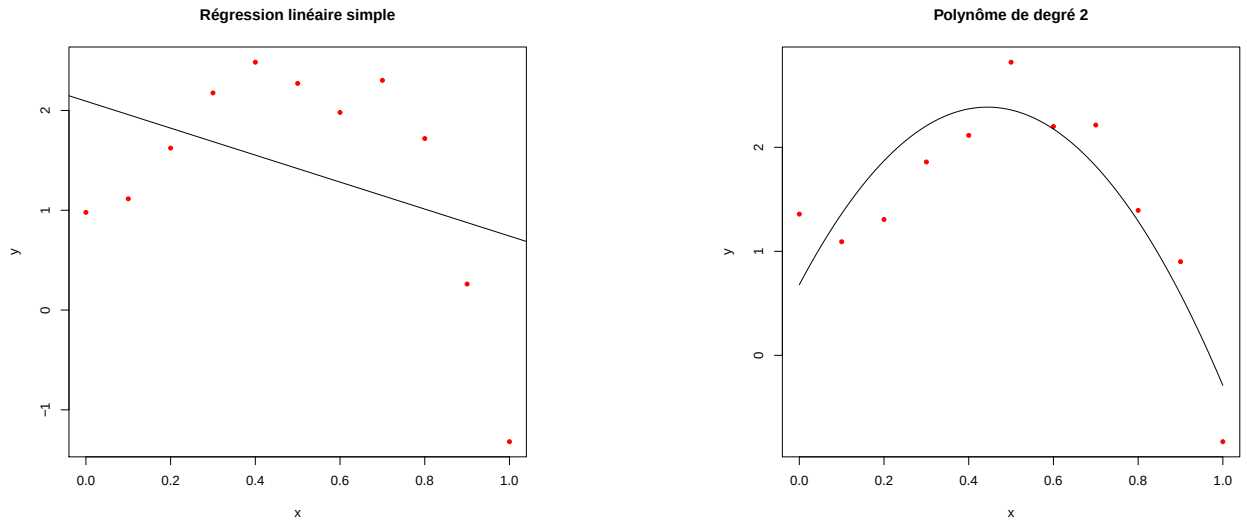


Figure – Polynomial regression : adjusted model, on the left : $y = \beta_0 + \beta_1 x + \epsilon$, $R^2 = 0.03$, on the right : $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$, $R^2 = 0.73$.

Determination coefficient and Model selection

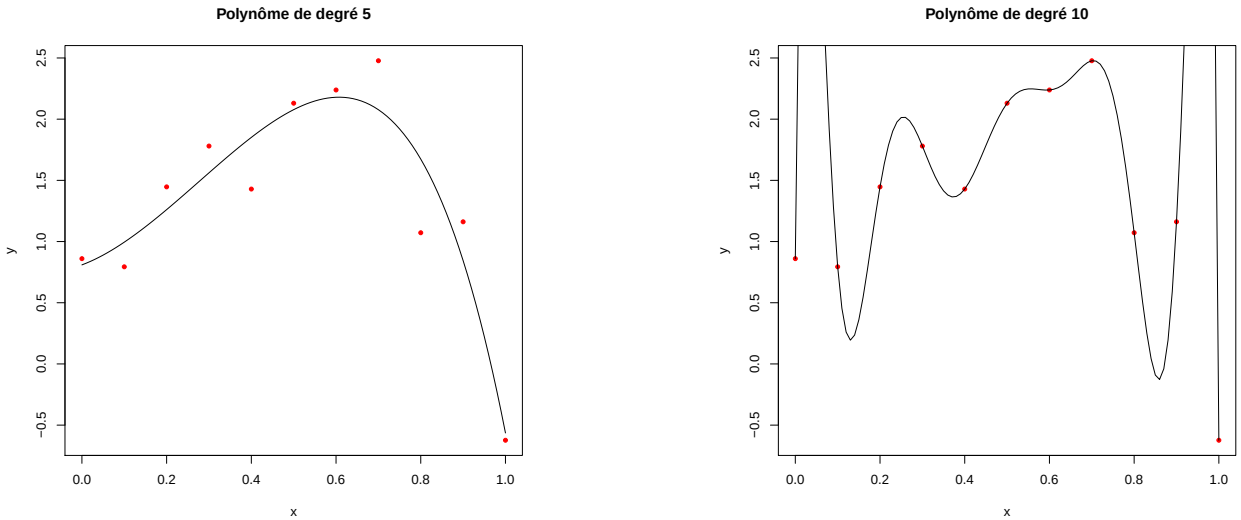


Figure – Polynomial regression : adjusted model, on the left : $y = \beta_0 + \beta_1 x + \ldots + \beta_5 x^5 + \epsilon$, $R^2 = 0.874$, on the right : $y = \beta_0 + \beta_1 x + \ldots + \beta_{10} x^{10} + \epsilon$, $R^2 = 1$.

The determination coefficient is equal to 1 for the polynomial of degree $n-1$ (which has n coefficients) and passes through all the training points.

Model selection

- The best model is the one that realizes the best trade-off between the bias term and the variance term.
- Maximizing the determination coefficient is not a good criterion to compare models with various complexity.
- It is more interesting to consider the adjusted determination coefficient defined by :

$$R'^2 = 1 - \frac{\mathrm{SSR}/(n-k-1)}{\mathrm{SST}/(n-1)}.$$

  The definition of R'^2 takes into account the complexity of the model, represented here by its number of coefficients : $k+1$ for a polynomial of degree k, and penalizes more complex models.
- One can choose, between several models, the one which maximizes the adjusted R^2. In the previous example, we would choose a polynomial of degree 3 with this criterion.

Ozone data

We first consider a simple linear regression model with the single variable $X =$ **MOCAGE**

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \ \ i = 1, \ldots, n.$$

For the least square estimation, we obtain the following results :

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	37.78887	3.42998	11.02	<2e-16 ***
MOCAGE	0.61006	0.02573	23.71	<2e-16 ***

Residual standard error : 33.04 on 1039 degrees of freedom

Multiple R-squared : 0.3511, Adjusted R-squared : 0.3505

F-statistic : 562.1 on 1 and 1039 DF, p-value : $<$ 2.2e-16

Ozone data

We consider here a linear regression model with all the variables :

$$Y_i = \beta_0 + \beta_1 X_i^1 + \ldots + \beta_p X_i^p + \varepsilon_i, \;\; i = 1, \ldots, n.$$

For the least square estimation, with the default constraints of R, we obtain the following results :

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-33.43948	6.98313	-4.789	1.93e-06 ****
JOUR1	0.46159	1.88646	0.245	0.806747
MOCAGE	0.37509	0.03694	10.153	$<$ 2e-16 ***
TEMPE	3.96507	0.22135	17.913	$<$ 2e-16 ***
...	...	...	...	...

Residual standard error : 27.83 on 1028 degrees of freedom

Multiple R-squared : 0.5445, Adjusted R-squared : 0.5391

F-statistic : 102.4 on 12 and 1028 DF, p-value : $<$ 2.2e-16

Model selection

- We have to define model selection procedures that realize a good compromise between a good adjustment to the data (small bias) and a small variance. We will prefer a biased model if this allows to reduce drastically the variance.
- There are several ways to do that :
  - Reducing the number of explanatory variables and by the same way simplifying the model (variable selection or *Lasso* penalization)
  - Adding some constraints on the parameters of the model by *shrinking* them (*Ridge* or *Lasso* penalization)

Variable selection

- We want to select a subset of variables among all possible subsets taken from the input variables.
- Each subset defines a model, and we want to select the "best model".
- Maximizing the R^2 is not a good criterion since this lead to select the full model.
- It is more interesting to select the model maximizing the adjusted determination coefficient R'^2.
- Many other penalized criterion have been introduce for variable selection such as the Mallow's C_P criterion or the BIC criterion.
- In both cases, it corresponds to the minimization of the least square criterion plus some penalty term, depending on the number k of parameters in the model m that is considered.

$$\text{Crit}(m) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \text{pen(k)}.$$

Variable selection

The Mallow's C_P criterion is

$$\text{Crit}_{C_P}(m) = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + 2k\sigma^2,$$

and the BIC criterion penalizes more the dimension of the model with an additional logarithmic term.

$$\text{Crit}_{BIC}(m) = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \log(n)k\sigma^2.$$

The aim is to select the model (among all possible subsets) that minimizes one of those criterion. On the example of the polynomial models, we obtain the results summarized in the next Figure.

Variable selection

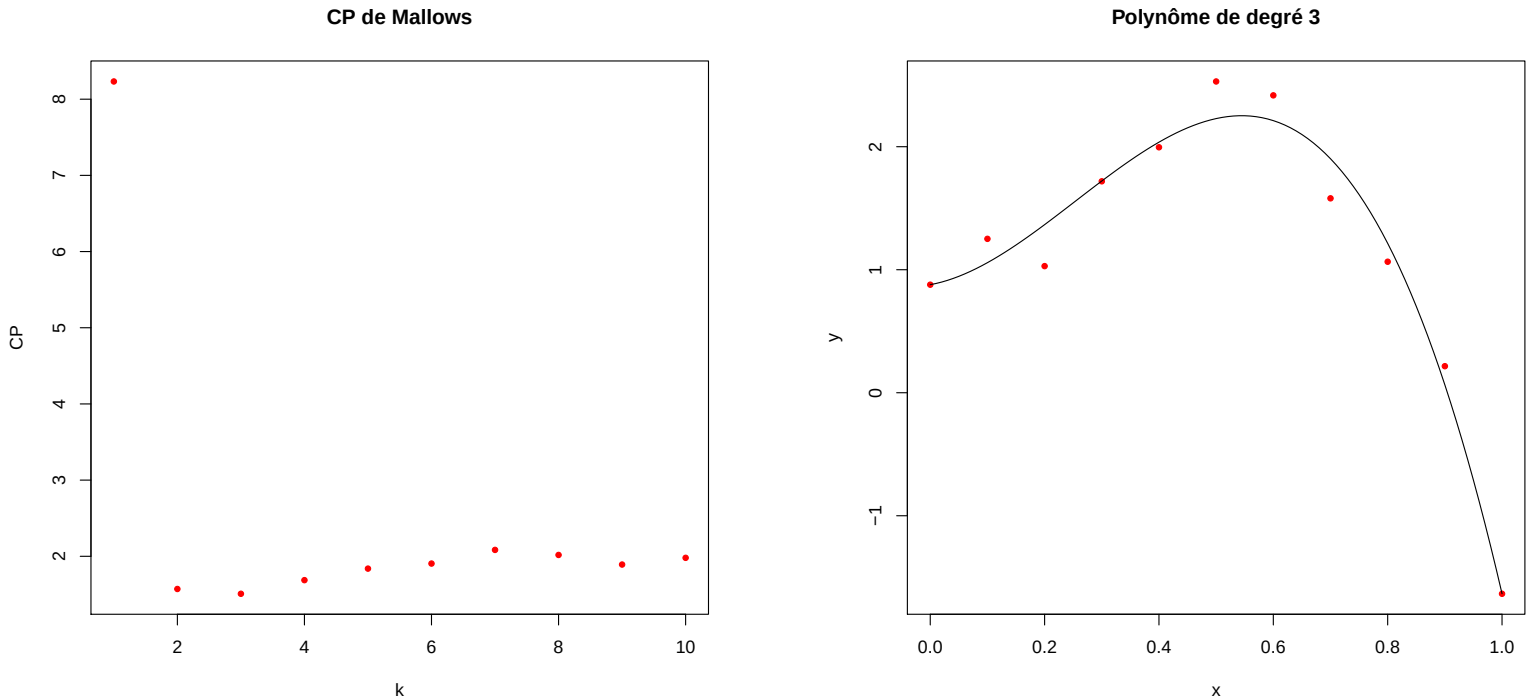


Figure – Mallows' C_P in function of the degree of the polynomial. Selected model : polynomial with degree 3.

Variable selection

- The number of subsets of a set of p variables is 2^p, and it is impossible (as soon as $p > 30$) to explore all the models to minimize the criterion.
- Fast algorithms have been developed to find a clever way to explore a subsample of the models.
- This are the *backward, forward* and *stepwise* algorithms :
  - **Forward selection :** We start from the constant model (only the intercept, no explanatory variable), and we add sequentially the variable that allows to reduce the more the criterion.
  - **Backward selection :** This is the same principle, but starting from the full model and removing one variable at each step in order to reduce the criterion.
  - **Stepwise selection :** This is a mixed algorithm, adding or removing one variable at each step in order to reduce the criterion in the best way.

All those algorithms stop when the criterion can no more be reduced.

Variable selection

Applications of the **Stepwise Algorithm** to the **Ozone** data. We apply the StepAIC algorithm, with the option **both** of the software R in order to select a subset of variables, and we present here an intermediate result :

```
                    Start: AIC=6953.05
O3obs ~ MOCAGE + TEMPE + RMH2O + NO2 + NO + VentMOD + VentANG
                Df   Sum of Sq   RSS      AIC
- VentMOD       1    1484        817158   6952.9
<none>                           815674   6953.0
- RMH2O         1    4562        8202354  6956.9
- VentANG       1    12115       827788   6966.4
- NO2           1    21348       837022   6977.9
- NO            1    21504       837178   6978.1
- MOCAGE        1    225453      1041127  7205.1
- TEMPE         1    268977      1084651  7247.7
                    Step: AIC= 6952.94
O3obs ~ MOCAGE + TEMPE + RMH2O + NO2 + NO + VentANG
```

Ridge regression

The principle of the Ridge regression is

- to consider all the explanatory variables
- to introduce constraints on the parameters in order to avoid overfitting, and by the same way in order to reduce the variance of the estimators.
- In the case of the Ridge regression, we introduce an l_2 constraint on the parameter β.

Model and estimation

We consider the linear model

$$\mathbf{Y} = \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon},$$

where

$$\widetilde{\mathbf{X}} = \begin{pmatrix} 1 & X_1^1 & X_1^2 & . & X_1^p \\ 1 & X_2^1 & X_2^2 & . & X_2^p \\ . & . & . & . & . \\ 1 & X_n^1 & X_n^2 & . & X_n^p \end{pmatrix},$$

$$\widetilde{\boldsymbol{\beta}} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ . \\ . \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ \beta_p \end{pmatrix}.$$

$\mathbf{X}$ is the matrix $\widetilde{\mathbf{X}}$ where we have removed the first column.

The *ridge* estimator is defined by a least square criterion plus a penalty term, with an l_2 type penalty (note that the parameter β_0 is not penalized).

Definition

The *ridge* estimator of $\widetilde{\boldsymbol{\beta}}$ in the model $\mathbf{Y} = \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}} + \boldsymbol{\epsilon}$, is defined by

$$\widehat{\boldsymbol{\beta}}_{\text{Ridge}} = \text{argmin}_{\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}} \left(\|\mathbf{Y} - \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}\|^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right),$$

where λ is a non negative parameter, that we have to calibrate (tuning parameter).

Assume that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are centered. We can find the *ridge* estimator by resolving the normal equations :

$$\mathbf{X}'\mathbf{Y} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)\beta.$$

We get

$$\hat{\beta}_0 = \bar{Y},\ \widehat{\beta}_R = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{Y}.$$

The solution is therefore explicit and linear with respect to $\mathbf{Y}$.

Remarks :

1. $\mathbf{X}'\mathbf{X}$ is a nonnegative symmetric matrix. Hence, for any $\lambda > 0$, $\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p$ is invertible.
2. The constant β_0 is not penalized, otherwise, the estimator would depend on the choice of the origin for $\mathbf{Y}$. We obtain $\widehat{\beta}_0 = \overline{\mathbf{Y}}$, adding a constant to $\mathbf{Y}$ does not modify the values of $\widehat{\beta}_j$ for $j \geq 1$.
3. The *ridge* estimator is not invariant by normalization of the vectors $X^{(j)}$, it is therefore important to normalize the vectors before minimizing the criterion.
4. The *ridge* regression is equivalent to the least square estimation under the constraint that the l_2-norm of the vector β is not too large : $\widehat{\beta}_R = \arg\min_\beta \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 \ ; \ \|\beta\|^2 < c \right\}$. The ridge regression keeps all the parameters, but, introducing constraints on the values of the β_j's avoids too large values for the estimated parameters, which reduces the variance.

Choice of the penalty term

- In the next Figure, we see results obtained by the *ridge* method for several values of the tuning parameter $\lambda = l$ on the polynomial regression example.
- Increasing the penalty leads to more regular solutions, the bias increases, and the variance decreases.
- We have overfitting when the penalty is equal to 0 and under-fitting when the penalty is too large.

Choice of the penalty term

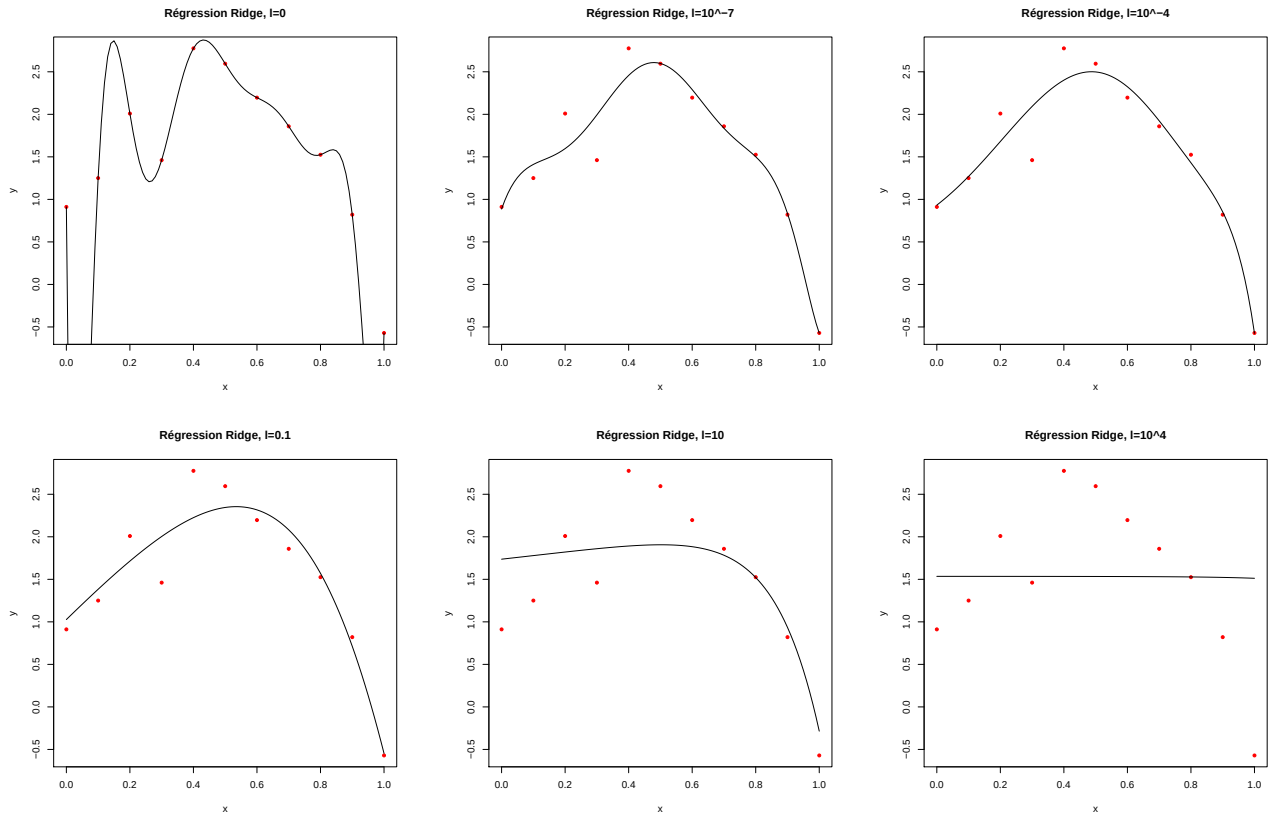
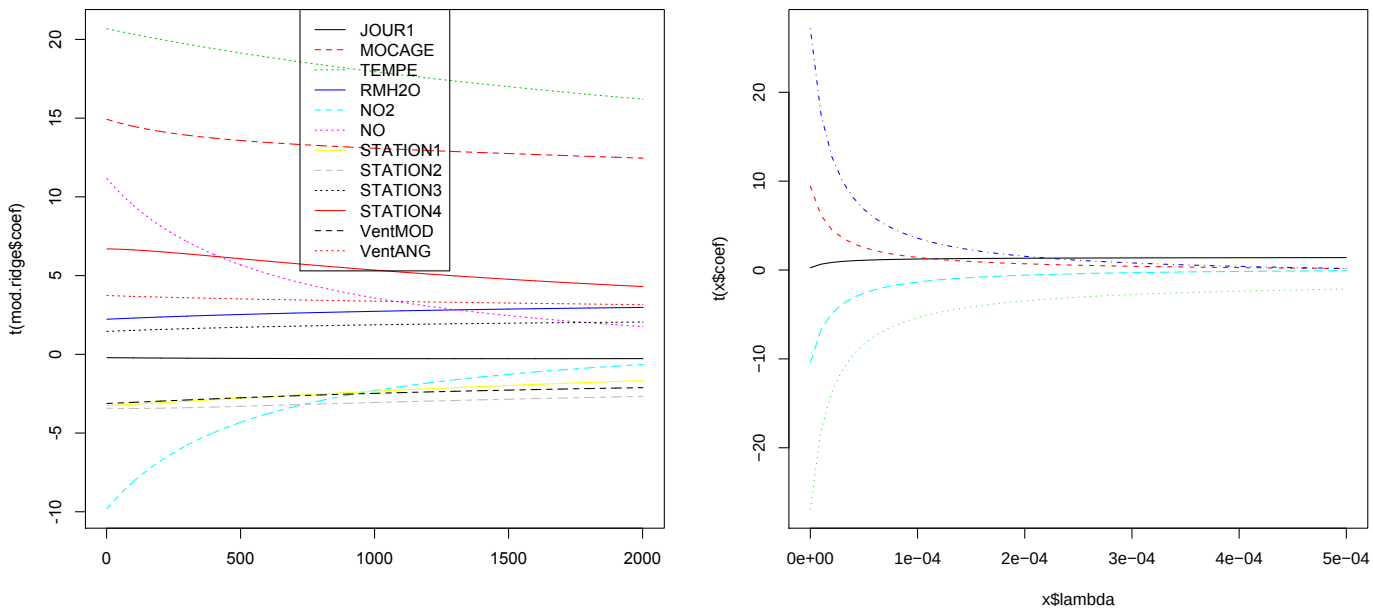


Figure – *Ridge* penalisation for the polynomial model

Choice of the penalty term

- For each regularization method, the choice of the parameter λ is determinant for the model selection. We see in next Figure the *Regularisation path*, showing the profiles of the estimated parameters when the tuning parameter λ increases.



Choice of the regularization parameter

Most softwares use the **cross-validation** to select the tuning parameter penalty. The principe is the following :

- We split the data into K sub-samples. For all I from 1 to K :
  - We compute the Ridge estimator associated to a regularization parameter λ from the data of all the subsamples, except the I-th (that will be a "'test"' sample).
  - We denote by $\hat{\beta}_\lambda^{(-I)}$ the obtained estimator.
  - We test the performances of this estimator on the data that have not been used to build it, that is the one of the I-th sub-sample.
- We compute the criterion :

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\hat{\beta}_\lambda^{(-\tau(i))})^2.$$

- We choose the value of λ which minimizes $CV(\lambda)$.

Application to the Ozone data : The value of λ selected by cross-validation is 5.4. We show the obtained value in the next Figure.

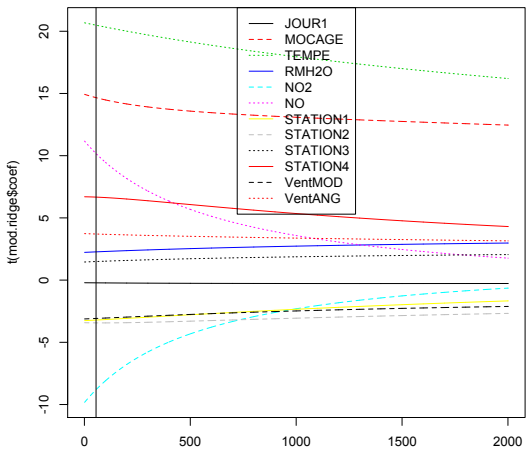


Figure – Selection of the regularization parameter by CV

The LASSO regression

- LASSO is the abbreviation of **Least Absolute Shrinkage and Selection Operator.**
- The Lasso estimator is introduced in the paper by Tibshirani, R. (1996) : Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc B., Vol. 58, No. 1, pages 267-288.
- The Lasso corresponds to the minimization of a least square criterion plus an l_1 penalty term.

Definition

The Lasso estimator of β in the model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, is defined by :

$$\widehat{\beta}_{\text{Lasso}} = \text{argmin}_{\beta \in \mathbb{R}^{p+1}} \left(\sum_{i=1}^{n} (Y_i - \sum_{j=0}^{p} X_i^{(j)} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right),$$

where λ is a nonnegative tuning parameter.

Model and estimation

We can show that this is equivalent to the minimization problem :

$$\widehat{\boldsymbol{\beta}}_L = \text{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p, \|\boldsymbol{\beta}\|_1 \leq t}(\|\mathbf{Y} - \mathbf{X}\beta\|^2),$$

where t is suitably chosen, and $\widehat{\beta}_{0\text{Lasso}} = \bar{Y}$.
Like for the Ridge regression, the parameter λ is a regularization parameter :

- If $\lambda = 0$, we recover the least square estimator.
- If λ tends to infinity, all the coefficients $\hat{\beta}_j$ are equal to 0 for $j = 1, \ldots, p$.

The solution to the Lasso is parsimonious (or sparse), since it has many null coefficients.
It is generally not explicit.

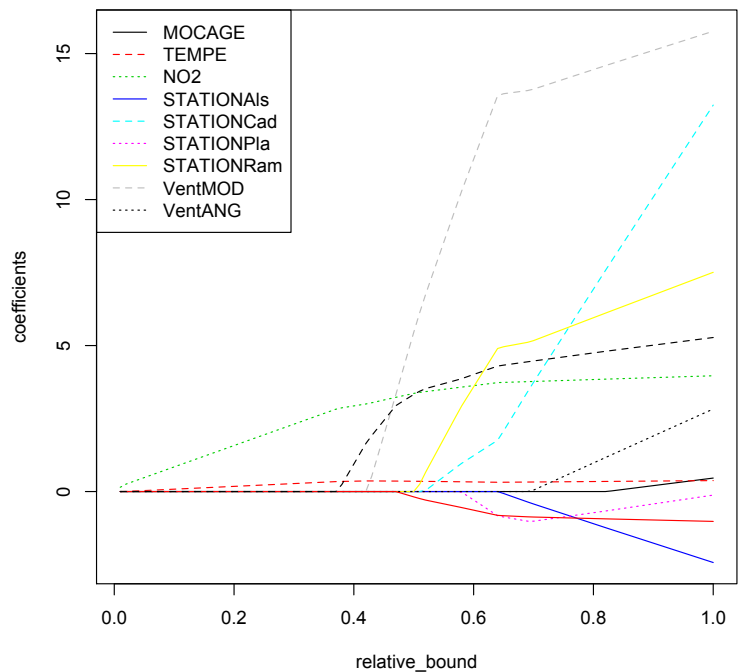
- The LASSO is equivalent to the minimization of the least square criterion under the constraint $\sum_{j=1}^{p} |\beta_j| \leq t$, for some $t > 0$.
- The statistical software R introduces a constraint expressed by a relative bound for $\sum_{j=1}^{p} |\beta_j|$:

$$\sum_{j=1}^{p} |\beta_j| \leq \kappa \sum_{j=1}^{p} |\hat{\beta}_j^{(0)}|,$$

where $\hat{\beta}^{(0)}$ is the least square estimator and $\kappa \in [0, 1]$.
For $\kappa = 1$ we recover the least square estimator and for $\kappa = 0$, all the $\hat{\beta}_j$, $j \geq 1$, vanish.

Applications

We represent in the next Figure the values of the coefficients in function of κ for the Ozone data : this are **the regularization paths of the LASSO**.As for the Ridge regression, the tuning parameter is generally calibrated by cross-validation.



Comparison LASSO/ RIDGE

The next Figure gives a geometric interpretation of the minimization problems for both the Ridge and Lasso estimators. This explains why the Lasso solution is sparse.

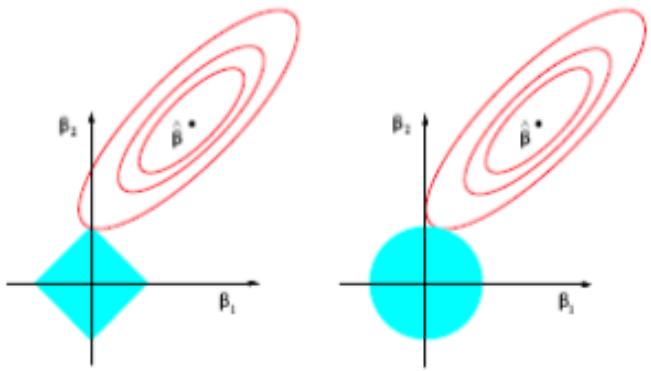


Figure 3.12: *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Conclusion

- Linear models are quite general since they can incorporate new variables defined as functions of the initial variables : X_j^2, $\sin(X_j)$, $\log(X_j)$ ...
- We have seen the importance of model/variable selection to avoid overfitting
- The next step will be to consider linear models for classification (such as logistic regression models, linear SVM) and nonlinear models for regression or classification.

Part I-2 : Classification

- We now consider supervised classification problems. We have a training data set with n observation points (or objects) $\boldsymbol{X}_i$ and their class (or label) Y_i.
- Suppose that $\boldsymbol{d}^n$ corresponds to the observation of a n-sample $\boldsymbol{D}^n = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ with joint unknown distribution P on $\mathcal{X} \times \mathcal{Y}$.
- A *classification rule* is a measurable function $f : \mathcal{X} \to \mathcal{Y}$ that associates the output $f(\boldsymbol{x})$ to the input $\boldsymbol{x} \in \mathcal{X}$.
- In order to quantify the quality of the prevision, we introduce a loss function.

Definition

A measurable function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a *loss function* if $\ell(y, y) = 0$ and $\ell(y, y') > 0$ for $y \neq y'$.

- **For classification** : $\mathcal{Y}$ is a finite set. We define $\ell(y, y') = \mathbb{1}_{y \neq y'}$.
- We consider the expectation of this loss, this leads to the definition of the *risk* :

Definition

Given a loss function ℓ, the *risk* - or *generalisation error* - of a prediction rule f is defined by

$$R_P(f) = \mathbb{E}_{(\boldsymbol{X}, Y) \sim P}[\ell(Y, f(\boldsymbol{X}))].$$

- It is important to note that, in the above definition, $(\boldsymbol{X}, Y)$ is independent of the training sample $\boldsymbol{D}^n$ that was used to build the prediction rule f.

- Let $\mathcal{F}$ denote the set of all possible prediction rules. We say that f^* is an optimal rule if $R_P(f^*) = \inf_{f \in \mathcal{F}} R_P(f)$.
- A natural question arises : is it possible to build optimal rules ?
- We define the Bayes rule, which is an optimal rule for classification.

Definition

We call *Bayes rule* any measurable function f^* in $\mathcal{F}$ such that for all $\boldsymbol{x} \in \mathcal{X}$, $\mathbb{P}(Y = f^*(\boldsymbol{x}) | \boldsymbol{X} = \boldsymbol{x}) = \max_{y \in \mathcal{Y}} \mathbb{P}(Y = y | \boldsymbol{X} = \boldsymbol{x})$.

Theorem

— *If f^* is a Bayes rule, then $R_P(f^*) = \inf_{f \in \mathcal{F}} R_P(f)$.*

- The definition of a Bayes rule depends on the knowledge of the distribution P of $(\boldsymbol{X}, Y)$.
- In practice, we have a training sample $\boldsymbol{D}^n = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ with joint unknown distribution P, and we construct a classification rule.
- The aim is to find a "good" classification rule, in the sense that its risk is close to the optimal risk of a Bayes rule.

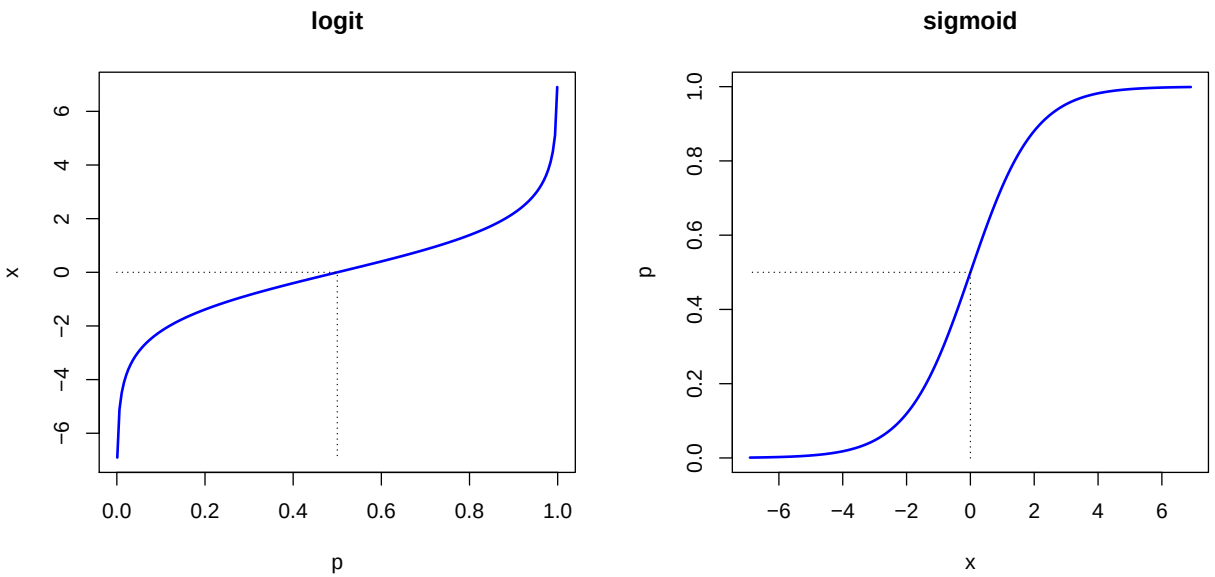
Part I-2

- Logistic Regression
  - Definitions
  - Estimation of the parameters
  - Application
- Roc curves

Logistic regression model

The idea for logistic regression is to use a linear model for probabilities, thanks to a one-to-one mapping ("link" function) from $[0, 1]$ to $\mathbb{R}$.
The most used is the logit function and its inverse, the sigmoid function :

$$\begin{array}{lccc} & [0,1] & & \mathbb{R} \\ \hline \textbf{logit} : & \pi & \rightarrow & \ln\left(\frac{\pi}{1-\pi}\right) \\ & \frac{\exp(x)}{1+\exp(x)} & \leftarrow & x \qquad : \textbf{sigmoid} \end{array}$$



Logistic Regression model

- We assume that $\mathcal{X} = \mathbb{R}^p$.
- One of the most popular model for binary classification when $\mathcal{Y} = \{0, 1\}$ is the **logistic regression model**, for which it is assumed that for all $x \in \mathcal{X}$ and for some $\beta \in \mathbb{R}^p$,

$$\pi(\boldsymbol{x}) = \mathbb{P}(Y = 1/\boldsymbol{X} = \boldsymbol{x}) = \frac{\exp(\langle \beta, \boldsymbol{x} \rangle)}{1 + \exp(\langle \beta, \boldsymbol{x} \rangle)} = \textbf{sigmoid}(\langle \beta, \boldsymbol{x} \rangle)$$

$$1 - \pi(\boldsymbol{x}) = \mathbb{P}(Y = 0/\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{1 + \exp(\langle \beta, \boldsymbol{x} \rangle)},$$

- Setting

$$g(\pi) = \mathit{logit}(\pi) = \ln\left(\frac{\pi}{1-\pi}\right),$$

the logistic regression model corresponds to

$$\mathit{logit}(\pi(\boldsymbol{x})) = \ln\left(\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}\right) = \langle \beta, \boldsymbol{x} \rangle,$$

where $\pi(\boldsymbol{x}) = \mathbb{P}(Y = 1/\boldsymbol{X} = \boldsymbol{x})$.

- g is called the **logit** "link" function.

Estimation of the parameters

- Given a n-sample $\boldsymbol{D}^n = \{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$, we can estimate the parameter β by maximizing the conditional likelihood of $\underline{Y} = (Y_1, \ldots, Y_n)$ given $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$.
- Since the distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$ is a Bernoulli distribution with parameter $\pi_\beta(\boldsymbol{x})$, the conditional likelihood is

$$L(Y_1, \ldots, Y_n, \beta) = \prod_{i=1}^{n} \pi_\beta(\boldsymbol{X_i})^{Y_i} (1 - \pi_\beta(\boldsymbol{X_i}))^{1-Y_i}$$

$$L(\underline{Y}, \beta) = \prod_{i, Y_i=1} \frac{\exp(\langle \beta, \boldsymbol{X_i} \rangle)}{1 + \exp(\langle \beta, \boldsymbol{X_i} \rangle)} \prod_{i, Y_i=0} \frac{1}{1 + \exp(\langle \beta, \boldsymbol{X_i} \rangle)}.$$

Estimation of the parameters

- Unlike the linear model, there is no explicit expression for the maximum likelihood estimator $\hat{\beta}$.
- It can be shown that computing $\hat{\beta}$ is a convex optimization problem.
- We compute the gradient of the log-likelihood, also called **the score function** $S(\underline{Y}, \beta)$ and use a **Newton-Raphson algorithm** to approximate $\hat{\beta}$ satisfying $S(\underline{Y}, \hat{\beta}) = 0$.
- Variable selection is also possible by maximizing the penalized likelihood (AIC, BIC, LASSO ..).

- We can then predict the probabilities :

$$\hat{\mathbb{P}}(Y = 1/\boldsymbol{X} = \boldsymbol{x})) = \pi_{\hat{\beta}}(\boldsymbol{x}) = \frac{\exp(\langle \hat{\beta}, \boldsymbol{x} \rangle)}{1 + \exp(\langle \hat{\beta}, \boldsymbol{x} \rangle)}$$

$$\hat{\mathbb{P}}(Y = 0/\boldsymbol{X} = \boldsymbol{x})) = 1 - \pi_{\hat{\beta}}(\boldsymbol{x}) = \frac{1}{1 + \exp(\langle \hat{\beta}, \boldsymbol{x} \rangle)}.$$

- We then compute the logistic regression classifier : we set $\hat{Y}(\boldsymbol{x}) = 1$ if $\hat{\mathbb{P}}(Y = 1/\boldsymbol{X} = \boldsymbol{x})) \geq \hat{\mathbb{P}}(Y = 0/\boldsymbol{X} = \boldsymbol{x})$ which is equivalent to $\langle \hat{\beta}, \boldsymbol{x} \rangle \geq 0$. Hence,

$$\hat{Y}(\boldsymbol{x}) = \mathbb{1}_{\langle \hat{\beta}, \boldsymbol{x} \rangle \geq 0}.$$

Illustration in 1D

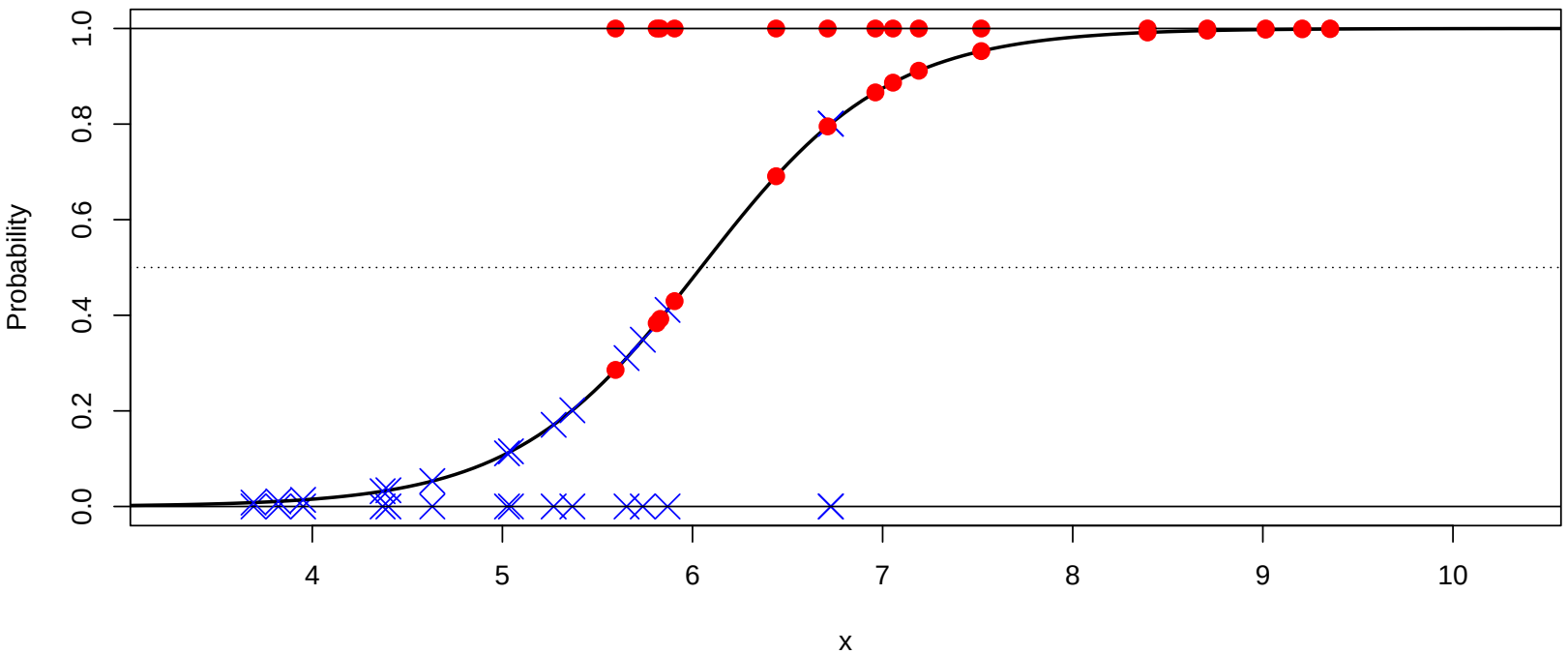


Figure – Logistic regression for a dataset composed of 2 groups of size 15, sampled from Normal distributions, centered at 5 and 7, with variance 1.

Application

- We use the logistic regression model to predict the exceedance of the threshold 150 for the variable O3obs.
- Only with the variable MOCAGE :

```
> logistic=glm(depseuil ~ MOCAGE,
data=ozone,family=binomial(link = "logit"))
> summary(logistic)
```

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.596493	0.389841	-14.36	<2e-16 ***
MOCAGE	0.028659	0.002528	11.34	<2e-16 ***

Application

- We compute the predicted values :

```
> pihat=logistic$fitted.values
> Yhat=(pihat>0.5)
> table(depseuil,Yhat)
```

$Y \setminus \hat{Y}$	0	1
0	830	33
1	152	26

- The misclassification error is 17.7%. There are many false negative .
- The model tends to underestimate the threshold overflow : only 15% of the overflows have been predicted.
- We try to improve the model by considering more variables.

Application

- We consider the variables JOUR, MOCAGE,TEMPE, RMH2O, NO2, NO

```
> logistic2=glm(depseuil ~ MOCAGE+TEMPE+RMH2O+NO2+NO+JOUR,
data=ozone,family=binomial(link = "logit"))
> summary(logistic2)
```

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-14.840457	1.116901	-13.287	< 2e-16 ***
MOCAGE	0.026924	0.004045	6.655	2.82e-11 ***
TEMPE	0.309566	0.029529	10.483	< 2e-16 ***
RMH2O	138.430723	28.548702	4.849	1.24e-06 ***
NO2	-0.210011	0.102607	-2.047	0.0407 *
NO	0.742302	0.552606	1.343	0.1792
JOUR1	0.159047	0.235654	0.675	0.4997

Application

- We compute the predicted values :

```
> pihat=logistic2$fitted.values
> Yhat=(pihat>0.5)
> table(depseuil,Yhat)
```

$Y \setminus \hat{Y}$	0	1
0	829	34
1	88	90

- The misclassification error is 11.7%.
- We have improved the results, but there are still many false negative : only 50% of the overflows have been predicted.

Part I-2

- Logistic Regression
  - Definitions
  - Estimation of the parameters
  - Application
- Roc curves

Two-classes problem : ROC curve

Motivation

For two classes $\mathcal{Y} = \{0, 1\}$, the optimal Bayes rule is :

$$\mathbb{P}(Y = 1 | \boldsymbol{X} = \boldsymbol{x}) > \frac{1}{2} \quad \Leftrightarrow \quad \boldsymbol{x} \text{ belongs to class 1}$$

This gives a symmetric role to classes 0 and 1, which is often not desirable (health context, for instance).

The idea is to parameterize the decision by a new threshold parameter s :

$$\mathbb{P}(Y = 1 | \boldsymbol{X} = \boldsymbol{x}) > s \quad \Leftrightarrow \quad \boldsymbol{x} \text{ belongs to class 1}$$

s should be chosen according to policy decision, typically a tradeoff between the rate of true positive and false positive.

Two-classes problem : ROC curve

Motivation

By analogy with the first and second kind errors for testing procedures, we introduce

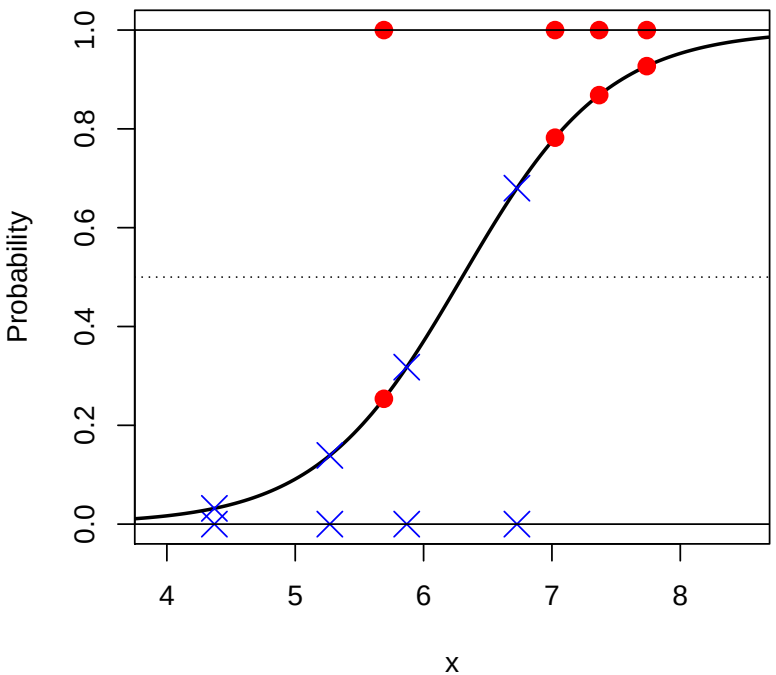
- The False Positive Rate :

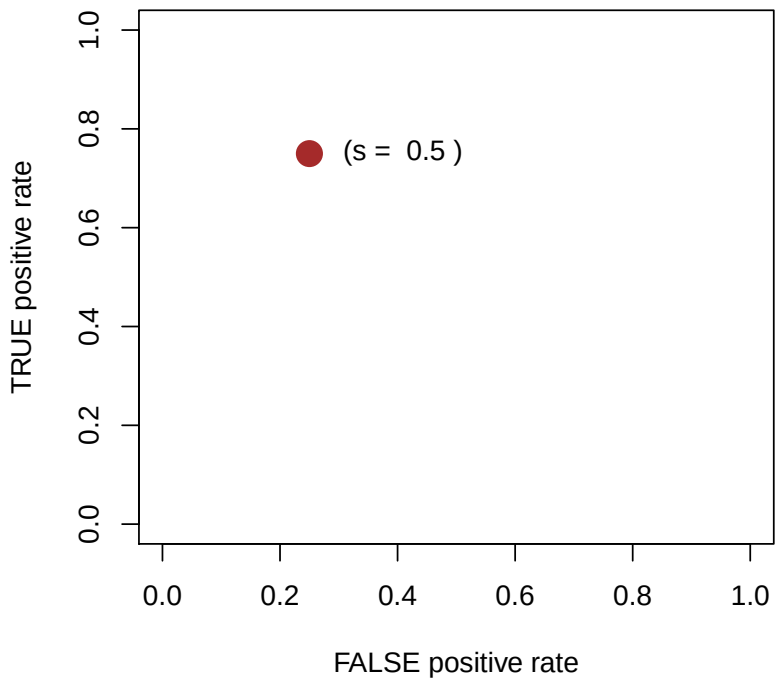
$$FPR = \frac{\sharp\left\{i, \hat{Y}_i = 1, Y_i = 0\right\}}{\sharp\left\{i, Y_i = 0\right\}}.$$

- The True Positive Rate :

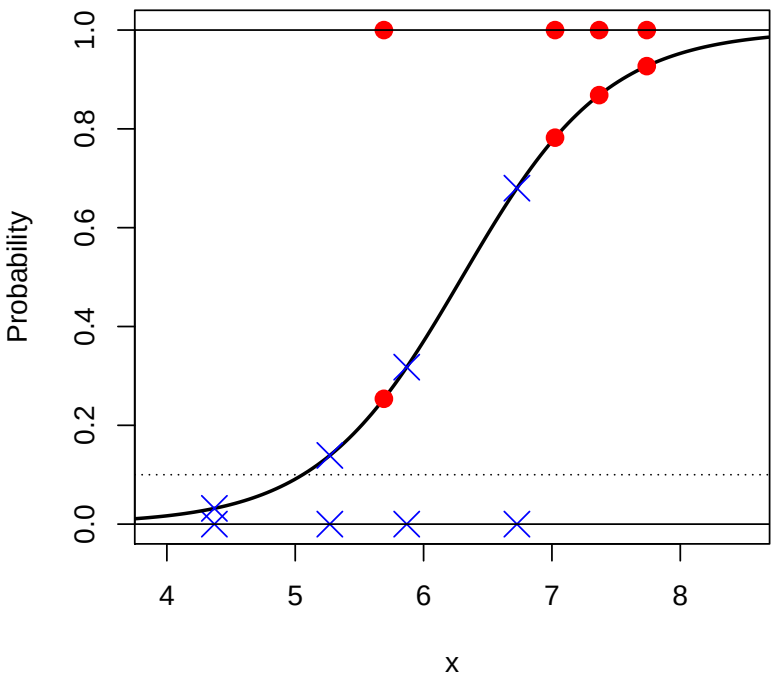
$$TPR = \frac{\sharp\left\{i, \hat{Y}_i = 1, Y_i = 1\right\}}{\sharp\left\{i, Y_i = 1\right\}}.$$

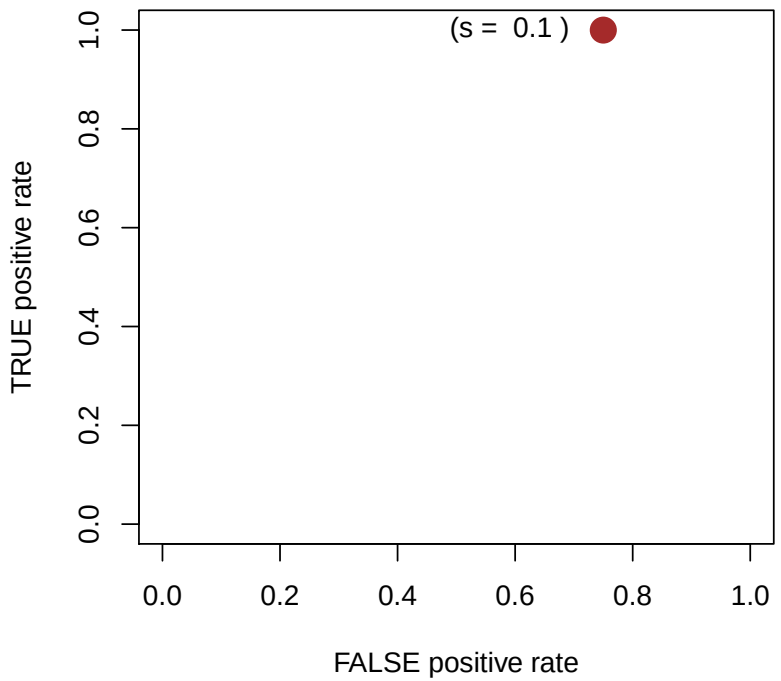
ROC curve - Illustration in 1D



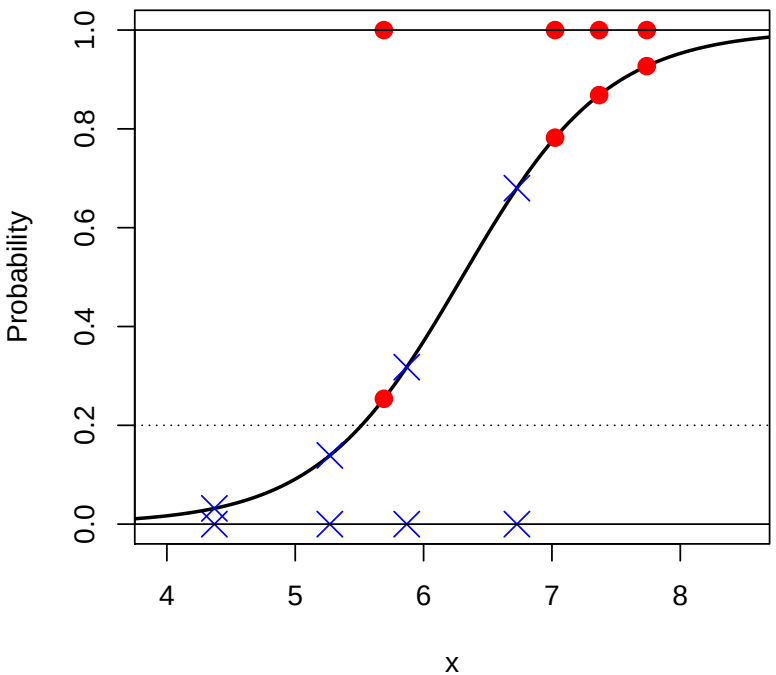


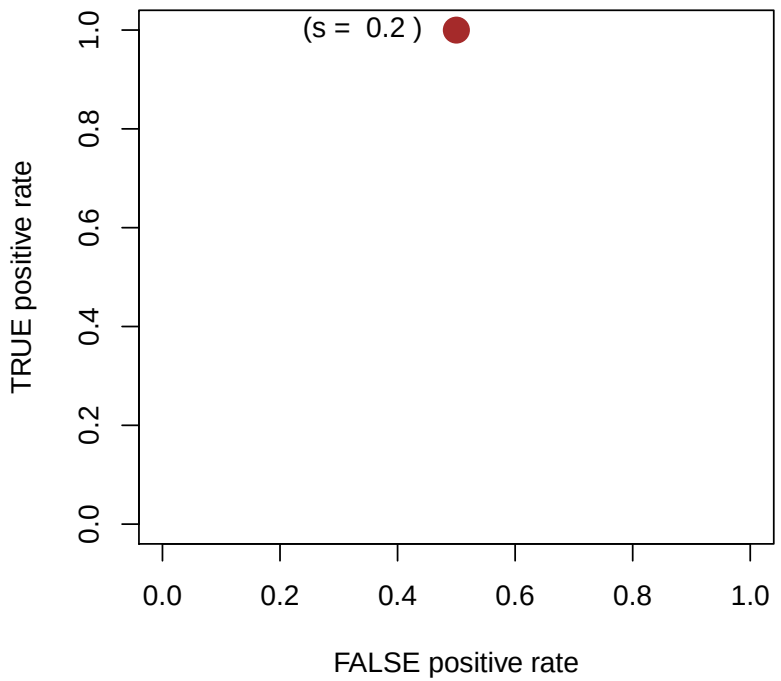
ROC curve - Illustration in 1D



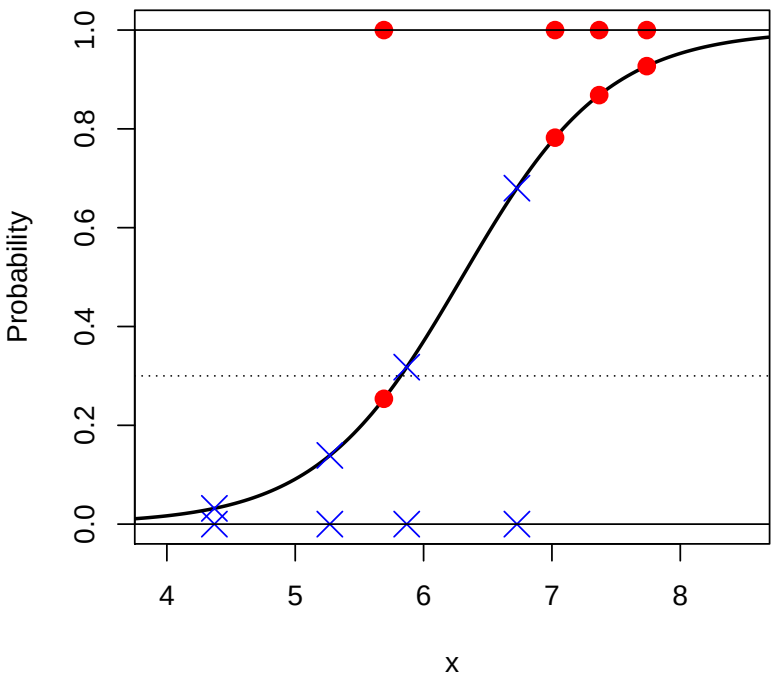


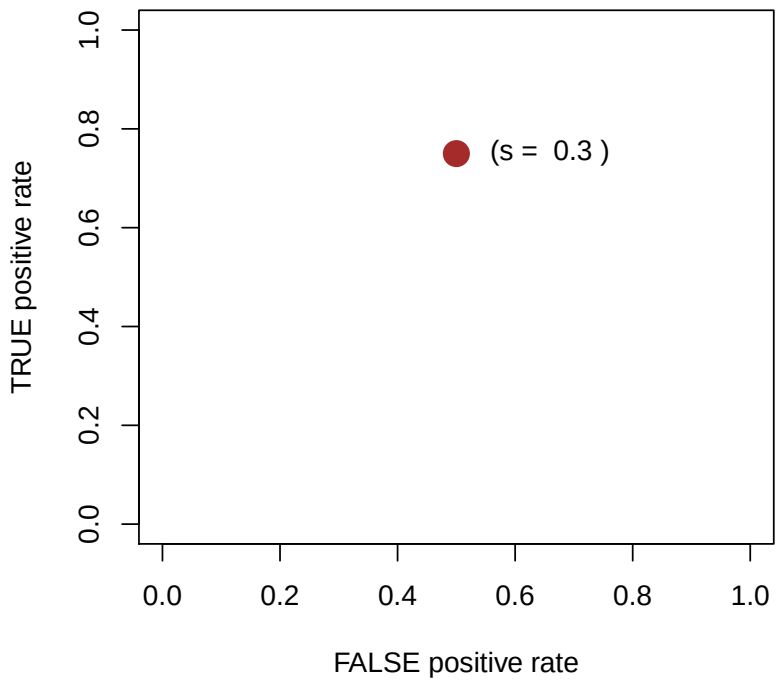
ROC curve - Illustration in 1D



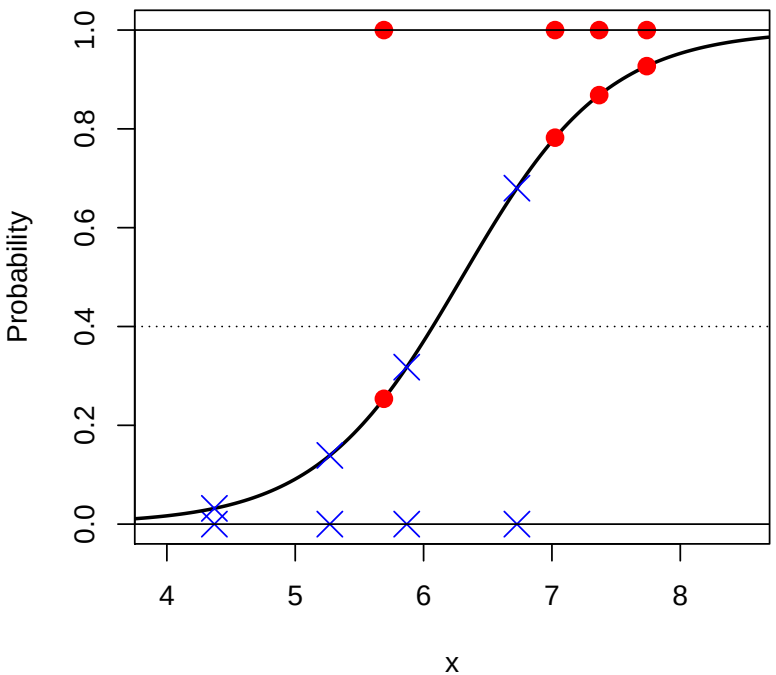


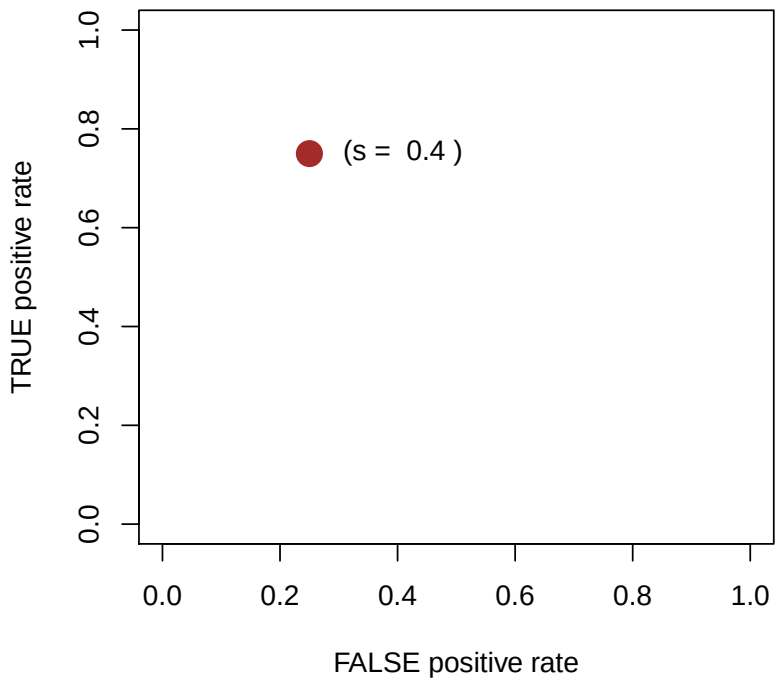
ROC curve - Illustration in 1D



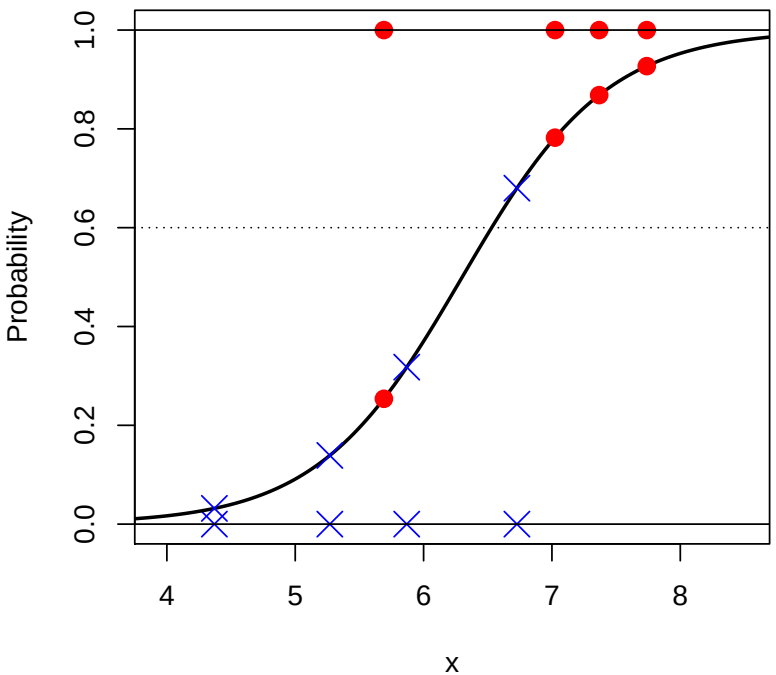


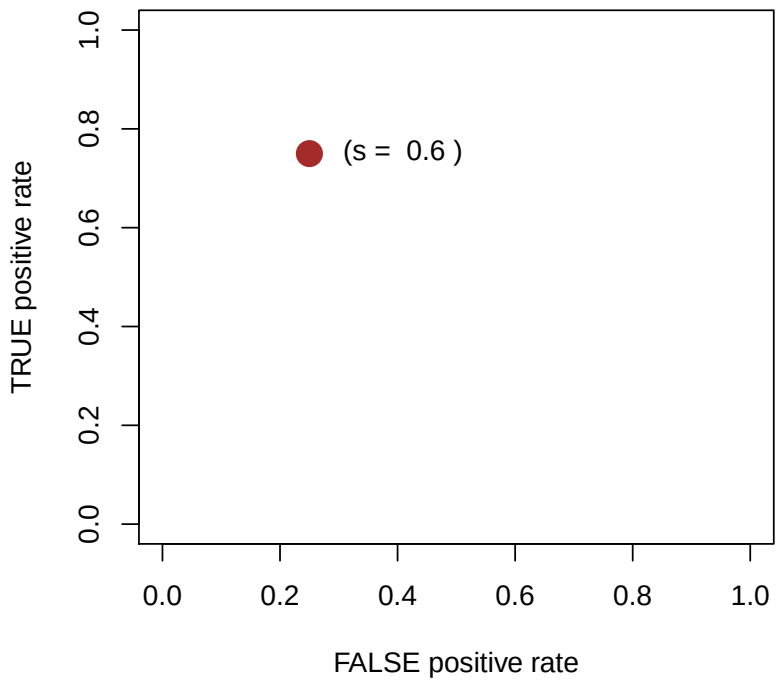
ROC curve - Illustration in 1D



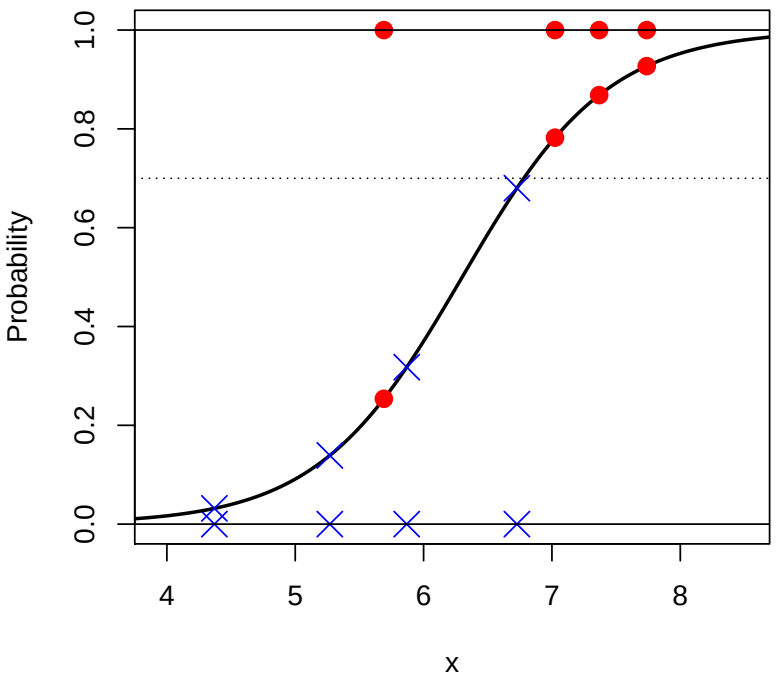


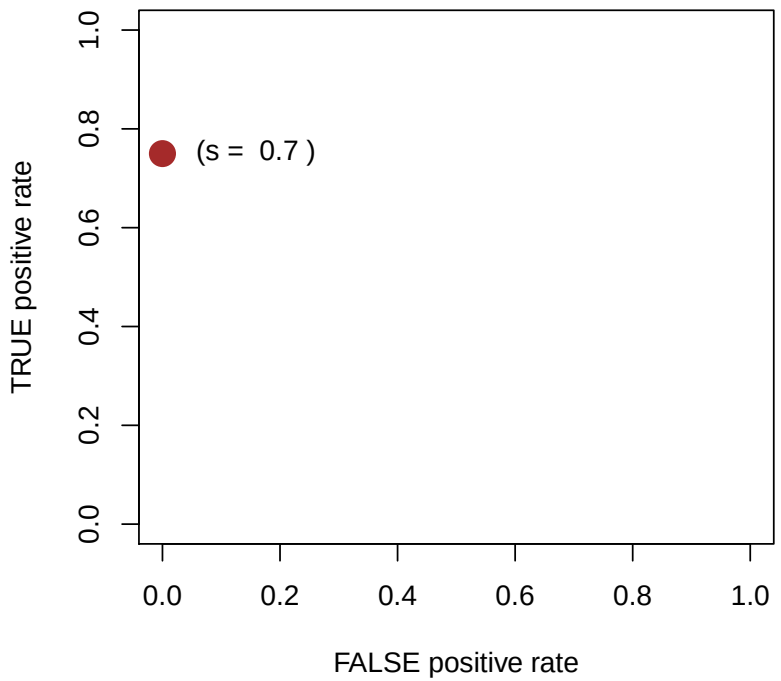
ROC curve - Illustration in 1D



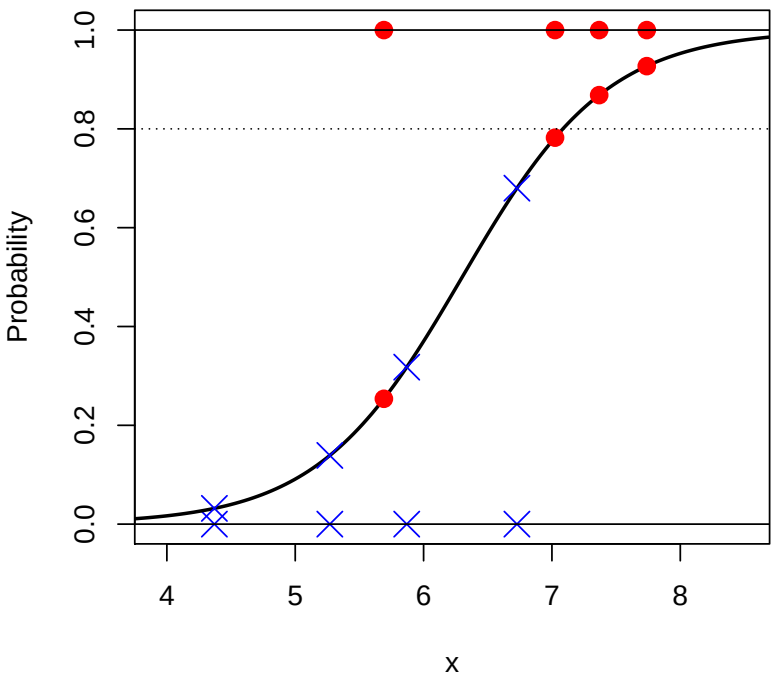


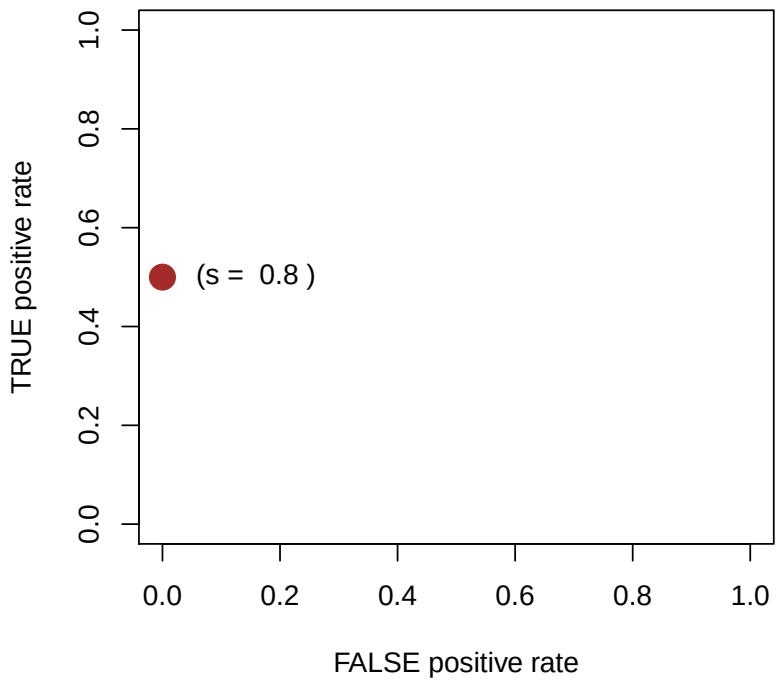
ROC curve - Illustration in 1D



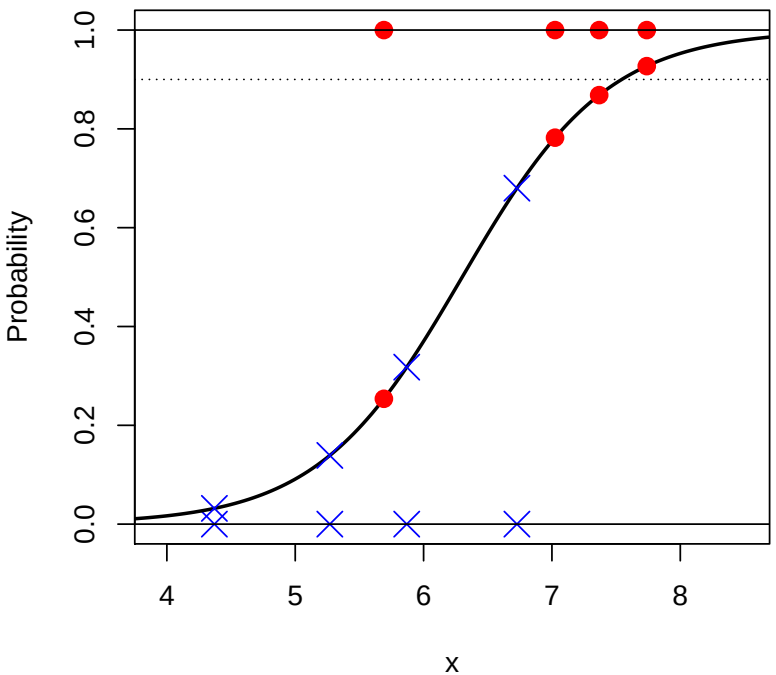


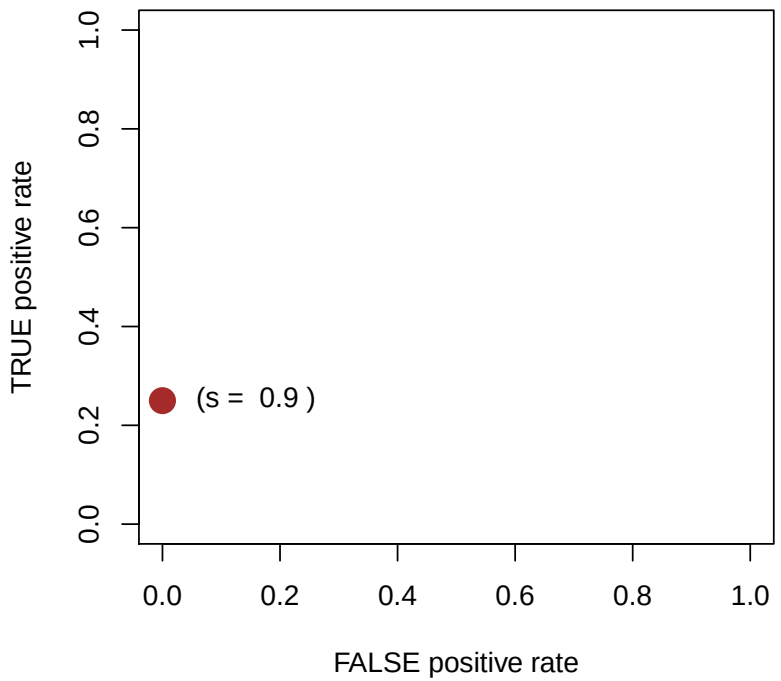
ROC curve - Illustration in 1D



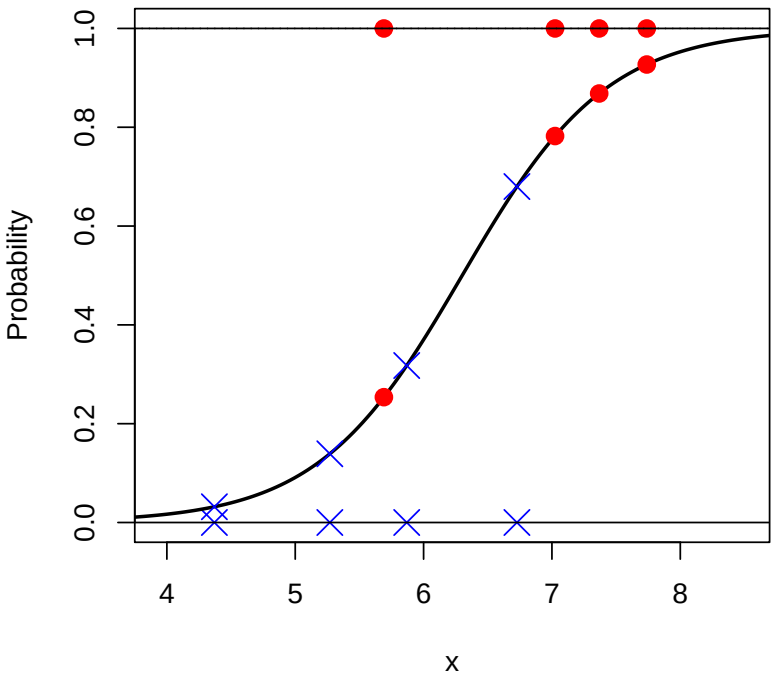


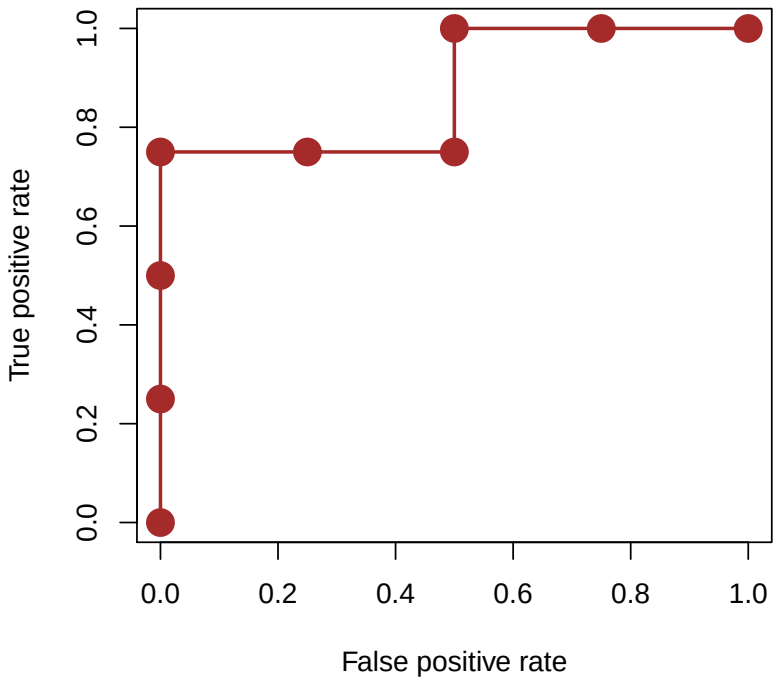
ROC curve - Illustration in 1D





ROC curve - Illustration in 1D





ROC curve - Definition

Definitions from the contingency table

Prediction : if $\widehat{\pi}_i > s$, $\widehat{Y}_i = 1$ else $\widehat{Y}_i = 0$

Prediction	Observation		Total
	$Y_i = 1$	$Y_i = 0$	
$\widehat{Y}_i = 1$	$n_{11}(s)$	$n_{10}(s)$	$n_{1+}(s)$
$\widehat{Y}_i = 0$	$n_{01}(s)$	$n_{00}(s)$	$n_{0+}(s)$
Total	n_{+1}	n_{+0}	n

- True positive rate : $TPR(s) = \frac{n_{11}(s)}{n_{+1}}$ *(sensitivity, recall)*
- False positive rate : $FPR(s) = \frac{n_{10}(s)}{n_{+0}}$

The ROC curve plots TPR(s) versus FPR(s) for all values of $s \in [0, 1]$.

Usage of ROC curve to select classifiers

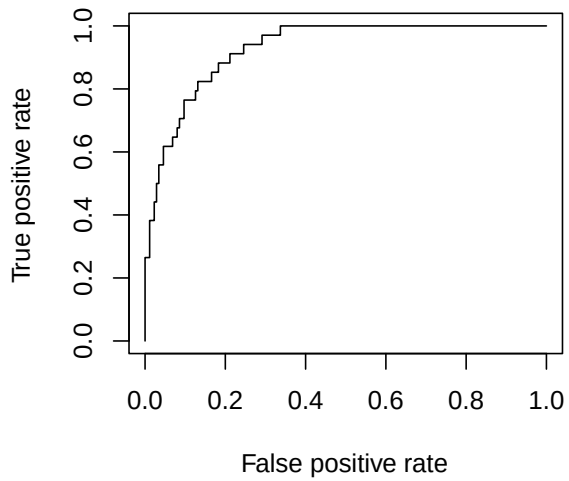


Figure – Ozone : ROC curve for logistic regression.

The Roc curve should be computed on a test sample.
The "ideal" Roc curve corresponds to FPR=0 and TPR =1 (no error of classification).
The AUC : Area Under the Curve can be a criterion to choose among several classification rules.

Part I-3 :

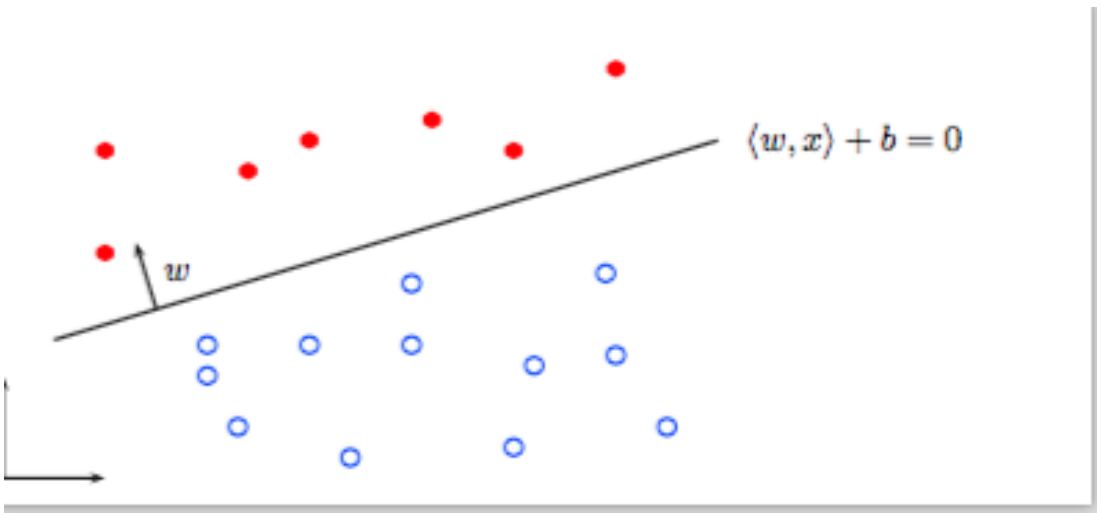
- Support Vector Machines.
  - Linear SVM in the separable case
  - Linear SVM in the non separable case
  - Non linear SVM and kernels
  - Conclusion

Linear Support Vector Machine

Definition

The training set $d_1^n = (x_1, y_1), \ldots, (x_n, y_n)$ is called **linearly separable** if there exists (w, b) such that for all i,
$y_i = 1$ if $\langle w, x_i \rangle + b > 0$, $y_i = -1$ if $\langle w, x_i \rangle + b < 0$,
which means that $\forall i \; y_i \left(\langle w, x_i \rangle + b \right) > 0$.

The equation $\langle w, x \rangle + b = 0$ defines a separating hyperplane with orthogonal vector w.

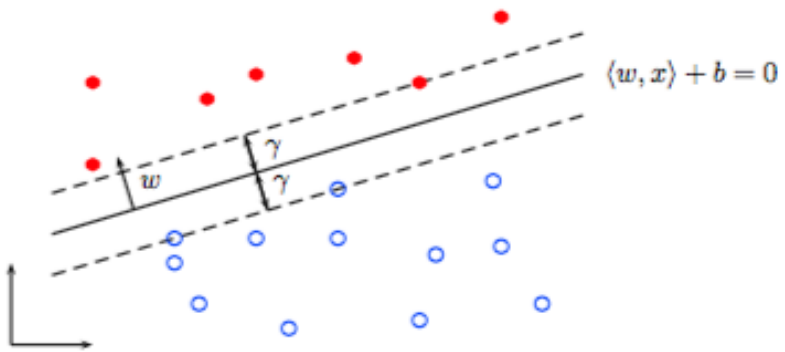


- The function $f_{w,b}(x) = \mathbb{1}_{\langle w,x\rangle+b\geq 0} - \mathbb{1}_{\langle w,x\rangle+b<0}$ defines a possible linear classification rule.

- The problem is that there exists an infinity of separating hyperplanes, and therefore an infinity of classification rules.

- Which one should we choose ? The response is given by Vapnik (1999).

- The classification rule with the best generalization properties corresponds to the separating hyperplane maximizing the margin γ between the two classes on the training set.



- If we consider two entries of the training set, that are on the border defining the margin, and that we call x_1 and x_{-1} with respective outputs 1 and -1, the separating hyperplane is located at the half-distance between x_1 and x_{-1}.

- The margin is therefore equal to the half of the distance between x_1 and x_{-1} projected onto the normal vector of the separating hyperplane :

$$\gamma = \frac{1}{2} \frac{\langle w, x_1 - x_{-1} \rangle}{\|w\|}.$$

Definition

The hyperplane $\langle w, x \rangle + b = 0$ is **canonical** with respect to the set of vectors $x_1, \ldots, x_k$ if

$$\min_{i=1\ldots k} |\langle w, x_i \rangle + b| = 1.$$

- The separating hyperplane has the canonical form relatively to the vectors $\{x_1, x_{-1}\}$ if it is defined by (w, b) where $\langle w, x_1 \rangle + b = 1$ and $\langle w, x_{-1} \rangle + b = -1$. In this case, we have $\langle w, x_1 - x_{-1} \rangle = 2$, hence

$$\gamma = \frac{1}{\|w\|}.$$

- Finding the separating hyperplane with maximal margin consists in finding (w, b) such that

$$\begin{gathered} \|w\|^2 \text{ or } \tfrac{1}{2}\|w\|^2 \text{ is minimal} \\ \text{under the constraint} \\ y_i \left(\langle w, x_i \rangle + b \right) \geq 1 \text{ for all } i. \end{gathered}$$

This leads to a convex optimization problem with linear constraints, hence there exists a unique global minimizer.

The primal problem to solve is :

$$\text{Minimizing } \frac{1}{2}\|w\|^2 \text{ s. t. } y_i\left(\langle w, x_i\rangle + b\right) \geq 1 \ \forall \ i.$$

Lagrangian $L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i\left(\langle w, x_i\rangle + b\right) - 1\right).$

Dual Function :

$$\begin{aligned}
\frac{\partial L}{\partial w}(w, b, \alpha) &= w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Leftrightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \\
\frac{\partial L}{\partial b}(w, b, \alpha) &= -\sum_{i=1}^{n} \alpha_i y_i = 0 \Leftrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}$$

$$\begin{aligned}
\theta(\alpha) &= \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle .
\end{aligned}$$

The corresponding **dual problem** is :
Maximizing

$$\theta(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

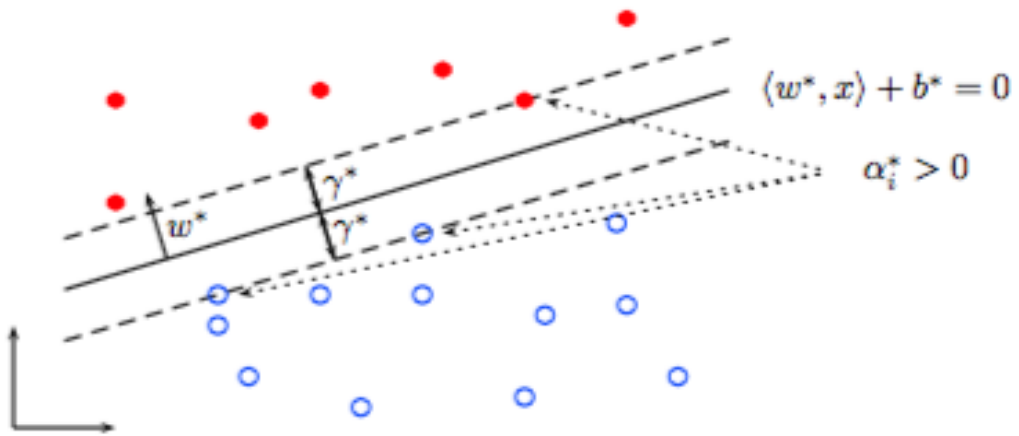
under the constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0 \ \forall i$.
The solution α^* of the dual problem can be obtained with classical optimization softwares.
Remark : The solution does not depend on the dimension d, but depends on the sample size n, hence it is interesting to notice that when $\mathcal{X}$ is high dimensional, linear SVM do not suffer from the curse of dimensionality.
For big data sets, n is very large, it is preferable to solve the primal problem.

Supports Vectors

- For our optimization problem, the **Karush-Kuhn-Tucker conditions** are
  - $\alpha_i^* \geq 0 \ \forall i = 1 \dots n$.
  - $y_i \left(\langle w^*, x_i \rangle + b^* \right) \geq 1 \ \forall i = 1 \dots n$.
  - $\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 \right) = 0 \ \forall \ i = 1 \dots n$. (complementary condition)
- Only the $\alpha_i^* > 0$ are involved in the resolution of the optimization problem.
- If the number of values $\alpha_i^* > 0$ is small, the solution of the dual problem is called **sparse**.

Definition

The x_i such that $\alpha_i^* > 0$ are called the **support vectors**. They are located on the border defining the maximal margin namely $y_i \left(\langle w^*, x_i \rangle + b^* \right) = 1$ (c.f. complementary KKT condition).



We finally obtain the following classification rule :

$$\hat{f}(x) = \mathbb{1}_{\langle w^*, x\rangle + b^* \geq 0} - \mathbb{1}_{\langle w^*, x\rangle + b^* < 0},$$

with

- $w^* = \sum_{i=1}^n \alpha_i^* x_i y_i$,
- $b^* = 1 - \min_{y_i=1} \langle w^*, x_i \rangle$.

The maximal margin equals $\gamma^* = \frac{1}{\|w^*\|} = \left(\sum_{i=1}^n (\alpha_i^*)^2\right)^{-1/2}$.
The α_i^* that do not correspond to support vectors (sv) are equal to 0, and therefore

$$\hat{f}(x) = \mathbb{1}_{\sum_{x_i \ sv} y_i \alpha_i^* \langle x_i, x\rangle + b^* \geq 0} - \mathbb{1}_{\sum_{x_i \ sv} y_i \alpha_i^* \langle x_i, x\rangle + b^* < 0}.$$

Linear SVM in the non separable case

- The previous method cannot be applied when the training set is not linearly separable. Moreover, the method is very sensitive to outliers.
- In the general case, we allow some points to be in the margin and even on the wrong side of the margin.
- We introduce the slack variable $\xi = (\xi_1, \ldots, \xi_n)$ and the constraint $y_i(\langle w, x_i \rangle + b) \geq 1$ becomes

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \text{ with } \xi_i \geq 0.$$

  - If $\xi_i \in [0, 1]$ the point is well classified but in the region defined by the margin.
  - If $\xi_i > 1$ the point is misclassified.
- The margin is called **flexible margin**.

Optimization problem with relaxed constraints

- In order to avoid too large margins, we penalize large values for the slack variable ξ_i.
- The **primal optimization problem** is formalized as follows :

Minimize with respect to (w, b, ξ) $\quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$
such that

$$y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i \ \forall \ i$$
$$\xi_i \geq 0$$

Remarks :

- $C > 0$ is a tuning parameter of the SVM algorithm. It will determine the tolerance to misclassifications.
- If C increases, the number of misclassified points decreases, and if C decreases, the number of misclassified points increases. C is generally calibrated by cross-validation.

The **Lagrangian** of this problem is :

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) &= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{n} \xi_i (C - \alpha_i - \beta_i) \\ &\quad + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \left(\langle w, x_i \rangle + b\right), \end{aligned}$$

with $\alpha_i \geq 0$ and $\beta_i \geq 0$.

The cancellation of the partial derivatives $\frac{\partial L}{\partial w}(w, b, \xi, \alpha, \beta)$, $\frac{\partial L}{\partial b}(w, b, \xi, \alpha, \beta)$ and $\frac{\partial L}{\partial \xi_i}(w, b, \xi, \alpha, \beta)$ leads to the following dual problem.

Dual problem :

Maximizing $\theta(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$

s. t. $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C \ \forall i$.

Karush-Kuhn-Tucker conditions :

- $0 \leq \alpha_i^* \leq C \ \forall i = 1 \dots n$.
- $y_i \left(\langle w^*, x_i \rangle + b^* \right) \geq 1 - \xi_i^* \ \forall i = 1 \dots n$.
- $\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) + \xi_i^* - 1 \right) = 0 \ \forall \ i = 1 \dots n$.
- $\xi_i^* (\alpha_i^* - C) = 0$.

Supports vectors

We have the complementary Karush-Kuhn-Tucker conditions :

$$\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) + \xi_i^* - 1 \right) = 0 \ \forall \ i = 1 \dots n,$$
$$\xi_i^* (\alpha_i^* - C) = 0$$

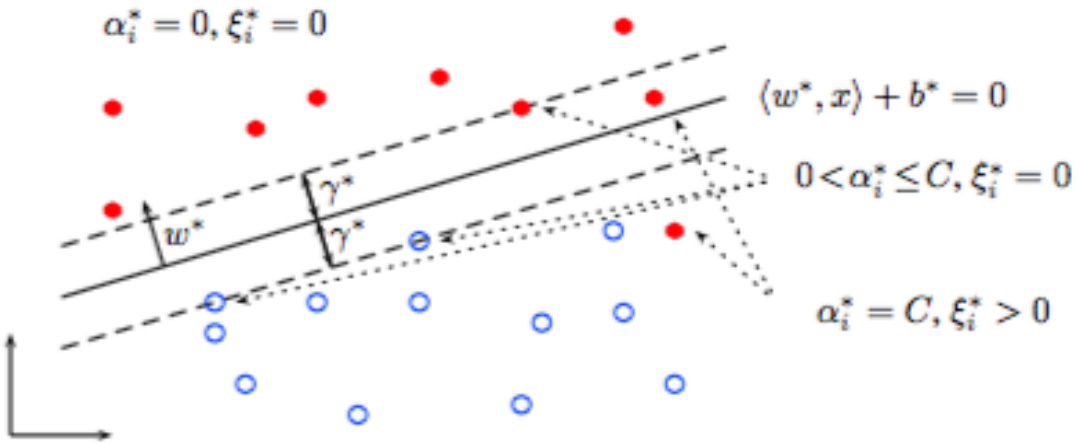
Definition

The points x_i such that $\alpha_i^* > 0$ are the **support vectors**.

We have two types of support vectors :

- The support vectors for which the slack variables are equal to 0. They are located on the border of the region defining the margin.
- The support vectors for which the slack variables are not equal to 0 : $\xi_i^* > 0$ and in this case $\alpha_i^* = C$.

For the vectors that are not support vectors, we have $\alpha_i^* = 0$ and $\xi_i^* = 0$.



The classification rule is defined by

$$\begin{aligned}\hat{f}(x) &= \mathbb{1}_{\langle w^*, x\rangle + b^* \geq 0} - \mathbb{1}_{\langle w^*, x\rangle + b^* < 0}, \\ &= \text{sign}\left(\langle w^*, x\rangle + b^*\right)\end{aligned}$$

with

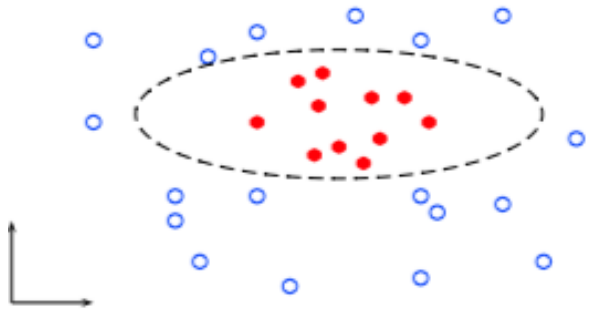
- $w^* = \sum_{i=1}^n \alpha_i^* x_i y_i$,
- b^* such that $y_i \left(\langle w^*, x_i\rangle + b^*\right) = 1 \ \forall x_i,\ 0 < \alpha_i^* < C$.

The maximal margin equals $\gamma^* = \frac{1}{\|w^*\|} = \left(\sum_{i=1}^n (\alpha_i^*)^2\right)^{-1/2}$.
The α_i^* that do not correspond to support vectors are equal to 0, hence

$$\hat{f}(x) = \mathbb{1}_{\sum_{x_i sv} y_i \alpha_i^* \langle x_i, x\rangle + b^* \geq 0} - \mathbb{1}_{\sum_{x_i sc} y_i \alpha_i^* \langle x_i, x\rangle + b^* < 0}.$$

Non linear SVM and kernels

A training set is rarely linearly separable and linear SVM are not appropriate in this case.



- The solution is to enlarge the feature space and send the entries in an Hilbert space $\mathcal{H}$, with high or possibly infinite dimension, via a function ϕ, and to apply a linear SVM procedure on the new training set $\{(\phi(x_i), y_i), i = 1 \dots n\}$. The space $\mathcal{H}$ is called the **feature space**. This idea is due to Boser, Guyon, Vapnik (1992).
- In the previous example, setting $\phi(x) = (x_1^2, x_2^2, x_1, x_2)$, the training set becomes linearly separable in $\mathbb{R}^4$.

The kernel trick

- A natural question arises : how can we choose $\mathcal{H}$ and ϕ ? In fact, we do not choose $\mathcal{H}$ and ϕ but a *kernel* .
- The classification rule is

$$\hat{f}(x) = \mathbb{1}_{\sum y_i \alpha_i^* \langle \phi(x_i), \phi(x) \rangle + b^* \geq 0} - \mathbb{1}_{\sum y_i \alpha_i^* \langle \phi(x_i), \phi(x) \rangle + b^* < 0},$$

  where the α_i^*'s are the solutions of the dual problem in the feature space $\mathcal{H}$:
- Maximizing $\theta(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle$ s. t. $\sum_{i=1}^n \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C \ \forall i$.

- It is important to notice that the final classification rule in the feature space depends on ϕ only through scalar products of the form $\langle \phi(x_i), \phi(x) \rangle$ or $\langle \phi(x_i), \phi(x_j) \rangle$.

- The only knowledge of the function k defined by $k(x, x') = \langle \phi(x), \phi(x') \rangle$ allows to define the SVM in the feature space $\mathcal{H}$ and to derive a classification rule in the space $\mathcal{X}$. The explicite computation of ϕ is not required.

Definition

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$ for a given function $\phi : \mathcal{X} \to \mathcal{H}$ is called a **kernel**.

- A kernel is generally more easy to compute than the function ϕ that returns values in a high dimensional space (or infinite dimensional space).
- Let us now give a property to ensure that a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defines a kernel.

Proposition

—**Mercer condition** *If the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is continuous, symmetric, and if for all finite subset $\{x_1, \ldots, x_n\}$ in $\mathcal{X}$, the matrix $(k(x_i, x_j))_{1 \leq i,j \leq n}$ is positive definite :*

$$\forall c_1, \ldots, c_n \in \mathbb{R}, \sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0,$$

then, there exists an Hilbert space $\mathcal{H}$ and a function $\phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$. The space $\mathcal{H}$ is called the **Reproducing kernel Hilbert Space (RKHS)** *associated to k.*
We have :

1. *For all $x \in \mathcal{X}$, $k(x, .) \in \mathcal{H}$ where $k(x, .) : y \mapsto k(x, y)$.*
2. **Reproducing property** *:*

$$h(x) = \langle h, k(x, .) \rangle_{\mathcal{H}} \text{ for all } x \in \mathcal{X} \text{ and } h \in \mathcal{H}.$$

- Let us give some examples. The Mercer condition is often hard to verify but we know some classical examples of kernels that can be used.
- We assume that $\mathcal{X} = \mathbb{R}^d$.
  - p **degree polynomial kernel** : $k(x, x') = (1 + \langle x, x' \rangle)^p$
  - **Gaussian kernel (RBF)** : $k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$
    ϕ returns values in a infinite dimensional space.
  - **Laplacian kernel** : $k(x, x') = e^{-\frac{\|x-x'\|}{\sigma}}$.
  - **Sigmoid kernel** : $k(x, x') = \tanh(\kappa\langle x, x' \rangle + \theta)$ (this kernel is not positive definite).

- We have seen some examples of kernels. One can construct new kernels by aggregating several kernels.
- For example let k_1 and k_2 be two kernels and f a function $\mathbb{R}^d \to \mathbb{R}$, $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$, B a positive definite matrix, P a polynomial with positive coefficients and $\lambda > 0$.
  The functions defined by $k(x, x') = k_1(x, x') + k_2(x, x')$, $\lambda k_1(x, x')$, $k_1(x, x')k_2(x, x')$, $f(x)f(x')$, $k_1(\phi(x), \phi(x'))$, $x^T B x'$, $P(k_1(x, x'))$, or $e^{k_1(x,x')}$ are still kernels.
- We have presented examples of kernels for the case where $\mathcal{X} = \mathbb{R}^d$ but a very interesting property is that kernels can be defined for very general input spaces, such as **sets, trees, graphs, texts, DNA sequences ...**

Conclusion

- Using kernels allows to delinearize classification algorithms by mapping $\mathcal{X}$ in the RKHS $\mathcal{H}$ with the map $x \mapsto k(x, .)$. It provides nonlinear algorithms with almost the same computational properties as linear ones.
- SVM have nice theoretical properties, cf. Vapnik's theory for empirical risk minimization.
- The use of RKHS allows to apply to any set $\mathcal{X}$ (such as set of graphs, texts, DNA sequences ..) algorithms that are defined for vectors as soon as we can define a kernel $k(x, y)$ corresponding to some measure of similarity between two objects of $\mathcal{X}$.

Conclusion

- Important issues concern the choice of the kernel, and of the tuning parameters to define the SVM procedure.
- Note that SVM can also be used for multi-class classification problems for example, one can built a SVM classifier for each pair of classes and predict the class for a new point by a majority vote.
- Kernels are also used for regression as mentioned above or for non supervised classification (kernel PCA).

References

- Cristianini N. and Shawe-TaylorJ. (2000) *An introduction to Support Vector Machines* Cambridge University Press.
- 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.
- McCullagh P. and Nelder J.A. (1989) *Generalized Linear Models*. 2nd edition. Chapman et Hall.
- Vapnik V. (1999) *Statistical Learning Theory.*