Linear and nonlinear methods for classification
Support Vector Machines

1 Introduction

In this chapter, we 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. For example, the MNIST data set is a database of handwritten digits, where the objects $\boldsymbol{X}_i$ are images and $Y_i \in \{0, 1, \ldots, 9\}$. Many other examples can be considered, such as the recognition of an object in an image, the detection of spams for emails, the presence of some illness for patients (the observation points may be gene expression data) ... We will first introduce the notion of best classifier, which is also called the Bayes classifier, we will then propose linear methods for classification. The core of the chapter will be devoted to the Support Vector machine, which are very powerful nonlinear methods for classification.

The main references for this course are the following books :

- An introduction to Support Vector Machines by N. Cristianini and J. Shawe-Taylor [1]
- Introduction to High-Dimensional Statistics by C. Giraud [3]
- The elements of Statistical Learning by T. Hastie et al [4].
- Learning with kernels by A. Smola and B. Scholkopf [5].
- Statistical Learning Theory by V. Vapnik [6].
- M2 courses, M. Fromont-Renoir, Université de Rennes 2 : https://perso.univ-rennes2.fr/magalie.fromont

2 Optimal rules for classification

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 $\mathcal{X} \times \mathcal{Y}$, and that $\boldsymbol{x}$ is one observation of the variable $\boldsymbol{X}$, where $(\boldsymbol{X}, Y)$ has joint distribution P and is *independent* of $\boldsymbol{D}^n$. Since we consider a classification problem, $\mathcal{Y}$ is a finite set. The sample $\boldsymbol{D}^n$ is called the *learning sample*.

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

If f is a classification rule, $\boldsymbol{x}$ an input, y the output that is really associated to the input $\boldsymbol{x}$, then $l(y, f(\boldsymbol{x}))$ quantifies the loss when we associate to the input $\boldsymbol{x}$ the predicted output $f(\boldsymbol{x})$.

In a regression framework when $\mathcal{Y} = \mathbb{R}$: we consider the $\mathbb{L}^p$ loss ($p \geq 1$)

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

If $p = 2$, this is the quadratic loss.

For binary classification : $\mathcal{Y} = \{-1, 1\}$

$$l(y, y') = \mathbf{1}_{y \neq y'} = \frac{|y - y'|}{2} = \frac{(y - y')^2}{4}.$$

We consider the expectation of this loss, this leads to the definition of the *risk* :

DEFINITION 2. — *Given a loss function l, the* risk - *or* generalisation error - *of a prediction rule f is defined by*

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

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 ? This is indeed the case. We focus here on the classification framework. We define the Bayes rule, and show that it is an optimal rule for classification.

DEFINITION 3. — *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 4. — *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.

Exercise. — Prove Theorem 4.

3 Linear discriminant analysis

Let $(\boldsymbol{X}, Y)$ with unknown distribution P on $\mathcal{X} \times \mathcal{Y}$, where we assume that $\mathcal{X} = \mathbb{R}^p$ and $\mathcal{Y} = \{1, 2, \ldots, K\}$. We define

$$f_k(\boldsymbol{x}) = \mathbb{P}(Y = k / \boldsymbol{X} = \boldsymbol{x}).$$

A Bayes rule is defined by

$$f^*(\boldsymbol{x}) = \underset{k \in \{1, 2, \ldots, K\}}{\operatorname{argmax}} f_k(\boldsymbol{x}).$$

We assume that the distribution of $\boldsymbol{X}$ has a density $f_{\boldsymbol{X}}$ and the distribution of $\boldsymbol{X}$ given $Y = k$ has a density g_k with respect to the Lebesgue measure on $\mathbb{R}^p$, and we set $\pi_k = \mathbb{P}(Y = k)$.

Exercise. — Prove that

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{l=1}^{K} g_l(\boldsymbol{x}) \pi_l$$

and that

$$f_k(\boldsymbol{x}) = \frac{g_k(\boldsymbol{x}) \pi_k}{\sum_{l=1}^{K} g_l(\boldsymbol{x}) \pi_l}.$$

If we assume that the distribution of $\boldsymbol{X}$ given $Y = k$ is a multivariate normal distribution, with mean μ_k and covariance matrix Σ_k, we have

$$g_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \mu_k)' \Sigma_k^{-1} (\boldsymbol{x} - \mu_k) \right).$$

For the linear discriminant analysis, we furthermore assume that $\Sigma_k = \Sigma$ for all k. In this case we have

$$\begin{aligned} \log\left(\frac{\mathbb{P}(Y = k / \boldsymbol{X} = \boldsymbol{x})}{\mathbb{P}(Y = l / \boldsymbol{X} = \boldsymbol{x})} \right) &= \log\left(\frac{\pi_k}{\pi_l} \right) + \boldsymbol{x}' \Sigma^{-1} (\mu_k - \mu_l) \\ &- \frac{1}{2} (\mu_k + \mu_l)' \Sigma^{-1} (\mu_k - \mu_l). \end{aligned}$$

Hence the decision boundary between the class k and the class l, $\{\boldsymbol{x}, \mathbb{P}(Y = k / \boldsymbol{X} = \boldsymbol{x}) = \mathbb{P}(Y = l / \boldsymbol{X} = \boldsymbol{x})\}$ is linear. We can write

$$\log \mathbb{P}(Y = k / \boldsymbol{X} = \boldsymbol{x}) = C(\boldsymbol{x}) \delta_k(\boldsymbol{x})$$

where $C(\boldsymbol{x})$ does not depend on the class k, and

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}' \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k' \Sigma^{-1} \mu_k + \log(\pi_k).$$

The Bayes rule will assign $\boldsymbol{x}$ to the class $f^*(\boldsymbol{x})$ which maximises $\delta_k(\boldsymbol{x})$.
We want now to built a decision rule from a training sample $\boldsymbol{D}^n =$

$\{(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)\}$ which is close to the Bayes rule. For this purpose, we have to estimate for all k π_k, μ_k and the matrix Σ. We consider the following estimators.

$$\begin{aligned}\hat{\pi}_k &= \frac{N_k}{n}\\ \hat{\mu}_k &= \frac{\sum_{i=1}^n \boldsymbol{X}_i \mathbf{1}_{Y_i=k}}{N_k}\end{aligned}$$

where $N_k = \sum_{i=1}^n \mathbf{1}_{Y_i=k}$. We estimate Σ by

$$\hat{\Sigma} = \sum_{k=1}^K \sum_{i=1}^n \frac{(\boldsymbol{X}_i - \hat{\mu}_k)(\boldsymbol{X}_i - \hat{\mu}_k)' \mathbf{1}_{Y_i=k}}{n-K}.$$

To conclude, the Linear Discriminant Analysis assigns the input $\boldsymbol{x}$ to the class $\hat{f}(\boldsymbol{x})$ which maximises $\hat{\delta}_k(\boldsymbol{x})$, where we have replaced in the expression of $\delta_k(\boldsymbol{x})$ the unknown quantities by their estimators.
Remark : If we no more assume that the matrix Σ does not depend on the class k, we obtain quadratic discriminant functions

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\boldsymbol{x} - \mu_k)'\Sigma_k^{-1}(\boldsymbol{x} - \mu_k) + \log(\pi_k).$$

This leads to the quadratic discriminant analysis.

4 Logistic regression

Noting that the Bayes classifier only depends on the conditional distribution of Y given $\boldsymbol{X}$, we can avoid to model the distribution of $\boldsymbol{X}$ as previously. We assume that $\mathcal{X} = \mathbb{R}^d$. One of the most popular model for binary classification when $\mathcal{Y} = \{-1, 1\}$ is the **logistic regression** model, for which it is assumed that

$$\mathbb{P}(Y = 1/\boldsymbol{X} = \boldsymbol{x}) = \frac{\exp(\alpha + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)}{1 + \exp(\alpha + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)} \text{ for all } x \in \mathcal{X},$$

with $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^d$.

Exercise. — Compute the Bayes classifier f^* for this model and determine the border between $f^* = 1$ and $f^* = -1$.

We can estimate the parameters $(\alpha, \boldsymbol{\beta})$ by maximizing the conditional likelihood of Y given $\boldsymbol{X}$.

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

We then compute the logistic regression classifier :

$$\forall x \in \mathcal{X}, \hat{f}(\boldsymbol{x}) = \text{sign}(\hat{\alpha} + \langle \hat{\boldsymbol{\beta}}, \boldsymbol{x} \rangle).$$

In general, the logistic regression provides different classifiers as LDA. If the distribution of $\boldsymbol{X}$ if far from a Gaussian distribution, the logistic regression outperforms the LDA, as soon as the logistic model is still valid.

5 Linear Support Vector Machine

5.1 Linearly separable training set

We assume that $\mathcal{X} = \mathbb{R}^d$, endowed with the usual scalar product $\langle ., . \rangle$, and that $\mathcal{Y} = \{-1, 1\}$.

DEFINITION 5. — *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

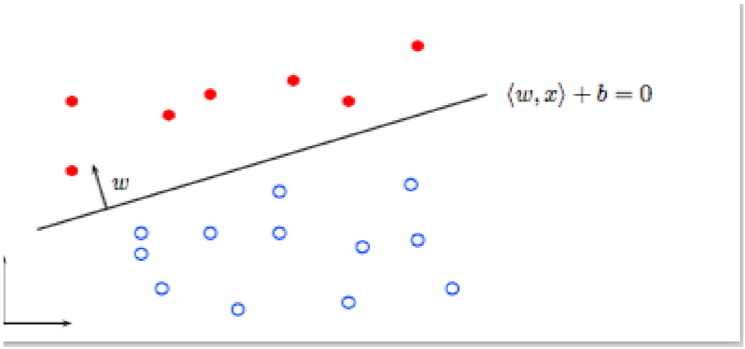
$$\forall i \; y_i \left(\langle w, x_i \rangle + b \right) > 0.$$

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

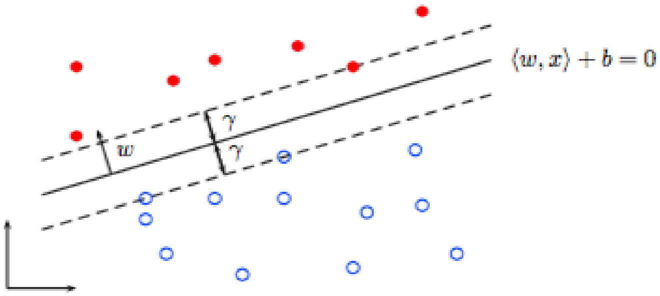
The function $f_{w,b}(x) = \mathbf{1}_{\langle w,x \rangle + b \geq 0} - \mathbf{1}_{\langle w,x \rangle + b < 0}$ defines a possible linear classification rule.

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 [6]. The classification rule with the best generalization properties cooresponds 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 broder 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 :

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

Let us notice that for all $\kappa \neq 0$, the couples $(\kappa w, \kappa b)$ and (w, b) define the same hyperplane.

DEFINITION 6. — *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

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

5.2 A convex optimisation problem

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.

Primal optimization problem Let us recall some definitions.

We want to minimize for $u \in \mathbb{R}^2$ $h(u)$ under the constraints $g_i(u) \geq 0$ for $i = 1 \ldots n$, h is a quadratic function, g_i are affine functions.

The **Lagrangian** of the optimization problem is the function defined on $\mathbb{R}^2 \times \mathbb{R}^n$ by

$$L(u,\alpha) = h(u) - \sum_{i=1}^{n} \alpha_i g_i(u).$$

The variables α_i are called the **dual variables** .
For all $\alpha \in \mathbb{R}^n$, u_α is the value of u minimizing $L(u,\alpha)$.
The **dual function** is defined by

$$\theta(\alpha) = L(u_\alpha, \alpha) = \min_{u\in\mathbb{R}^2} L(u,\alpha).$$

Dual optimization problem

Maximizing $\theta(\alpha) = L(u_\alpha,\alpha) = \min_{u\in\mathbb{R}^2} L(u,\alpha)$ under the constraints $\alpha_i \geq 0$ for $i = 1\dots n$.

The solution of the dual problem α^* gives the solution of the primal problem:

$$u^* = u_{\alpha^*}.$$

Karush-Kuhn-Tucker conditions for the dual problem

- $\alpha_i^* \geq 0$ for all $i = 1\dots n$.
- $g_i(u_{\alpha^*}) \geq 0$ for all $i = 1\dots n$.
- Back to the dual problem :

We have to minimize $L(u,\alpha) = h(u) - \sum_{i=1}^n \alpha_i g_i(u)$ with respect to u and to maximize $L(u_\alpha,\alpha)$ with respect to the dual variables α_i. Note that if $g_i(u_{\alpha^*}) > 0$, then we get $\alpha_i^* = 0$.
The complementary Karush-Kuhn-Tucker condition writes $\alpha_i^* g_i(u_{\alpha^*}) = 0$.

Let us come back to the linear support vector machine optimization problem.

The primal problem to solve is :

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

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.

5.3 Supports Vectors

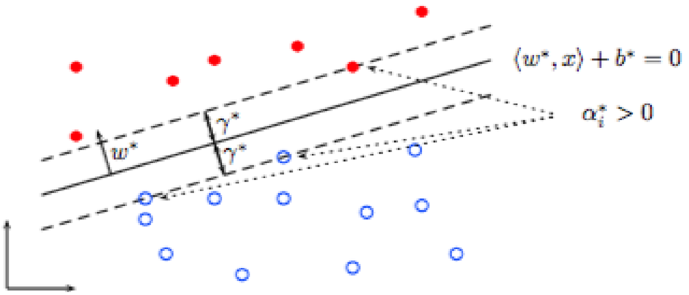
For our optimization problem, the **Karush-Kuhn-Tucker conditions** are

- $\alpha_i^* \geq 0\ \forall i = 1\dots n.$

- $y_i\left(\langle w^*, x_i\rangle + b^*\right) \geq 1 \; \forall i = 1 \dots n.$
- $\alpha_i^* \left(y_i\left(\langle w^*, x_i\rangle + b^*\right) - 1\right) = 0 \; \forall \; i = 1 \dots n.$ (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 7. — *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 :

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

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

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

6 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.

6.1 Flexible margin

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 $\xi = (\xi_1, \dots, \xi_n)$ and the constraint $y_i(\langle w, x_i\rangle + b) \geq 1$ becomes $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$, 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**.

6.2 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 $\sum_{i=1}^n \mathbf{1}_{\xi_i>1}$ (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:

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

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 :

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

6.3 Supports vectors

We have the complementary Karush-Kuhn-Tucker conditions:

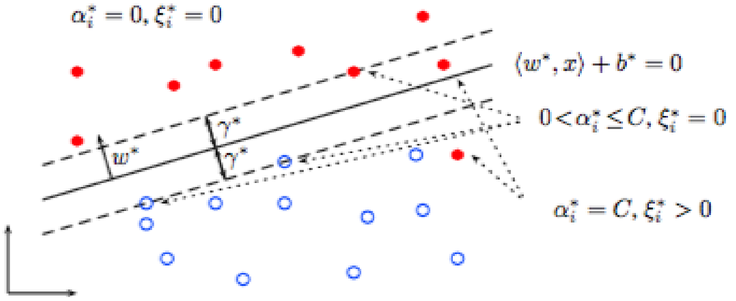
$$\begin{aligned} &\alpha_i^*\left(y_i\left(\langle w^*, x_i\rangle + b^*\right) + \xi_i^* - 1\right) = 0\ \forall\, i = 1 \ldots n, \\ &\xi_i^*(\alpha_i^* - C) = 0 \end{aligned}$$

DEFINITION 8. — *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 $\xi_i^* = 0$.


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

The classification rule is defined by

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

with

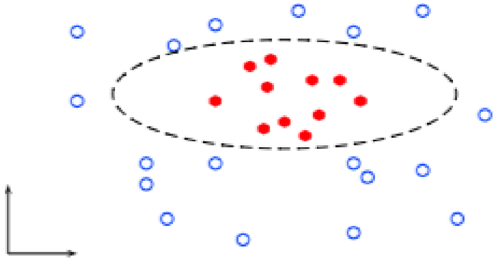
- $w^* = \sum_{i=1}^n \alpha_i^* x_i y_i$,
- b^* such that $y_i\left(\langle w^*, x_i\rangle + b^*\right) = 1 \; \forall x_i, \; 0 < \alpha_i^* < C$.

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

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

7 Non linear SVM and kernels

A training set is rarely linearly separable.



In this case, a linear SVM leads to bad performances and a high number of support vectors. We can make the classification procedure more flexible by enlarging the feature space and sending the entries $\{x_i, i = 1 \ldots n\}$ in an Hilbert space $\mathcal{H}$, with high or possibly infinite dimension, via a function ϕ, and we apply a linear SVM procedure on the new training set $\{(\phi(x_i), y_i), i = 1 \ldots n\}$. 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 $\mathbb{R}^4$, and a linear SVM is appropriate.

7.1 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

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

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 explicit computation of ϕ is not required.

DEFINITION 9. — *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 10. — **Mercer condition** *If the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is continuous, symmetric, and if for all finite subset $\{x_1, \ldots, x_k\}$ in $\mathcal{X}$, the matrix $(k(x_i, x_j))_{1 \leq i,j \leq k}$ is positive definite :*

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

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).

By way of example, let us precise the RKHS associated with the Gaussian kernel.

PROPOSITION 11. — *For any function* $h \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ *and* $\omega \in \mathbb{R}^d$, *we define the Fourier transform*

$$\boldsymbol{F}[f](\omega) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} f(t) e^{-i\langle \omega, t\rangle} dt.$$

For any $\sigma > 0$, *the functional space*

$$\mathcal{H}_\sigma = \{f \in C_0(\mathbb{R}^d) \cap L^1(\mathbb{R}^d) \text{ such that } \int_{\mathbb{R}^d} |\boldsymbol{F}[f](\omega)|^2 e^{\sigma|\omega|^2/2} d\omega < +\infty\}$$

endowed with the scalar product

$$\langle f, g\rangle_{\mathcal{H}_\sigma} = (2\pi\sigma^2)^{-d/2} \int_{\mathbb{R}^d} \overline{\boldsymbol{F}[f](\omega)} \boldsymbol{F}[g](\omega) e^{\sigma|\omega|^2/2} d\omega,$$

is the RKHS associated with the Gaussian kernel $k(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$.

Indeed, for all $x \in \mathbb{R}^d$, the function $k(x,.)$ belongs to $\mathcal{H}_\sigma$ and we have

$$\langle h, k(x,.)\rangle_{\mathcal{H}_\sigma} = \boldsymbol{F}^{-1}[\boldsymbol{F}[h]](x) = h(x).$$

The RKHS $\mathcal{H}_\sigma$ contains very regular functions, and the norm $\|h\|_{\mathcal{H}_\sigma}$ controls the smoothness of the function h. When σ increases, the functions of the RKHS become smoother. See A. Smola and B. Scholkopf [5] for more details on RKHS.

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 ...

7.2 Minimization of the convexified empirical risk

The ideal classification rule is the one which minimizes the risk $L(f) = \mathbb{P}(Y \neq f(X))$, we have seen that the solution is the Bayes rule f^*. A classical way in nonparametric estimation or classification problems is to replace the risk by the empirical risk and to minimize the empirical risk :

$$L_n(f) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{Y_i \neq f(X_i)}.$$

In order to avoid overfitting, the minimization is restricted to a set $\mathcal{F}$:

$$\hat{f} = \text{argmin}_{f \in \mathcal{F}} L_n(f).$$

The risk of $\hat{f}$ can be decomposed in two terms :

$$0 \leq L(\hat{f}) - L(f^*) = \min_{f \in \mathcal{F}} L(f) - L(f^*) + L(\hat{f}) - \min_{f \in \mathcal{F}} L(f).$$

The first term $\min_{f \in \mathcal{F}} L(f) - L(f^*)$ is the approximation error, or bias term, the second term $L(\hat{f}) - \min_{f \in \mathcal{F}} L(f)$ is the stochastic error or variance term. Enlarging the class $\mathcal{F}$ reduces the approximation error but increases the stochastic error.
The empirical risk minimization classifier cannot be used in practice because of its computational cost, indeed L_n is not convex. This is the reason why we generally replace the empirical misclassification probability L_n by some convex surrogate, and we consider convex classes $\mathcal{F}$. We consider a loss function l, and we require the condition $l(z) \geq \mathbf{1}_{z<0}$, which will allow to give an upper bound for the misclassification probability; indeed

$$\mathbb{E}(l(Yf(X))) \geq \mathbb{E}(\mathbf{1}_{Yf(X)<0}) = \mathbb{P}(Y \neq f(X)).$$

Classical convex losses l are the hinge loss $l(z) = (1-z)_+$, the exponential loss $l(z) = \exp(-z)$, the logit loss $l(z) = \log_2(1 + \exp(-z))$.

Let us show that SVM are solutions of the minimization of the convexified and penalized empirical risk. For the sake of simplicity, we consider the linear case.

We first notice that the following optimization problem : Minimizing $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i$s. t. $\begin{cases} y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i \;\forall\, i \\ \xi_i \geq 0 \end{cases}$

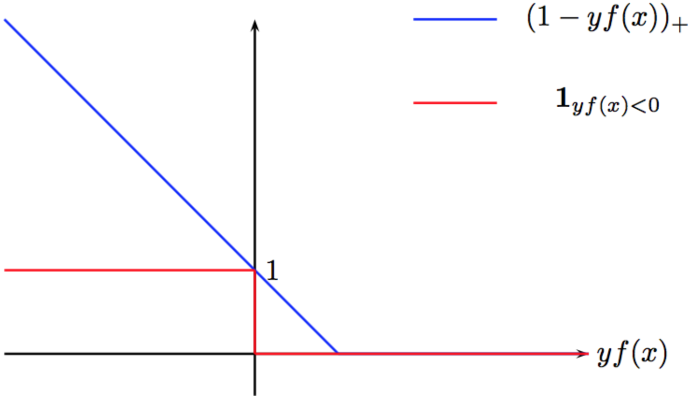
is equivalent to minimize

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \left(1 - y_i\left(\langle w, x_i\rangle + b\right)\right)_+ ,$$

or equivalently

$$\frac{1}{n}\sum_{i}^{n} \left(1 - y_i\left(\langle w, x_i\rangle + b\right)\right)_+ + \frac{1}{2Cn}\|w\|^2.$$

$\gamma(w, b, x_i, y_i) = \left(1 - y_i\left(\langle w, x_i\rangle + b\right)\right)_+$ is a convex upper bound of the empirical risk $\mathbf{1}_{y_i(\langle w, x_i\rangle + b)<0}$



Hence, SVM are solutions of the minimization of the convexified empirical risk with the hinge loss l plus a penalty term. Indeed, SVM are solutions of

$$\text{argmin}_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n l(y_i f(x_i)) + \text{pen}(f),$$

where

$$\mathcal{F} = \{\langle w, x\rangle + b, w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

and

$$\forall f \in \mathcal{F}, \text{pen}(f) = \frac{1}{2Cn}\|w\|^2.$$

8 Agregation of classifiers with Adaboost

Adaboost (or adaptive boosting) is a boosting method introduced by Freund and Schapire [2] to combine several classifiers $f_1, \dots, f_k$, for example SVM obtained with different kernels or different penalty terms. The principle of the algorithm is to minimize the empirical risk for the exponential loss function over the linear space $\mathcal{F}$ generated by the classifiers $f_1, \dots, f_k$. The aim is to compute

$$\hat{f} = \underset{f \in \text{span}(f_1,\dots,f_k)}{\text{argmin}} \{\frac{1}{n}\sum_{i=1}^{n} \exp(-Y_i f(X_i))\}.$$

To approximate the solution, Adaboost computes a sequence of functions $\hat{f}_m$ for $m = 0, \dots M$ with

$$\begin{aligned} \hat{f}_0 &= 0 \\ \hat{f}_m &= \hat{f}_{m-1} + \beta_m h_{j_m} \end{aligned}$$

where (β_m, j_m) minimizes the empirical convexified risk

$$\underset{\beta \in \mathbb{R}, j=1,\dots,p}{\text{argmin}} \{\frac{1}{n}\sum_{i=1}^{n} \exp(-Y_i(\hat{f}_{m-1}(X_i) + \beta h_j))\}.$$

The final classification rule is given by

$$\hat{f} = \text{sign}(\hat{f}_M).$$

Exercise. — We denote

$$w_i^{(m)} = \frac{1}{n} \exp(-Y_i \hat{f}_{m-1}(X_i))$$

and we assume that for all $j = 1, \dots, p$,

$$err_m(j) = \frac{\sum_{i=1}^{n} w_i^{(m)} \mathbf{1}_{h_j(X_i) \neq Y_i}}{\sum_{i=1}^{n} w_i^{(m)}} \in]0, 1[.$$

Prove that

$$j_m = \underset{j=1,\dots,p}{\text{argmin}}\, err_m(j),$$

and

$$\beta_m = \frac{1}{2} \log\left(\frac{1 - err_m(j)}{err_m(j)}\right).$$

This leads to the AdaBoost algorithm :

- $w_i^{(1)} = 1/n$ for $i = 1, \dots, n$.
- For $m = 1, \dots, M$

$$\begin{aligned} j_m &= \underset{j=1,\dots,p}{\text{argmin}}\, err_m(j) \\ \beta_m &= \frac{1}{2}\log\left(\frac{1 - err_m(j)}{err_m(j)}\right) \\ w_i^{(m+1)} &= w_i^{(m)} \exp(2\beta_m \mathbf{1}_{f_{j_m}(X_i) \neq Y_i} - \beta_m) \text{ for } i = 1, \dots, n \end{aligned}$$

- $\hat{f}_M(x) = \sum_{m=1}^{M} \beta_m f_{j_m}(x)$.
- $\hat{f} = \text{sign}(\hat{f}_M)$.

In the above computation, note that, since $f_{j_m}(X_i) \in \{-1, 1\}$, $Y_i f_{j_m}(X_i) = 2\mathbf{1}_{f_{j_m}(X_i) \neq Y_i} - 1$.

9 Regression and kernels

Although the framework of the chapter is classification, let us mention that kernel methods can also be used for regression function estimation. We present here the Kernel Regression Least Square procedure. It is based on a penalized least square criterion. Let $(\boldsymbol{X_i}, Y_i)_{1 \leq i \leq n}$ the observations, with $\boldsymbol{X_i} \in \mathbb{R}^p$, $Y_i \in \mathbb{R}$. We consider a positive definite kernel k defined on $\mathbb{R}^p$:

$$k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{y}, \boldsymbol{x}); \quad \sum_{i,j=1}^{n} c_i c_j k(\boldsymbol{X_i}, \boldsymbol{X_j}) \geq 0.$$

We are looking for a predictor of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} c_j k(\boldsymbol{X_j}, \boldsymbol{x}), \ \boldsymbol{c} \in \mathbb{R}^n.$$

Let us denote by $\boldsymbol{K}$ the matrix defined by $\boldsymbol{K}_{i,j} = k(\boldsymbol{X_i}, \boldsymbol{X_j})$. The KRLS method consists in minimizing for f on the form defined above the penalized least square criterion

$$\sum_{i=1}^{n}(Y_i - f(\boldsymbol{X_i}))^2 + \lambda \|f\|^2_{\boldsymbol{K}},$$

where

$$\|f\|^2_{\boldsymbol{K}} = \sum_{i,j=1}^{n} c_i c_j k(\boldsymbol{X_i}, \boldsymbol{X_j}).$$

Equivalently, we minimize for $c \in \mathbb{R}^n$ the criterion

$$\|\boldsymbol{Y} - \boldsymbol{K}\boldsymbol{c}\|^2 + \lambda \boldsymbol{c}'\boldsymbol{K}\boldsymbol{c}.$$

There exists an explicit solution

$$\hat{\boldsymbol{c}} = (\boldsymbol{K} + \lambda I_n)^{-1} Y,$$

which leads to the predictor

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{n} \hat{c}_j k(\boldsymbol{X_j}, \boldsymbol{x}).$$

$$\hat{\boldsymbol{Y}} = \boldsymbol{K}\hat{\boldsymbol{c}}.$$

With a kernel corresponding to the scalar product, we recover a linear predictor

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X'}, \hat{\boldsymbol{c}} = (\boldsymbol{X}\boldsymbol{X'} + \lambda I_n)^{-1} Y,$$

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{n} \hat{c}_j \langle \boldsymbol{X_j}, \boldsymbol{x} \rangle.$$

For polynomial or Gaussian kernels for example, we obtain non linear predictors. As for SVM, an important interest of this method is the possibility to be generalized to complex predictors such as text, graphs, DNA sequences .. as soon as one can define a kernel function on such objects.

10 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 [6].
- 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 $\mathcal{X}$.
- 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

[1] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, 2000.

[2] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: proceedings of the Thirteenth International Conference*, pages 148–156. Morgan Kaufman, 1996. San Francisco.

[3] C. Giraud. *Introduction to high-dimensional statistics*, volume 139 of *Monographs on Statistics and Applied Probability*. CRC Press, Boca Raton, FL, 2015.

[4] T. Hastie, R. Tibshirani, and J Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2009. Second edition.

[5] B. Scholkopf and A. Smola. *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[6] V.N. Vapnik. *Statistical learning theory*. Wiley Inter science, 1999.