Institut de Mathématiques de Toulouse, INSA Toulouse

Risk estimation and Model selection

Machine Learning

Béatrice Laurent - Philippe Besse - Mélisande Albert
Cathy Maugis - Olivier Roustant

Outline

- Introduction
- Risk and model selection
- Estimation of the generalization error
- Visualization of the risk

Introduction

- We consider supervised classification problems.
- The performance of a regression or classification algorithm is evaluated by a *risk* or *generalization error*.
- The measurement of this performance has several goals:
  - It allows to operate a *model selection* in a family of models associated with a specific learning method
  - It guides the *choice of the best method* by comparing each of the optimized models at the previous step.
  - Finally, it provides a measure of the quality or even of the *confidence* that we can give to the prediction with the selected model.

Introduction

- The main issue is to construct an *unbiased* estimator of this risk.
- The empirical risk (based on the training sample), also called the training error is *biased by optimism*, it underestimates the risk.
- If we compute an empirical estimator of the risk on a test sample (independent of the training sample), measuring the generalization capacity of the algorithm, we generally obtain higher values.
- If these new data are representative of the whole distribution of the data, we obtain an unbiased estimator of the risk.

Introduction

- Three strategies are described to obtain *unbiased estimates of risk*:
  1. a *penalisation* of the empirical risk
  2. a split of the sample: train set and test set. The train set is itself decomposed into a learning set to estimate the models for a given algorithm and a validation set to estimate the generalization error of each model in order to choose the best one, the test set is used to estimate the risk of each optimized model.
  3. by simulation: cross validation, *bootstrap*.
- The choice depends on several factors including the desired objective, the size of the initial sample, the complexity of the models, the computational complexity of the algorithms.

Outline

- Introduction
- Risk and model selection
- Estimation of the generalization error
- Visualization of the risk

Risk and model selection

- We consider **supervised regression or 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 **prediction rule** is a measurable function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that associates the output $\hat{f}(\boldsymbol{x})$ to the input $\boldsymbol{x} \in \mathcal{X}$. It depends on $\boldsymbol{D}^n$ and is thus random.
- In order to quantify the quality of the prevision, we introduce a loss function.

Risk and model selection

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 $y \neq y'$.

In real regression it is natural to consider $\mathbb{L}^p$ $(p \geq 1)$ losses

$$\ell(y, y') = |y - y'|^p.$$

If $p = 2$, the $\mathbb{L}^2$ loss is called "quadratic loss".
In classification, one can consider the 0-1 loss defined, for all $y, y' \in \mathcal{Y}$ by

$$\ell(y, y') = \mathbb{1}_{y \neq y'}.$$

Since the 0-1 loss is not smooth, it may be useful to consider other losses.

Risk and model selection

- Assuming that $Y \in \{1, 2, \ldots, K\}$, rather than providing a class, many classification algorithms provide an estimation of the probability that the output Y belongs to each class, given the input $\boldsymbol{X} = \boldsymbol{x}$, that is

$$\hat{f}_k(\boldsymbol{x}) = \hat{P}(Y = k | \boldsymbol{X} = \boldsymbol{x}), \forall k = 1, \ldots, K.$$

- Then, the prediction rule generally assigns to the input x the class that maximizes the estimated probability that is

$$\hat{Y}(\boldsymbol{x}) = \text{argmax}_{k \in \{1,2,\ldots,K\}} \hat{f}_k(\boldsymbol{x}).$$

- In this setting, a loss function often used is the so-called **cross-entropy** (or negative log-likelihood). Minimizing this loss function is equivalent to maximize the log-likelihood.
- Setting $\hat{f}(\boldsymbol{X}) = (\hat{f}_k(\boldsymbol{X}))_{1 \leq k \leq K}$, it is defined as

$$\ell(Y, \hat{f}(\boldsymbol{X})) = -\sum_{k=1}^{K} \mathbb{1}_{Y=k} \log(\hat{f}_k(\boldsymbol{X}))$$

Definition of the risk

- In all cases, the goal is to minimize the expectation of the loss function, leading to the notion of *risk*.

Definition

Let f be a prediction rule build on the learning sample $\boldsymbol{D}^n$. Given a loss function ℓ, the **risk** - or **generalisation error** - of f is defined by

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

where, in the above expression $(\boldsymbol{X}, Y)$ is independent from the learning sample $\boldsymbol{D}^n$.

- Let $\mathcal{F}$ be the set of possible prediction rules. f^* is called an optimal rule if

$$R_P(f^*) = \inf_{f\in\mathcal{F}} R_P(f).$$

A natural question then arises: is it possible to build optimal rules ?

Optimal rules

Case of real regression with $\mathbb{L}_2$ loss:

$$\mathcal{Y} = \mathbb{R}, \quad \ell(y, y') = (y - y')^2.$$

Definition

We call **regression function** the function $\eta^* : \mathcal{X} \to \mathcal{Y}$ defined by

$$\eta^*(\boldsymbol{x}) = \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}].$$

Theorem

— *The regression function* $\eta^* : \boldsymbol{x} \mapsto \mathbb{E}[Y|\boldsymbol{X} = \boldsymbol{x}]$ *satisfies:*

$$R_P(\eta^*) = \inf_{f \in \mathcal{F}} R_P(f).$$

Optimal rules

Case of real regression with $\mathbb{L}_1$ loss:

$$\mathcal{Y} = \mathbb{R}, \quad \ell(y, y') = |y - y'|.$$

Theorem

— *The regression rule defined by* $\mu^*(\boldsymbol{x}) = \text{median}[Y|\boldsymbol{X} = \boldsymbol{x}]$ *verifies:*

$$R_P(\mu^*) = \inf_{f \in \mathcal{F}} R_P(f).$$

Optimal rules

Case of classification with $0-1$ loss:

$$\ell(y, y') = \mathbb{1}_{y \neq y'}.$$

Definition

We call *Bayes rule* any function f^* of $\mathcal{F}$ 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 $R_P(f^*) = \inf_{f \in \mathcal{F}} R_P(f)$.*

Optimal rules

- The definition of the optimal rules described above 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, from which we construct a regression or classification rule.
- The aim is to find a "good" classification rule, in the sense that its risk is as small as possible.
- A first idea to estimate the risk $R_P(f) = \mathbb{E}_{(\boldsymbol{X},Y)\sim P}[\ell(Y, f(\boldsymbol{X}))]$ is to consider the **empirical risk**, or **training error**:

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

- This estimator is optimistic and will under estimate the risk as illustrated below.

Determination coefficient

- Minimizing the empirical risk $R_n(f)$ is equivalent to maximize the determination coefficient $R^2(f)$ defined by

$$R^2(f) = 1 - \frac{\sum_{i=1}^n (Y_i - f(\boldsymbol{X_i}))^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{R_n(f)}{R_n(\bar{Y})}.$$

- This is not a good idea for model selection. Among models with increasing complexity, it leads to select the most complex model, inducing overfitting.

Empirical risk

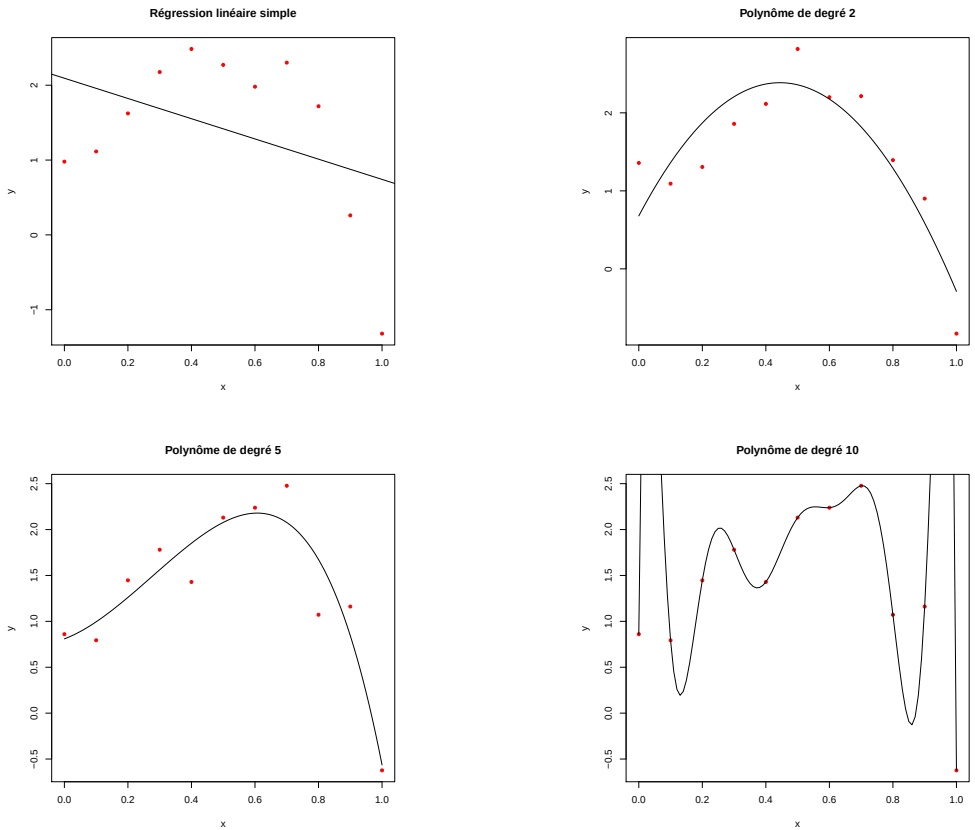


Figure: Polynomial regression: adjusted model are polynomials with respective degrees 1: $R^2 = 0.03$, 2: $R^2 = 0.73$, 5: $R^2 = 0.874$ and 10: $R^2 = 1$.

Empirical risk

- The empirical risk is equal to 0 for the polynomial of degree $n-1$ (which has n coefficients) and passes through all the training points.
- The empirical risk (or training error) is not a good estimate of the generalization error: it decreases as the complexity of the model increases.
- Minimizing the training error leads to select the most complex model, this leads to **overfitting**.
- The next figure illustrates the optimism of the training error, that underestimates the generalization error, which is estimated here on a test sample.

Optimism of the empirical risk

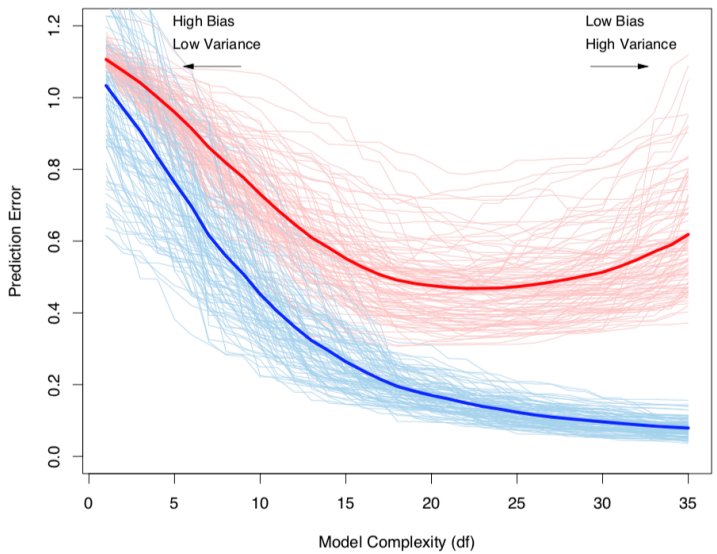


Figure: Behavior of training error (in blue) and test error (in red) as the complexity of the model increases. Source: "The elements of Statistical Learning", T. Hastie, R. Tibshirani, J. Friedman.

Minimisation of the penalized empirical risk

A first way to have a good criterion for model selection is to minimize the empirical risk **plus a penalty term**, the penalty term will penalize too complex models to prevent overfitting.
Let f^* an optimal rule, such that $R_P(f^*) = \inf_f R_P(f)$.
From the training sample, the objective is to determine a model F for which the risk of the estimator $\hat{f}_F$ built on this model (for example by minimization of the empirical risk) is close to the one of the oracle

$$R_P(\hat{f}_F) - R_P(f^*) = \underbrace{\left\{ R_P(\hat{f}_F) - \inf_{f \in F} R_P(f) \right\}}_{\textbf{Estimation error}} + \underbrace{\left\{ \inf_{f \in F} R_P(f) - R_P(f^*) \right\}}_{\textbf{Approximation error}}$$

(Variance) $\nearrow$ (Bias) $\searrow$ (dimension of F)

These two terms are of different natures. To evaluate them, we use tools respectively from statistics and approximation theory.

Minimisation of the penalized empirical risk

- The selection of a model $\hat{F}$ in a collection of models $\mathcal{C}$ for which the risk of the estimator $\hat{f}_{\hat{F}}$ is close to the one of the oracle will be obtained by the minimization of a penalized criterion of the type:

$$\hat{F} = \operatorname{argmin}_{F \in \mathcal{C}} \left\{ R_n(\hat{f}_F) + \text{pen}(F) \right\}.$$

- In the above formula, a penalty is added to the empirical risk. The role of the penalty is to penalize models with "large" dimension, in order to avoid overfitting.
- The optimal choice of the penalty (according to the statistical models considered) is a very active research topic in statistics.

Minimisation of the penalized empirical risk

- The more complex a model, the more flexible it is and can adjust to the observed data and therefore the smaller the bias.
- On the other hand, the variance increases with the number of parameters to be estimated and therefore with this complexity.
- The objective is to minimize the quadratic risk, which is a sum of the variance and the squared bias term.
- Hence, we are looking for the best compromise between the bias and the variance term: it is sometimes preferable to accept to bias the estimate as for example in *ridge* regression to reduce its variance.

Penalized criterion: Mallow's C_P

- The Mallow's C_p (1973) was historically the first penalized criterion, introduced for Gaussian linear model.
- It is based on the penalization of the least square criterion by a penalty which is proportional to the dimension of the model.
- This criterion is defined as follows : for a model m,

$$C_p(m) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i(m))^2 + 2p\widehat{\sigma}^2$$

where p is the number of parameters of the model m, $\hat{Y}_i(m)$ the predictions obtained with the model m and $\widehat{\sigma}^2$ is an estimation of the variance of the error obtained by a model with large dimension (small bias).

Penalized criterion: Mallow's C_P

- In framework of a the linear model $Y = X\beta + \varepsilon$, for which this criterion was historically introduced, the expression becomes

$$C_p(m) = \sum_{i=1}^{n} (Y_i - (X\hat{\beta}(m))_i)^2 + 2p\widehat{\sigma}^2,$$

where $\beta \in \mathbb{R}^p$, $\hat{\beta}(m)$ is the least square estimator of β obtained with model m.

Penalized criterion: Mallow's C_P

- The next figure shows the behavior of the Mallow's C_p in the pedagogical example of polynomial regression.
- This criterion selects a polynomial with degree 3.

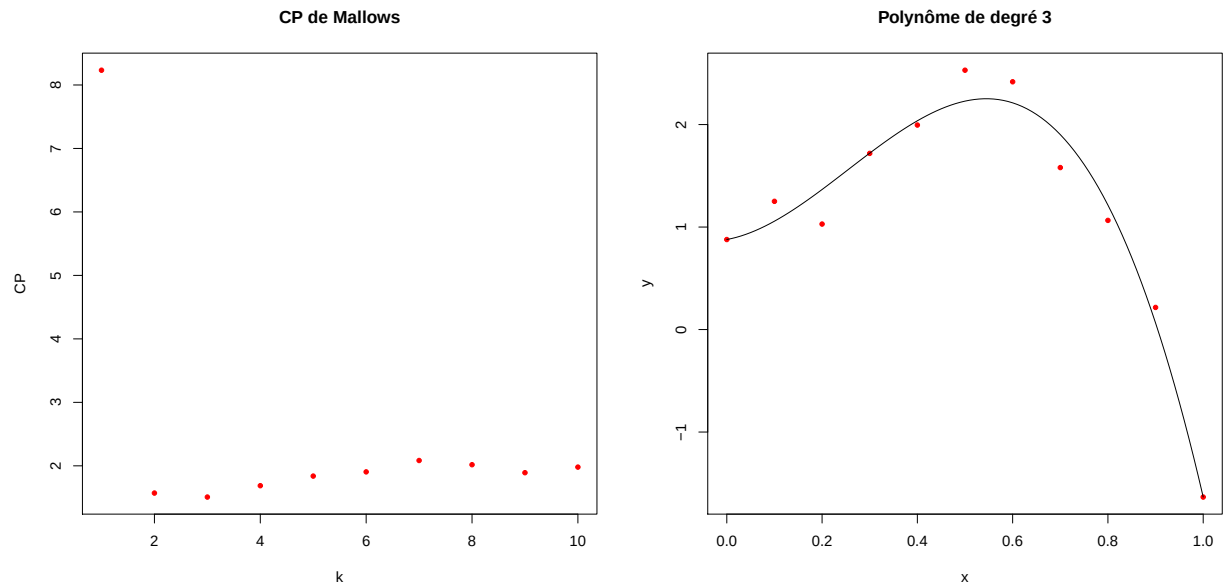


Figure: *Polynomial regression: Mallow's C_P against the degree of the polynomial. A polynomial with degree 3 is selected*

AIC, AIC_c, BIC

- While Mallow's C_P is associated to the quadratic loss, Aikaike's Information Criterion (1974) (AIC) is related to the log-likelihood.
- It corresponds to the opposite of twice the log of the likelihood $L(., \beta)$, for β equal to the maximum likelihood estimator $\hat{\beta}(m)$, plus a penalty term proportional to the dimension of the model:

$$\text{AIC}(m) = -2 \sum_{i=1}^{n} \log(L(X_i, \hat{\beta}(m)) + 2p.$$

- The quantity $-2 \log(L(., \beta))$ is also called *deviance*.
- In the Gaussian model with variance assumed to be known, the deviance and least square criterion coïncide. In this case, AIC is equivalent to C_P.

AIC, AIC_c, BIC

- A refined version of the AIC criterion, called corrected AIC is defined as

$$AIC_c(m) = -2\sum_{i=1}^{n} \log(L(X_i, \hat{\beta}(m)) + \frac{n+p}{n-p-2}.$$

  It is recommended for small sample sizes and asymptotically equivalent to AIC for large values of n.

- Another criterion called BIC (*Bayesian Information Criterion*), (Schwartz; 1978) derives from Bayesian arguments. It is also based on the penalization of the negative log likelihood, but with a higher penalty than AIC, and hence will generally select simpler model than AIC.

$$BIC(m) = -2\sum_{i=1}^{n} \log(L(X_i, \hat{\beta}(m)) + \log(n)p.$$

- Whatever the chosen criterion, the strategy is to select a model minimizing this criterion, among a collection of possible models.

Outline

- Introduction
- Risk and model selection
- Estimation of the generalization error
- Visualization of the risk

Estimation of the generalization error

- Instead of minimizing a penalized criterion, other strategies for model selection consists in estimating the generalization error, either with data that where not used during the training phase, or by Bootstrap's methods.
- 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.
- We concentrate here on the first objective, assuming that we have a **test set** for model assessment.

Estimation by cross-validation

- As seen previously, it is crucial to evaluate the performances of an algorithm on data that were not used during the learning step.
- For this purpose, cross-validation methods are widely used. The main variations of this method are presented here.
- **Holdout cross-validation**: If we have enough data in the training set, the recommended approach is to divide randomly the training set into a learning sample and a validation sample.
  - The learning sample denoted $\boldsymbol{D}_1^{n_1}$ is used to train the models (generally by minimizing the training error).
  - The validation sample denoted $\boldsymbol{D}_2^{n_2}$ is used to estimate the generalization error of each model by the quantity

$$\frac{1}{n_2} \sum_{(\boldsymbol{X_i}, Y_i) \in \boldsymbol{D}_2^{n_2}} \ell(Y_i, f(\boldsymbol{X_i})).$$

It is generally recommended to take $2/3$ of the training data for the learning sample, $1/3$ of the data for the validation sample.

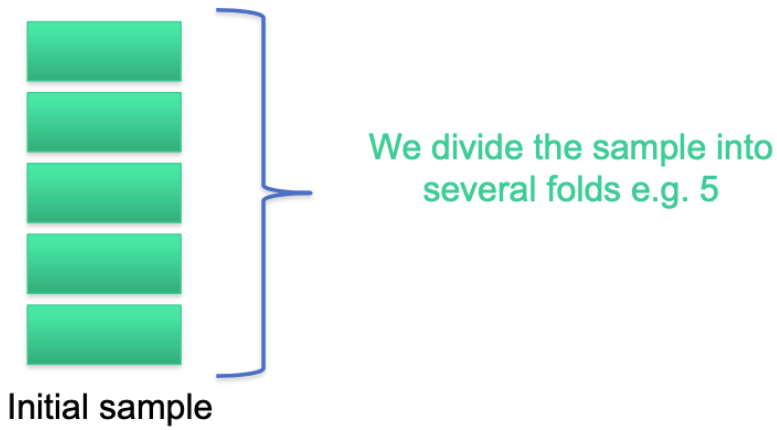
Estimation by cross-validation

- Often, taking only 2/3 of the data set to train the models may lead to bad performances, especially if we do not have too much data.
- Moreover, if the size of the validation set is small, the estimation of the generalization error will have a high variance and be highly dependent on this validation set.
- To prevent this problem, K **fold cross-validation** is widely used.
- K-fold cross-validation is a widely used method to estimate the generalization error without splitting the training set as done in the previous section.

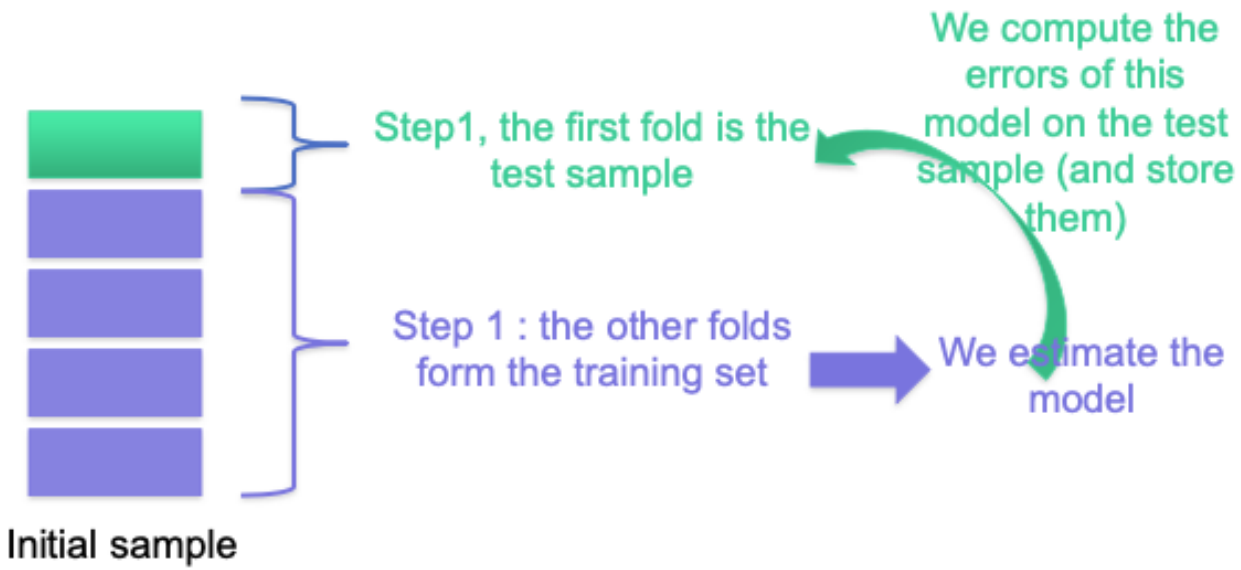
K-fold cross-validation

- We split randomly the training data into K subsamples, with (almost) the same size ($K = 10$ generally).
- Each of the K folds will be successively used as a validation sample.
- When the fold k is the validation 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.

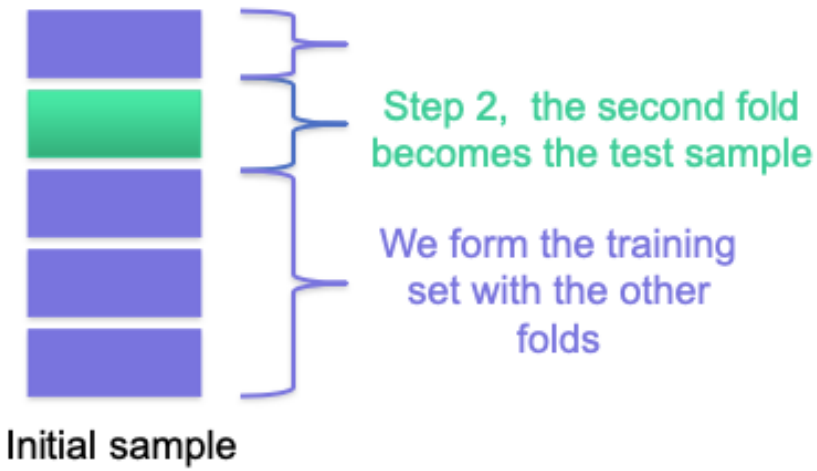
K-fold cross-validation



K-fold cross-validation



K-fold cross-validation



K-fold 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 $k = 1, \ldots, K$, 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 $m \in \mathcal{M}$. which minimizes $CV(m)$.

K-fold cross-validation

- If K is small (for example $K = 2$), each estimator $\hat{f}_m^{(-k)}$ is trained with around $n/2$ observations. Hence, these estimators are less accurate than an estimator built with n observations, leading to a bias in the estimation of the generalization error by cross-validation.
- When the number of folds $K = n$, the method is called leave-one-out cross-validation.
- This method has a low bias to estimate the generalization error, but a high variance since all the estimators $\hat{f}_m^{(-i)}$ are highly correlated. The computation time is also high for the leave-one-out method.
- This is why, in practice an intermediate choice such as $K = 10$ is often recommended. This is generally the default value in softwares.

Monte Carlo Cross-Validation

- This method consists in iterating several times the random subdivision of the initial sample into a train set and a test set.
- The most simple way to apply Monte Carlo Cross-Validation is to iterate the holdout procedure.
- The advantage of this method is to provide an estimation of the whole distribution of the risk, for all considered methods. The disadvantage is the computational time.

Monte Carlo Cross-Validation

Monte Carlo Cross-Validation

- For k=1 to B
  - Split randomly the sample into two parts: *train* set and *test* set with a prescribed proportion
  - For models *in* collection of models
    - Estimate the parameters of the current model with the training set.
    - Compute the test error by the empirical risk on the test set.
  - End For
- End For
- For each model, compute the mean of the B test errors and draw the boxplots of the distributions of these errors.

Monte Carlo K-fold Cross-Validation

This strategy can also be coupled with K-fold cross-validation as described in the following Algorithm.

Monte Carlo K-fold Cross-Validation

- For k=1 to B
  - Split randomly the sample into two parts: *train* set and *test* set with a prescribed proportion.
  - For method *in* list of methods
    - Optimise the complexity (or tuning parameters) of the method by *K-fold cross-validation* with the training set.
    - Estimate the parameters of the optimized model for this method with the training set.
    - Compute the test error by the empirical risk on the test set for the optimized model of the current method.
  - End For
- End For
- For each optimized method, compute the mean of the B test errors and draw the boxplots of the distributions of these errors.

Monte Carlo K-fold Cross-Validation

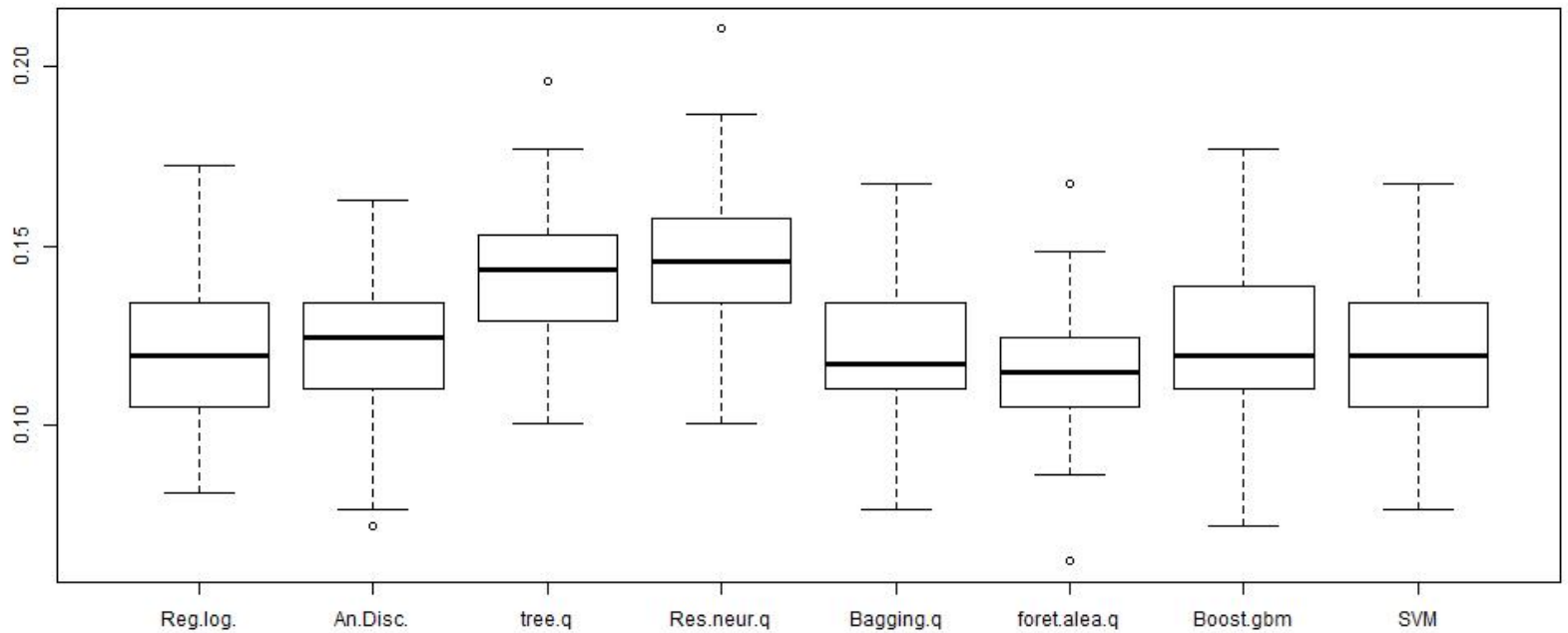


Figure: Boxplot of the test errors for various methods optimized by Monte Carlo K-fold Cross-Validation on Ozone data set

Estimation by Bootstrap

- Let us first describe the Bootstrap, before showing how it can be used to estimate the extra-sample prediction error.
- Suppose we have a training data set $\mathbf{Z} = \{z_1, \dots z_n\}$, with $z_i = (\boldsymbol{x_i}, y_i)$ and a model to be fitted on these data.
- We denote by $\hat{f}$ the model fitted with the sample $\mathbf{Z}$.
- The principle of the bootstrap is to randomly draw datasets of size n with replacement from the original sample $\mathbf{Z}$.
- Conditionally on $\mathbf{Z}$, all these draws are independent.

Bootstrap samples

The two next slides show two bootstrap samples from the original dataset presented here.

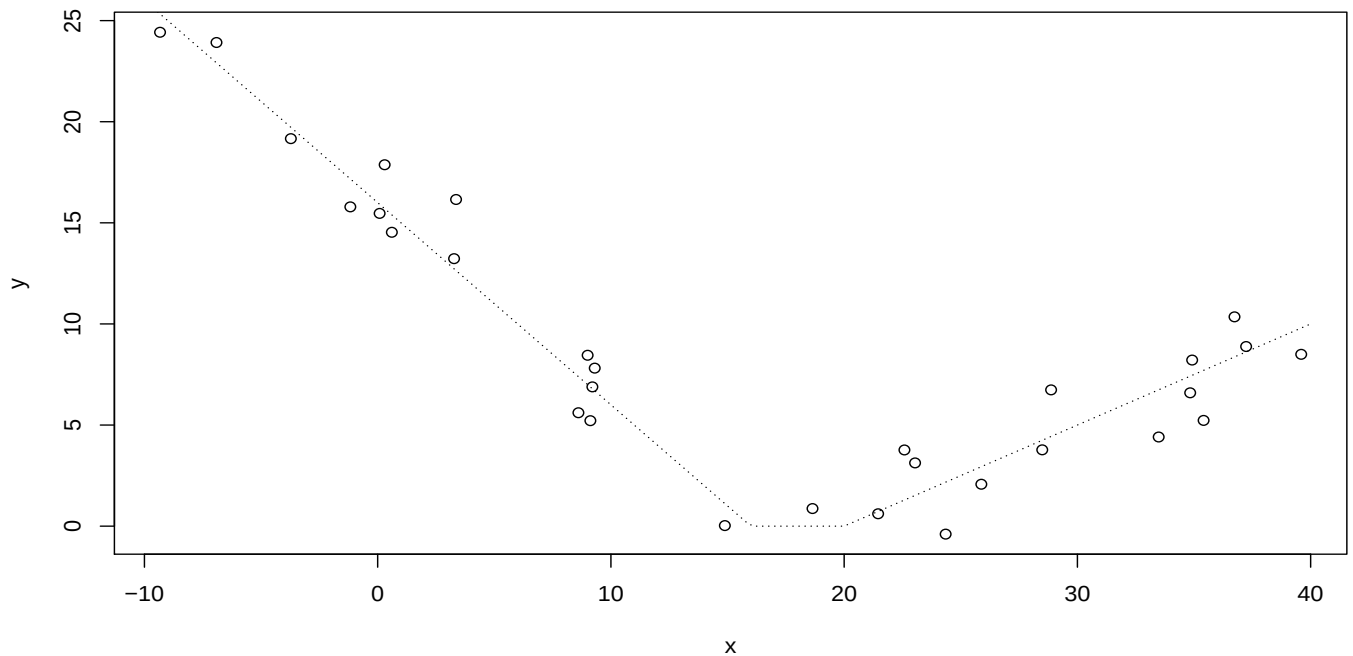


Figure: Original data

Bootstrap samples

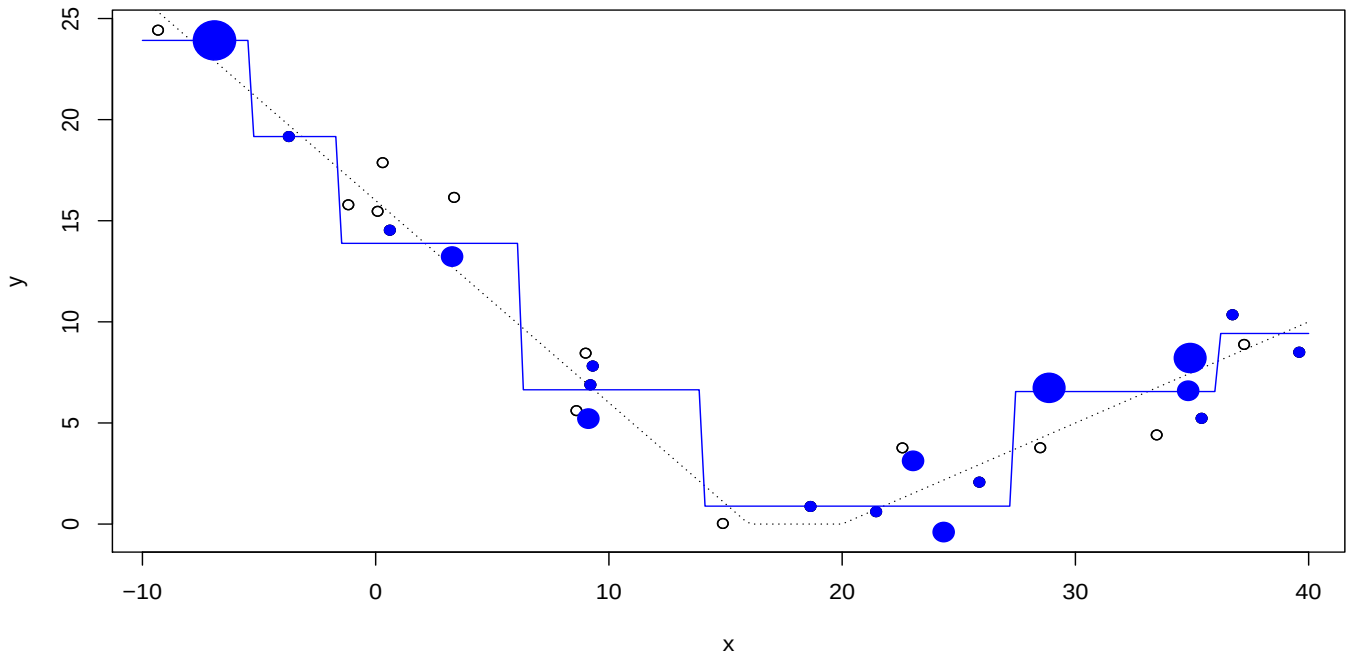


Figure: Bootstrap sample $n^o 1$ (in blue), and corresp. prediction with tree. The point size is proportional to the number of replicates.

Bootstrap samples

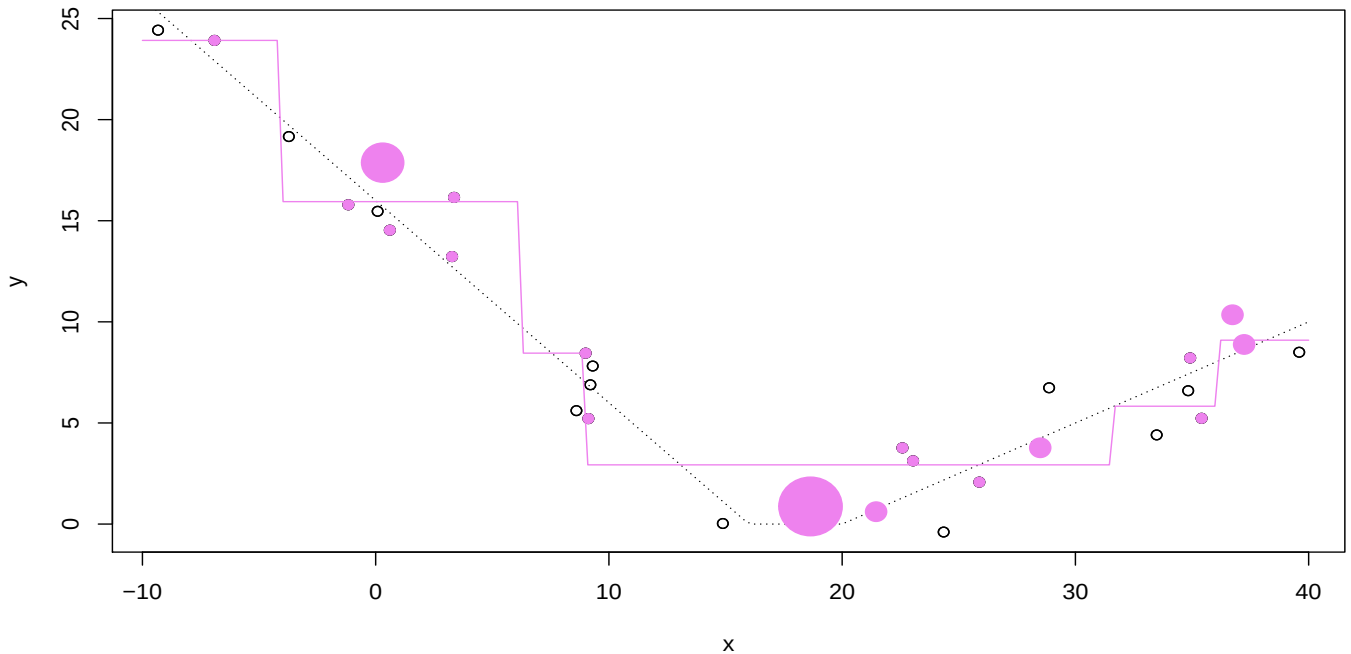


Figure: Bootstrap sample $n^o 2$ (in violet), and corresp. prediction with tree. The point size is proportional to the number of replicates.

- We draw B bootstrap samples (for example $B = 500$) that we denote $(\mathbf{Z}^{*b}, b = 1, \ldots, B)$.
- We fit the model with each of these bootstrap samples. We denote $\hat{f}^{*b}$ the model fitted with the sample $\mathbf{Z}^{*b}$.
- A first idea to estimate the prediction error of $\hat{f}$ is to consider the following estimator:

$$\widehat{\mathrm{Err}}_{boot} = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n}\ell(y_i, \hat{f}^{*b}(\boldsymbol{x_i})),$$

measuring the mean, over the B bootstrap predictors, of the error on the training sample $\mathbf{Z}$.

- This is not a good estimate of the generalization error since the bootstrap samples and the original sample have many observations in common. Hence, this estimator will be too optimistic: it will underestimate the generalization error.

Out-of-bag estimator

- A better idea is to exploit the fact that each bootstrap sample does not contain all the observations of the original sample.

$$\mathbb{P}\left(\text{Observation } z_i \notin \text{bootstrap sample } b\right) = \left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e} = 0.368.$$

- Mimicking the idea of cross-validation, we denote by C^{-i} the set of indices b in $\{1, \dots B\}$ such that $\mathbf{Z}^{*b}$ does not contain the observation z_i, and we introduce the estimator

$$\widehat{\mathrm{Err}}_{oob} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} \ell(y_i, \hat{f}^{*b}(x_i)).$$

- This estimator is called the out-of-bag estimator.

Out-of-bag estimator

- If B is large enough, then for all i , $|C^{-i}| \neq 0$. Otherwise, the observation i for which $|C^{-i}| = 0$ can be removed from the above formula.
- This estimator uses extra sample observation to estimate the error of each predictor $\hat{f}^{*b}$, avoiding the overfitting problem encountered by $\widehat{\text{Err}}_{boot}$.
- Nevertheless, in expectation, each bootstrap sample contains $0.632n$ observations, which is less that $2n/3$ and we would like to estimate the generalization error of a predictor $\hat{f}$ built with n observations.
- Each bootstrap predictor $\hat{f}^{*b}$ will be less accurate than $\hat{f}$ since it is built with a smaller sample size.
- This induces a bias in the estimation of the generalization error of $\hat{f}$ by $\widehat{\text{Err}}_{oob}$.

Out-of-bag estimator

- To correct this bias, the ".632 bootstrap estimator " has been introduced by Efron and Tibshirani (1997).
- It is defined by

$$\widehat{\mathrm{Err}}^{(.632)} = .368\bar{err} + .632\widehat{\mathrm{Err}}_{oob},$$

where $\bar{err}$ is the training error of $\hat{f}$.

Remarks

1. All the estimators proposed to estimate the generalization error are asymptotically equivalent, and it is not possible to know which method will be more precise for a fixed sample size n.
2. The boostrap is time consuming and more complicated. It is less used in practice. Nevertheless, it plays a central role in recent methods of aggregation, involving the bagging (for bootstrap aggregating) such as random forests.
3. In conclusion, the estimation of a generalization error is delicate, and it is recommended to consider the same estimator to compare two prediction methods and to be very careful, without theoretical justification, to use one of these estimation to certify an algorithm. For this last purpose, the use of a test sample, with sufficiently large size, would be recommended.

We will now present the strategy adopted in the practical works.

Strategy for the pratical works (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
  - SVM
  - Regression trees, Random Forests
  - Neural networks
  - Boosting ...
- Each algorithm has 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.

Outline

- Introduction
- Risk and model selection
- Estimation of the generalization error
- Visualization of the risk

Visualization of the risk

- In the case of the regression, as seen above, we can represent a boxplot of the risk of each optimized algorithm if we have used a Monte Carlo K folds cross-validation procedure.
- For binary classification problem, ROC curves are a convenient way to visualize the relative performances of several binary classification methods.

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 - Definition

Definitions from the contingency table

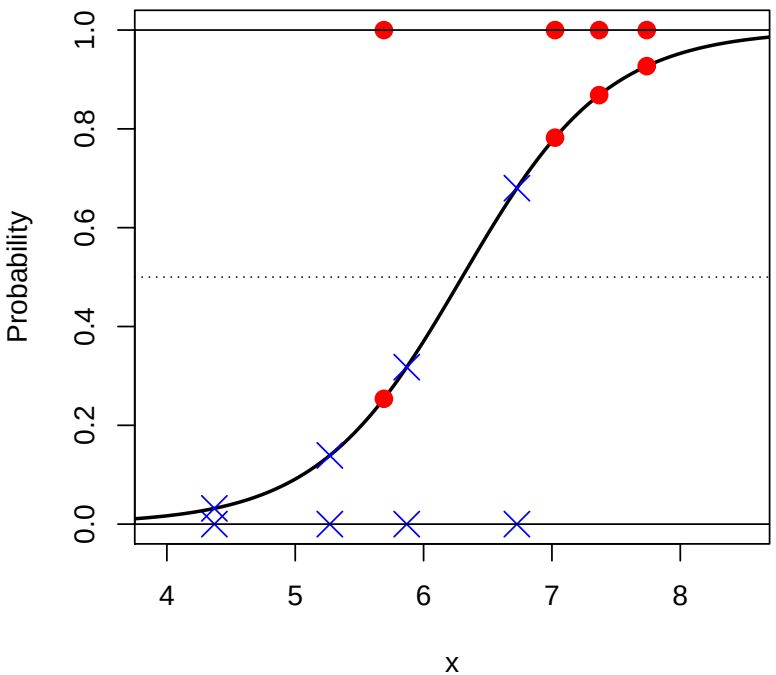
Prediction: if $\widehat{\pi}_i > s$, $\widehat{Y}_i = 1$ else $\widehat{Y}_i = 0$

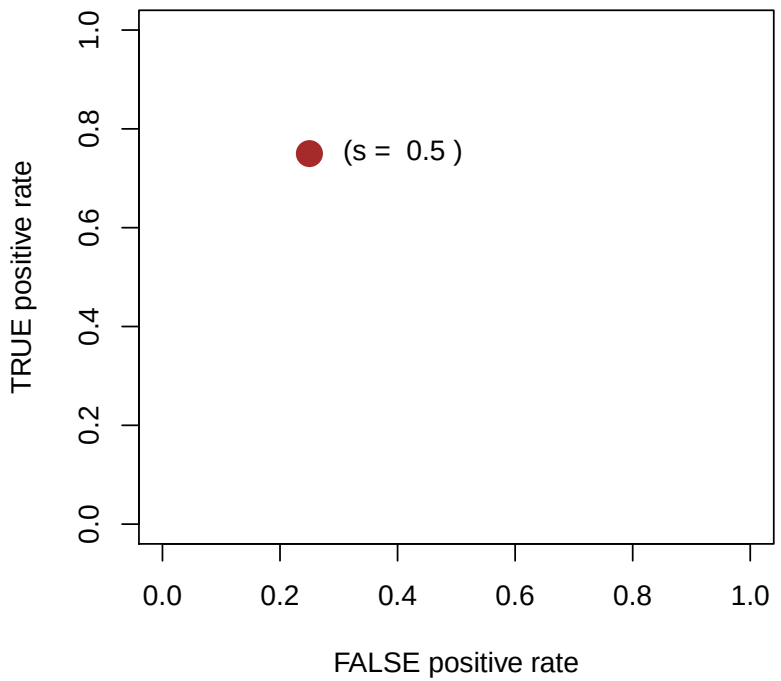
Prediction	Observation		Total
	$Y_i = 1$	$Y_i = 0$	
$\widehat{Y}_i = 1$	$n_{11}(s)$	$n_{10}(s)$	$n_{1+}(s)$
$\widehat{Y}_i = 0$	$n_{01}(s)$	$n_{00}(s)$	$n_{0+}(s)$
Total	n_{+1}	n_{+0}	n

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

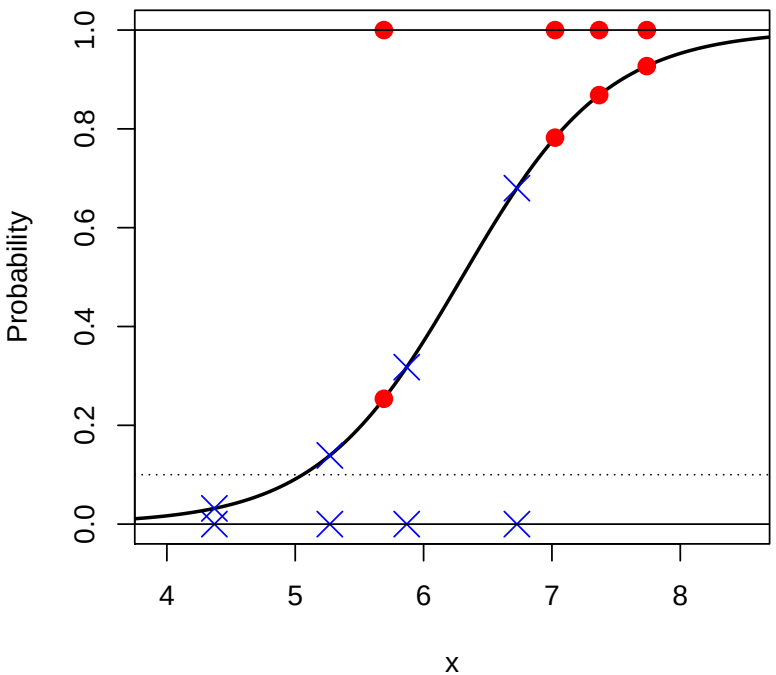
The ROC curve plots TPR(s) versus FPR(s) for all values of $s \in [0, 1]$.

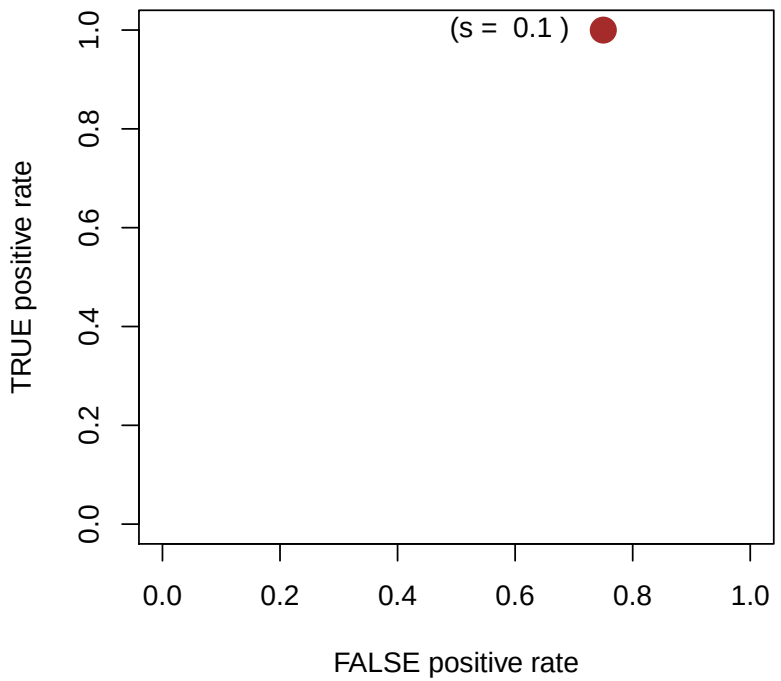
ROC curve - Illustration in 1D



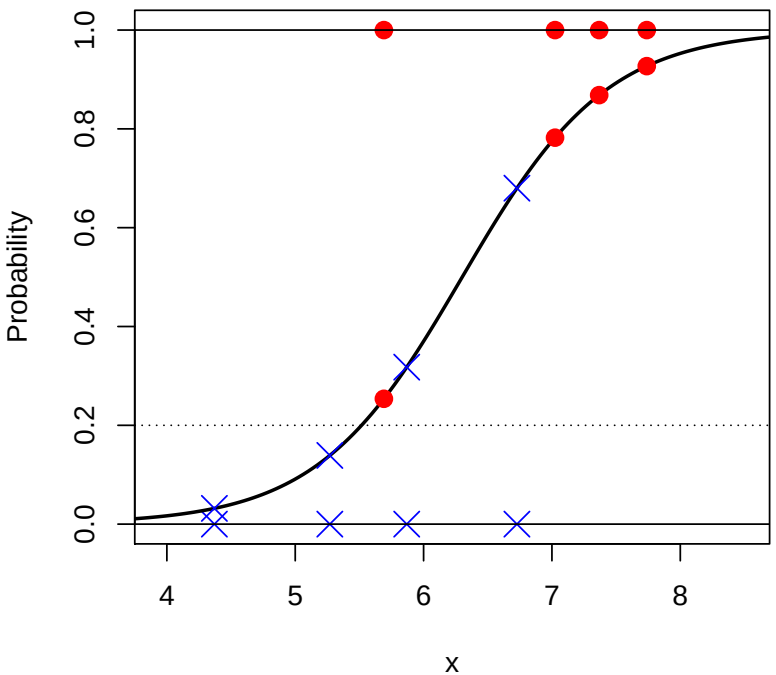


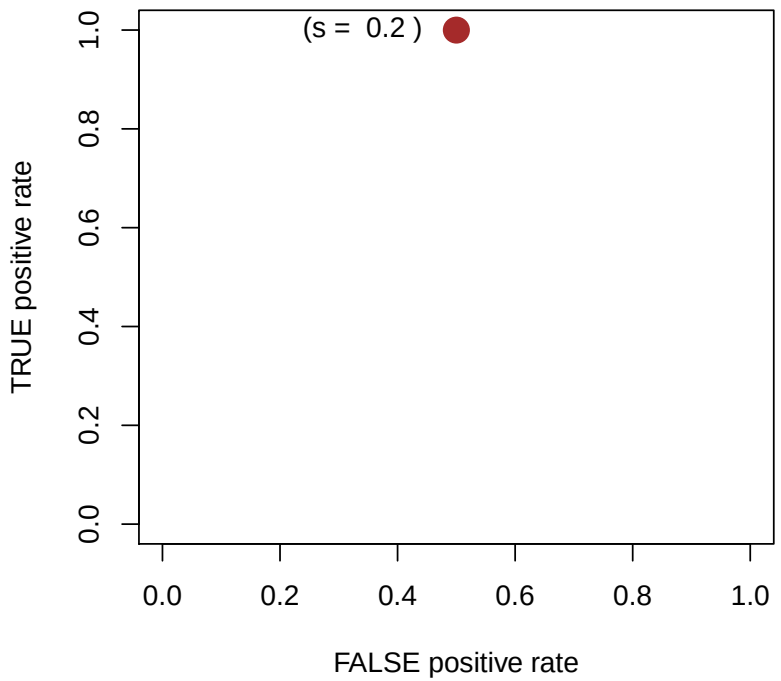
ROC curve - Illustration in 1D



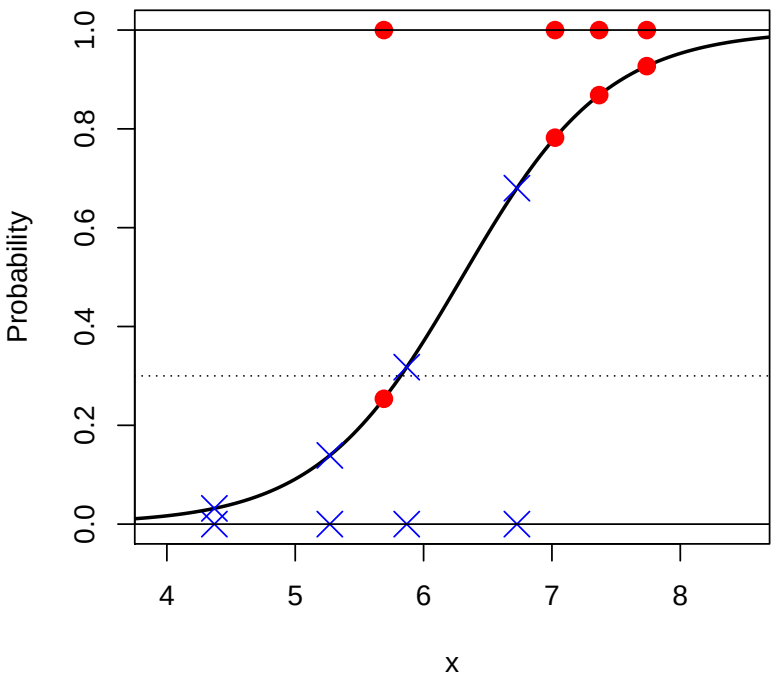


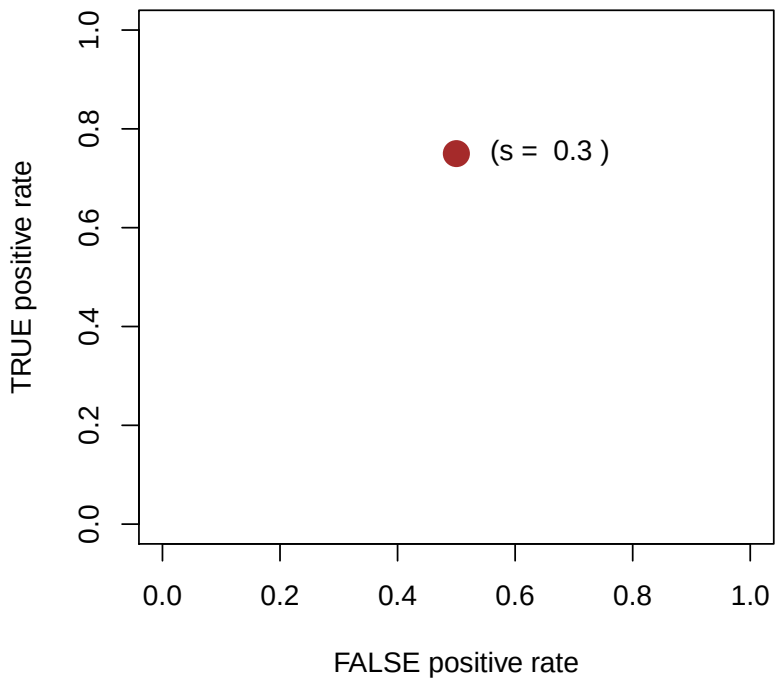
ROC curve - Illustration in 1D



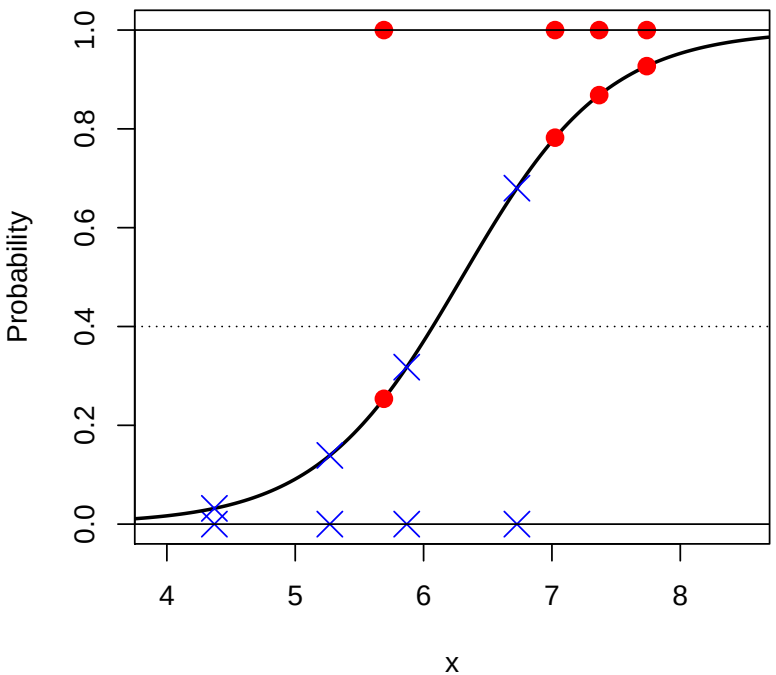


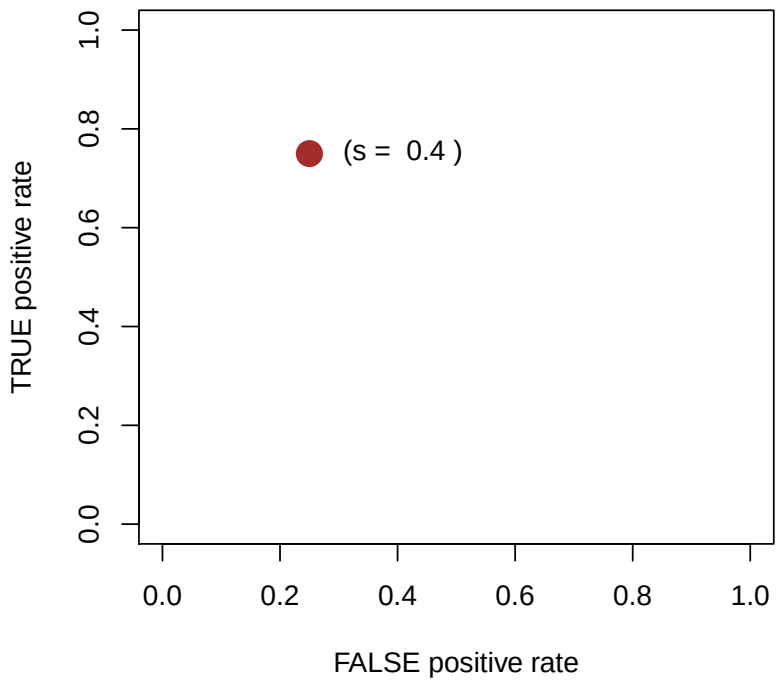
ROC curve - Illustration in 1D



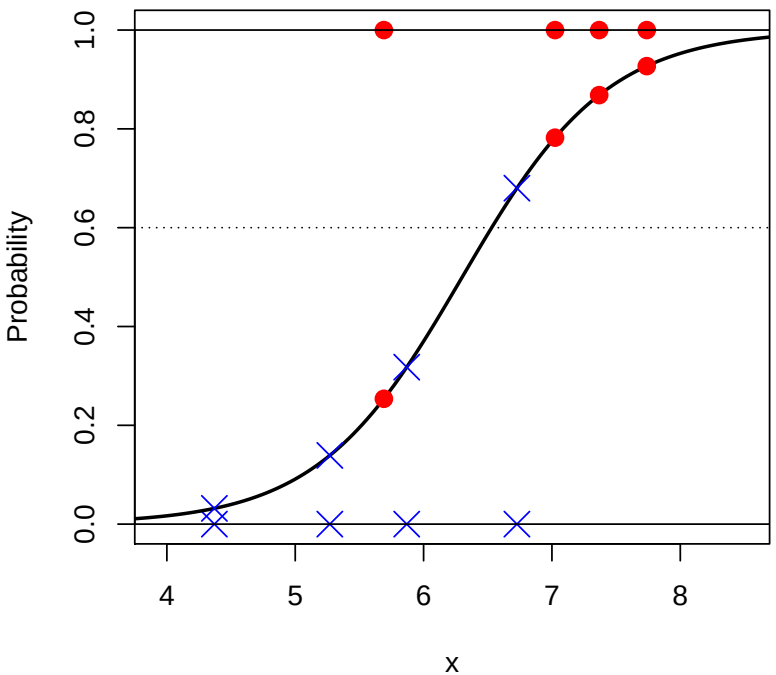


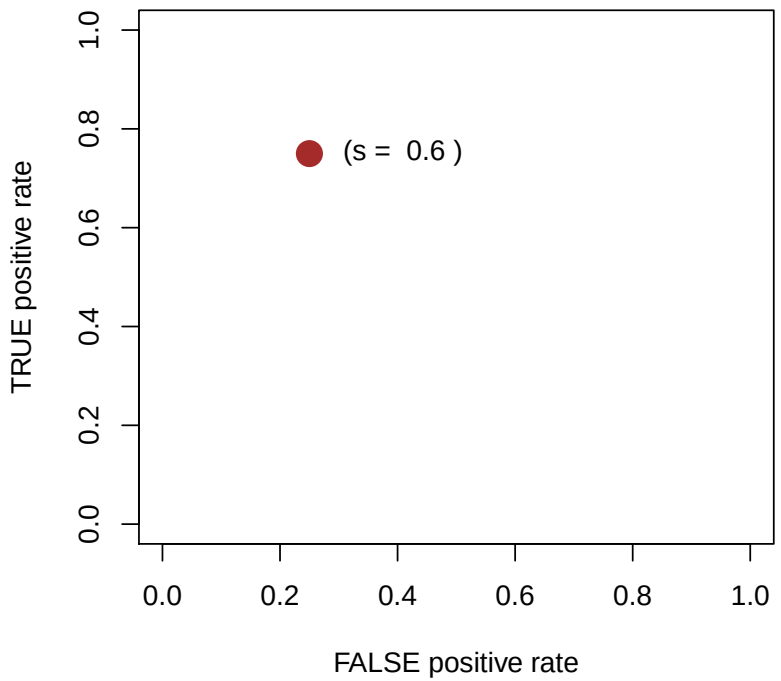
ROC curve - Illustration in 1D



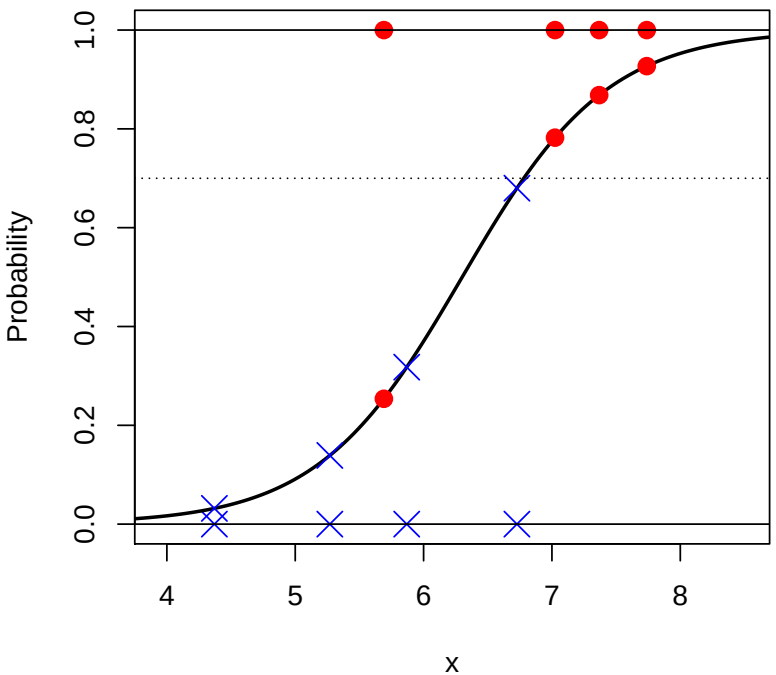


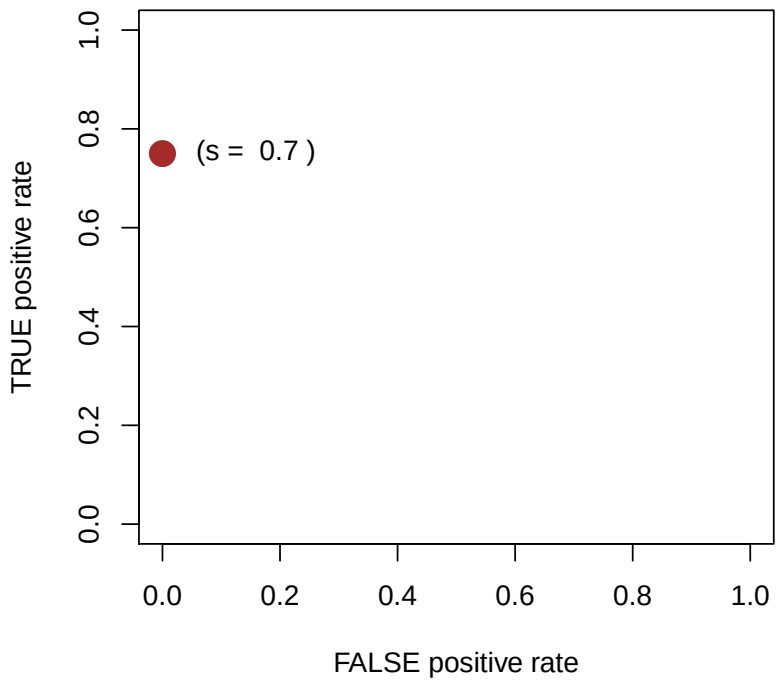
ROC curve - Illustration in 1D



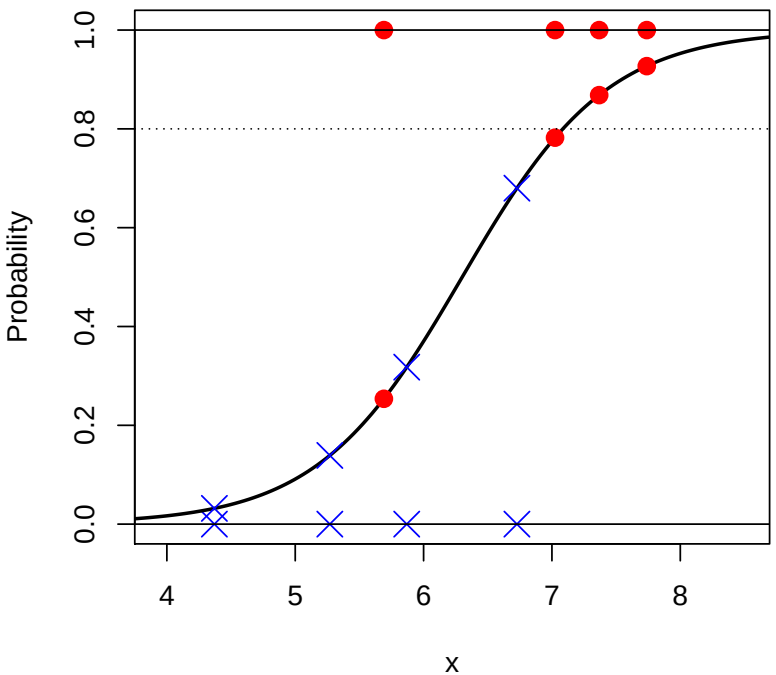


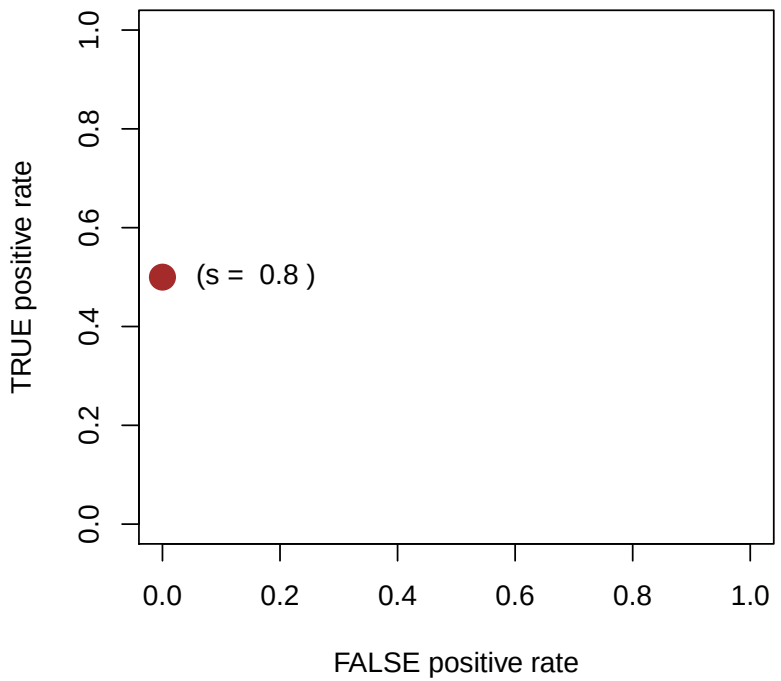
ROC curve - Illustration in 1D



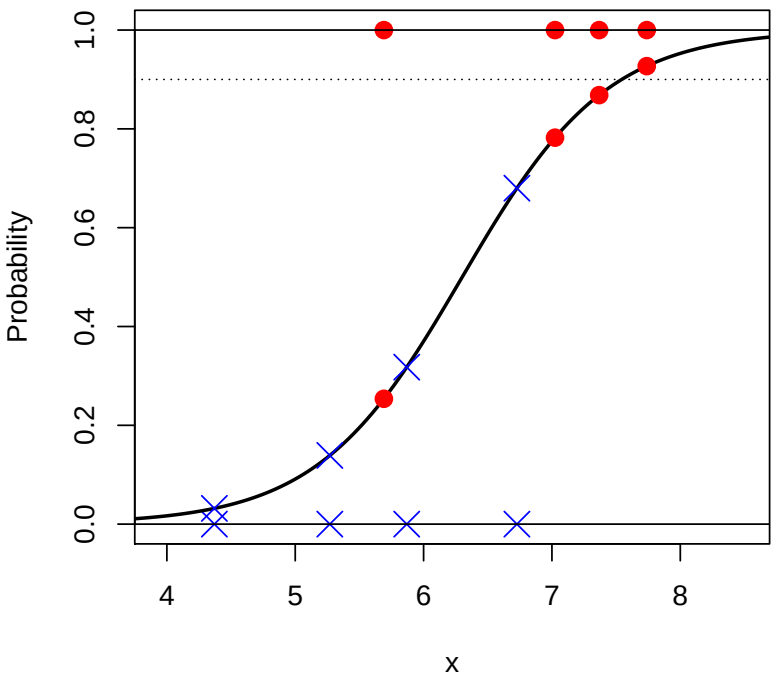


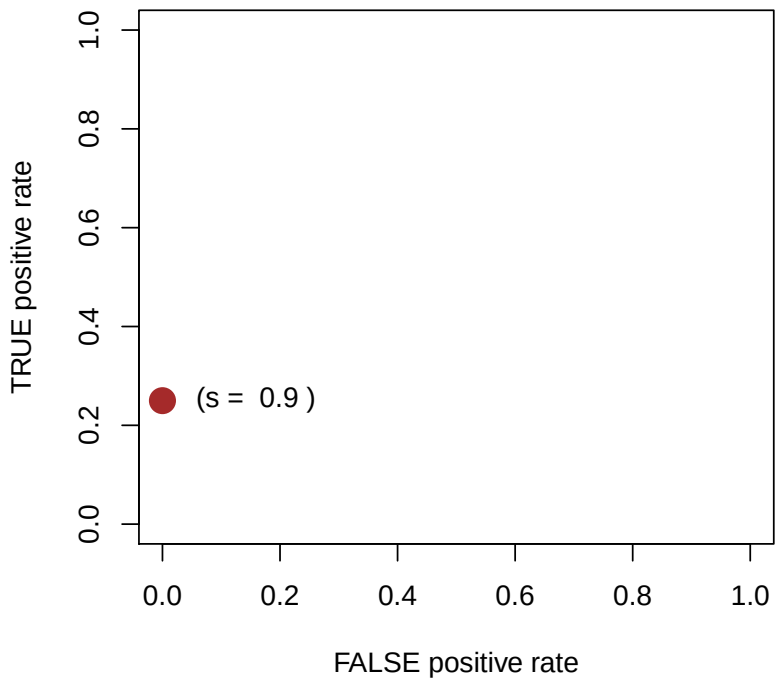
ROC curve - Illustration in 1D



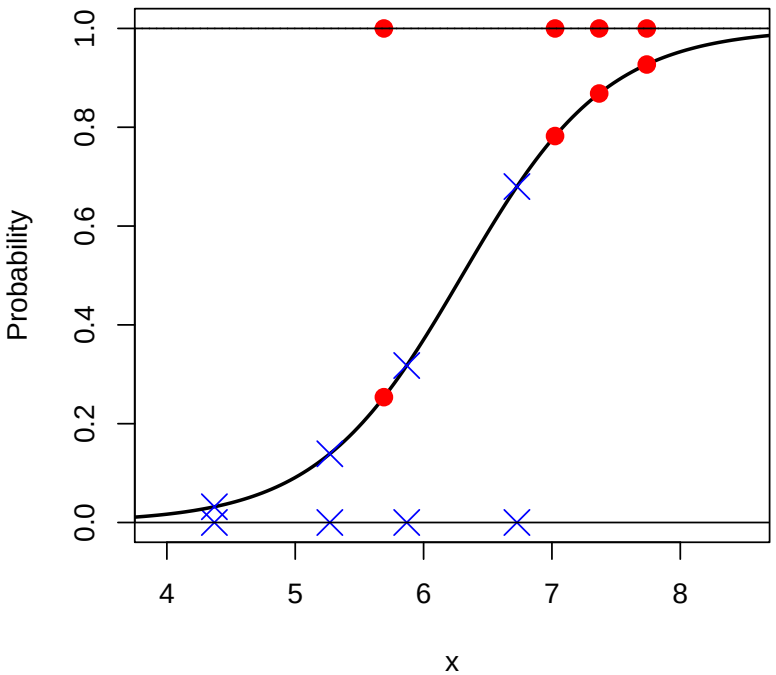


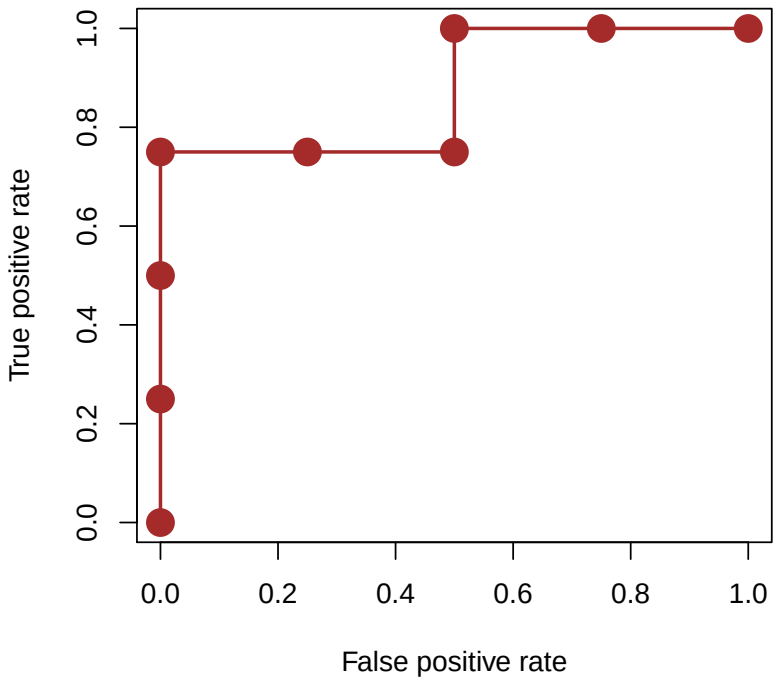
ROC curve - Illustration in 1D





ROC curve - Illustration in 1D





Usage of ROC curve to select classifiers

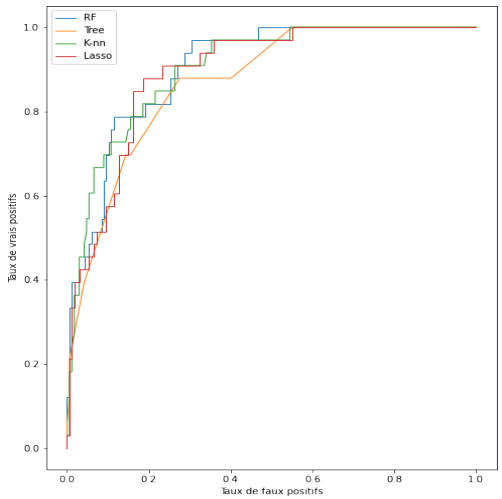


Figure: Ozone: ROC curve for several algorithms.

- In order to compare several methods with various complexity, the ROC curves should be estimated 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.
- For a given expected TPR, we can select the algorithm with the smallest FPR.

Conclusion

- The estimation of a generalization error is delicate but very important both for model selection and model assessment.
- The choice of the strategy depends on several factors including the desired objective, the size of the data set, the computational complexity of the algorithms...
- If the data set is small, a double loop of K folds cross-validation is recommended or the use of Monte Carlo K folds cross-validation algorithm.

References

- H. Akaïke, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (1974).
- B. Efron, The Jackknife, the Bootstrap and other Resampling Methods, SIAM (1982).
- B. Efron et R. Tibshirani, Improvements on Cross-Validation: The .632+ Bootstrap Method, Journal of the American Statistical Association 92 (1997), no 438, 548–560.
- T. Hastie, R. Tibshirani et J Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer, (2009), Second edition.
- C.L. Mallows, Some Comments on Cp, Technometrics 15 (1973), 661–675.
- G. Schwarz, Estimating the dimension of a model, Annals of Statistics, 6 (1978), 461–464.
- M. Stone, An Asymptotic Equivalence of Choice of Model by Cross- Validation and Akaike's Criterion, Journal of The Royal Statistical So- ciety B 39 (1977), 44–47.
- Wikistat.fr, Github Wikistat