Institut de Mathématiques de Toulouse, INSA Toulouse

Linear methods for classification

Data Mining
October 2022

Béatrice Laurent - Philippe Besse - Olivier Roustant

Outline

- Introduction to supervised classification
- Logistic regression
- ROC curves
- Support Vector Machine

Introduction to supervised 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.

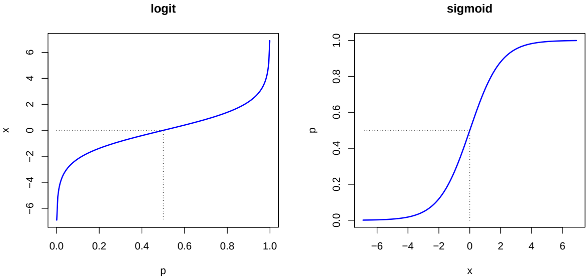
Generalized linear models

- Logistic regression
  - Definitions
  - Estimation of the parameters
  - Application
  - Multiclass classification

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 :

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



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 $\beta \in \mathbb{R}^p$,

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

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

- The quantity $odds(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}$ is called the odds for $\boldsymbol{x}$.
  For example, if $\pi(\boldsymbol{x}) = 0.8$, then $odd(\boldsymbol{x}) = 4$ which means that the chance of success $(Y = 1)$ when $\boldsymbol{X} = \boldsymbol{x}$ is 4 against 1.
- The odds ratio between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ is $OR(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = odds(\boldsymbol{x})/odds(\tilde{\boldsymbol{x}})$.

- Setting

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

the logistic regression model corresponds to

$$logit(\pi(\boldsymbol{x})) = \ln(odds(\boldsymbol{x})) = \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle.$$

- This is a linear model for the logarithm of the odds.
- g is called the **logit** “link” function.
- Other link functions can be considered such as :
  - The **probit** function $g(\pi) = F^{-1}(\pi)$ where F is the distribution function of the standard normal distribution.
  - The **log-log** function $g(\pi) = \ln(-\ln(1-\pi))$.

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

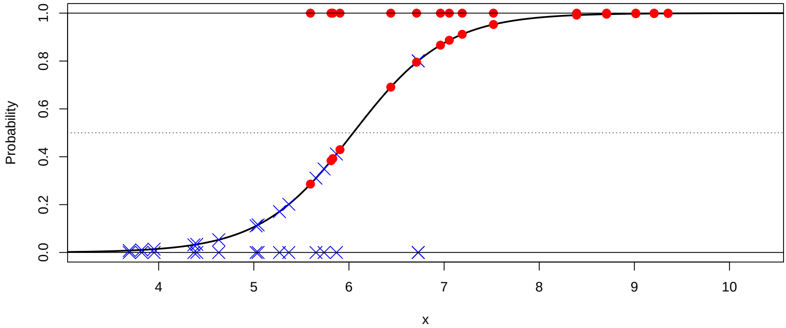


FIGURE – Logistic regression for a dataset composed of 2 groups of size 15, sampled from Normal distributions, centered at 5 and 7, with variance 1.

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 \ $\hat{Y}$	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.

Multinomial or polytomic regression

- We consider here the case where the response variable Y has M non ordered levels $u_1, \dots, u_M$.
- We set $\pi_m(\boldsymbol{x}) = \mathbb{P}(Y = u_m | \boldsymbol{X} = \boldsymbol{x})$ for $m = 1, \dots M$.

$$\sum_{m=1}^{M} \pi_m(\boldsymbol{x}) = 1.$$

- We choose a reference in the levels, we assume that this is the first level u_1.
- The multinomial regression model is defined by

$$\log\left(\frac{\pi_m(\boldsymbol{x})}{\pi_1(\boldsymbol{x})}\right) = \langle \beta^{(\boldsymbol{m})}, \boldsymbol{x} \rangle \ \forall m = 2, \dots M.$$

- This is equivalent to

$$\pi_m(\boldsymbol{x}) = \frac{\exp(\langle \beta^{(\boldsymbol{m})}, \boldsymbol{x} \rangle)}{1 + \sum_{m'=2}^{M} \exp(\langle \beta^{(\boldsymbol{m'})}, \boldsymbol{x} \rangle)}$$

  which generalizes the logistic regression model (where $u_1 = 0$ and $u_2 = 1$).

- In order to estimate the parameters $\beta^{(\boldsymbol{m})}$, we maximize the likelihood :

$$L(\underline{\mathrm{Y}}, \beta) = \prod_{i=1}^{n} \prod_{m=1}^{M} \pi_m(\boldsymbol{X_i})^{\mathbb{1}_{Y_i = u_m}}.$$

Outline

- Logistic regression
  - Definitions
  - Estimation of the parameters
  - Application
  - Multiclass classification
- ROC curves
- Support Vector Machine

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

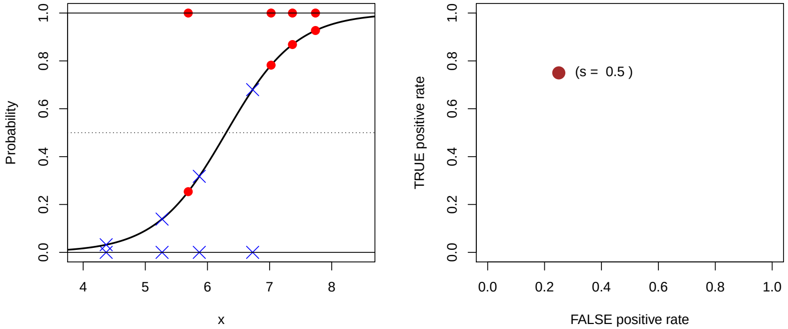
- The False Positive Rate :

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

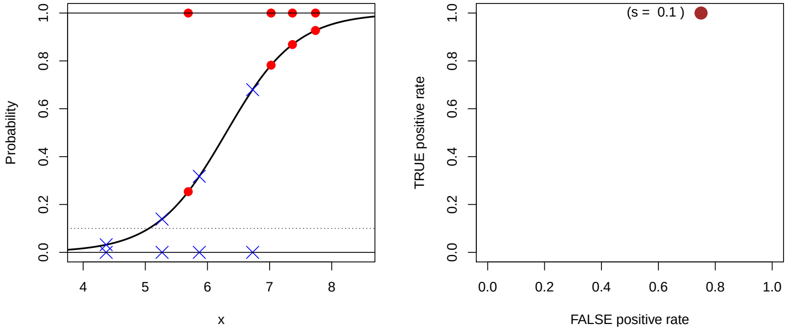
- The True Positive Rate :

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

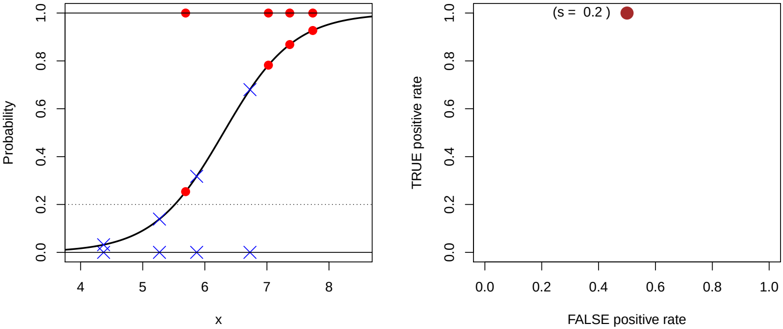
ROC curve - Illustration in 1D



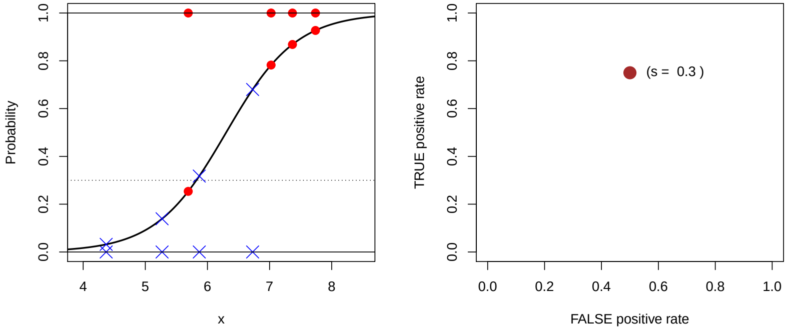
ROC curve - Illustration in 1D



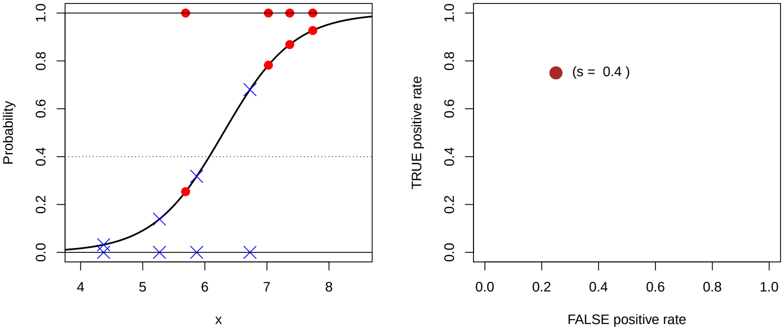
ROC curve - Illustration in 1D



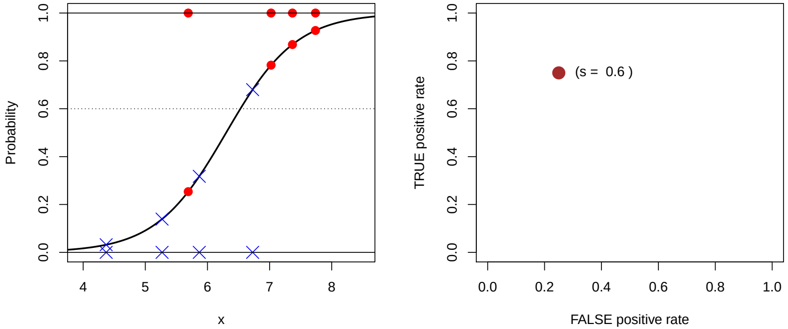
ROC curve - Illustration in 1D



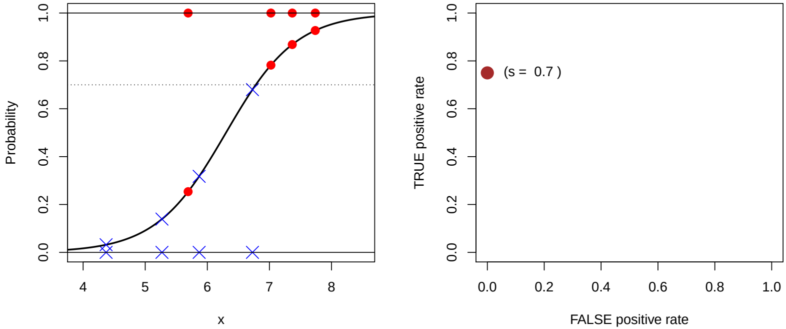
ROC curve - Illustration in 1D



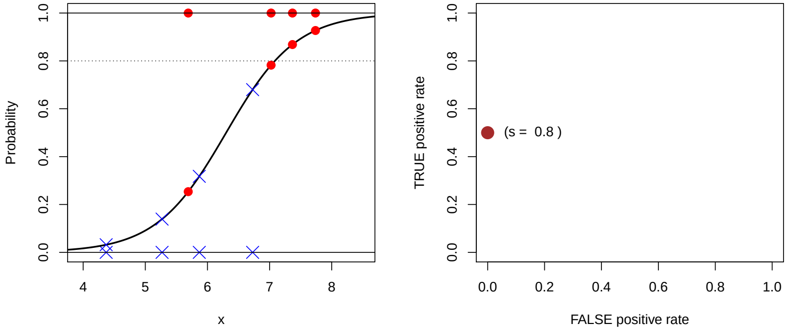
ROC curve - Illustration in 1D



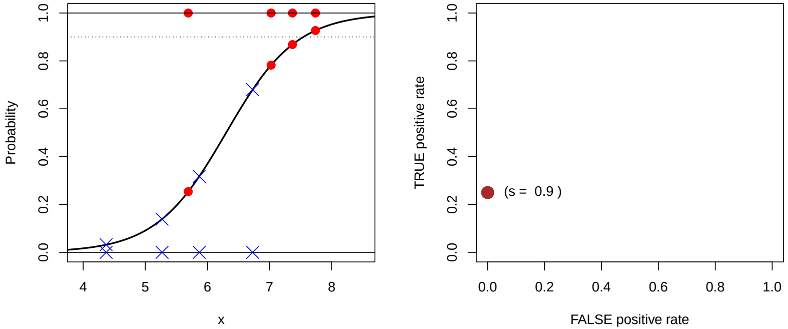
ROC curve - Illustration in 1D



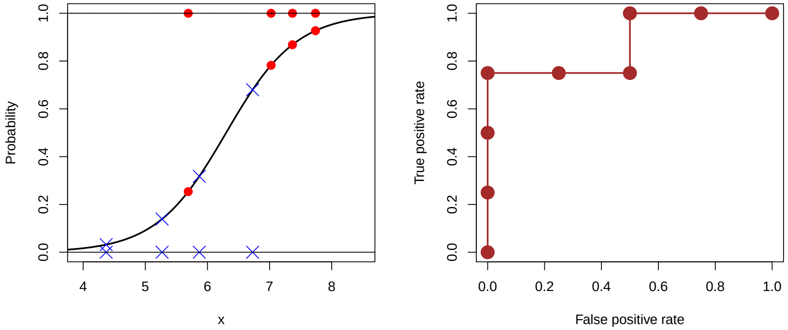
ROC curve - Illustration in 1D



ROC curve - Illustration in 1D



ROC curve - Illustration in 1D



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

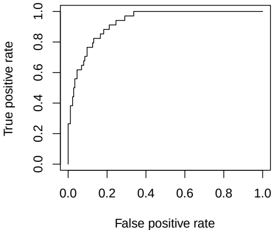


FIGURE – Ozone : ROC curve for logistic regression.

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.

Outline

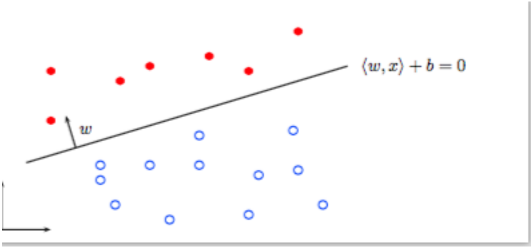
- Support Vector Machines.
  - Linear SVM in the separable case
  - Linear SVM in the non separable case
  - Non linear SVM and kernels
  - Conclusion

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.

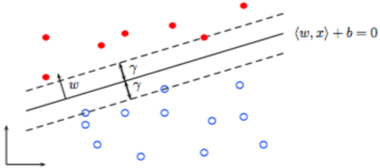


- The function $f_{w,b}(x) = \mathbb{1}_{\langle w,x\rangle + b \geq 0} - \mathbb{1}_{\langle w,x\rangle + b < 0}$ defines a possible linear classification rule.

- 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 $x_1, \dots, x_k$ if

$$\mathrm{min}_{i=1\dots k} \left| \langle w, x_i \rangle + b \right| = 1.$$

- 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

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

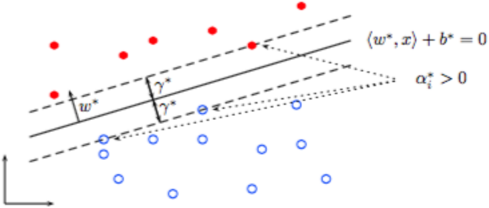
under the constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\alpha_i \geq 0 \ \forall i$.
The solution α^* of the dual problem can be obtained with classical optimization softwares.
Remark : The solution does not depend on the dimension d, but depends on the sample size n, hence it is interesting to 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
  - $\alpha_i^* \geq 0 \; \forall i = 1 \ldots n$.
  - $y_i \left(\langle w^*, x_i \rangle + b^* \right) \geq 1 \; \forall i = 1 \ldots n$.
  - $\alpha_i^* \left(y_i \left(\langle w^*, x_i \rangle + b^* \right) - 1 \right) = 0 \; \forall \; i = 1 \ldots n$.
    (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

- $w^* = \sum_{i=1}^{n} \alpha_i^* x_i y_i$,
- $b^* = -\frac{1}{2}\left\{\min_{y_i=1}\langle w^*, x_i\rangle + \min_{y_i=-1}\langle w^*, x_i\rangle\right\}$.

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

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

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.
- One can also minimize $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i^k$, $k = 2, 3, \ldots$, we still have a **convex optimization problem**.
  The choice $\sum_{i=1}^n \mathbb{1}_{\xi_i > 1}$ (number of errors) instead of $\sum_{i=1}^n \xi_i^k$ would lead to a non convex optimization problem.

The **Lagrangian** of this problem is :

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

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 :

$$\begin{aligned}&\alpha_i^* \left(y_i \left(\langle w^*, x_i\rangle + b^*\right) + \xi_i^* - 1\right) = 0 \ \forall\ i = 1 \ldots n,\\ &\xi_i^*(\alpha_i^* - C) = 0\end{aligned}$$

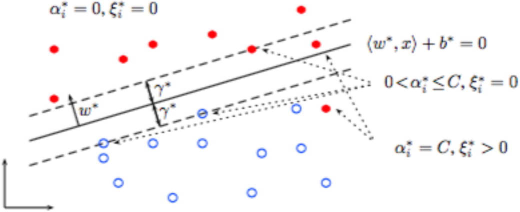
Definition

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 $\xi_i^* = 0$.


$\alpha_i^* = 0, \xi_i^* = 0$
$\langle w^*, x\rangle + b^* = 0$
$0 < \alpha_i^* \leq C, \xi_i^* = 0$
$\alpha_i^* = C, \xi_i^* > 0$
w^*
γ^*
γ^*

The classification rule is defined by

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

with

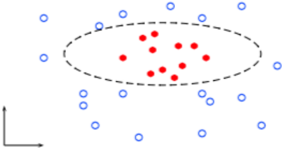
- $w^* = \sum_{i=1}^n \alpha_i^* x_i y_i,$
- b^* such that $y_i\left(\langle w^*, x_i\rangle + b^*\right) = 1 \;\forall x_i,\; 0 < \alpha_i^* < C.$

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. For example, for $x = (x_1, x_2) \in \mathbb{R}^2$, $\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, and $k(x, x') = \langle x, x' \rangle^2$.
- 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 $\{x_1, \dots, x_n\}$ in $\mathcal{X}$, the matrix $(k(x_i, x_j))_{1 \leq i,j \leq n}$ is positive definite :*

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

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 $k(x, x') = k_1(x, x') + k_2(x, x')$, $\lambda k_1(x, x')$, $k_1(x, x')k_2(x, x')$, $f(x)f(x')$, $k_1(\phi(x), \phi(x'))$, $x^T Bx'$, $P(k_1(x, x'))$, 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.*