Pattern recognition
Advanced decision methods

Support vector machines

- Chapitre 3 -

ERM Principle :
This is to minimize the following functional risk :

$$P_e(d) = \int \frac{1}{2} |y - d(\boldsymbol{x}; \boldsymbol{w}, b)| \, p(\boldsymbol{x}, y) \, d\boldsymbol{x} \, dy.$$

The probability density function $p(\boldsymbol{x}, y)$ is unknown.
To minimize $P_e(d)$ we minimize its estimator, the empirical risk :

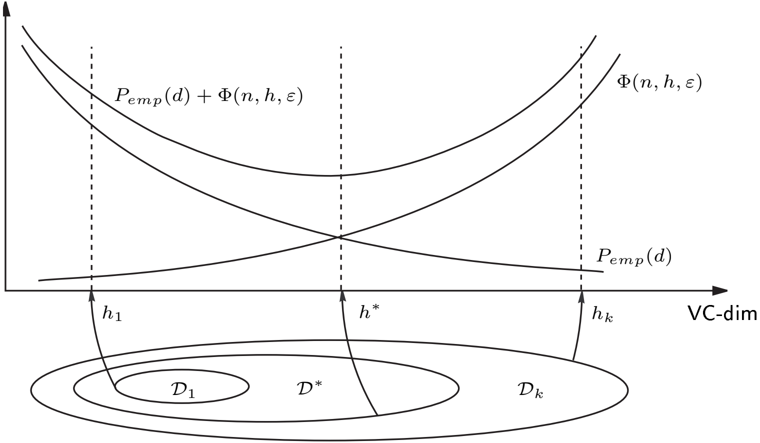
$$P_{emp}(d) = \frac{1}{2n} \sum_{i=1}^{n} |y_i - d(\boldsymbol{x}_i; \boldsymbol{w}, b)|$$

can be computed with the training data set $\mathcal{A}_n$.

SRM Principle :
With a probability $1-\varepsilon$, the following relation holds :

$$P_e(d) \leq P_{emp}(d) + \sqrt{\frac{h\,\ln\left(\frac{2n}{h}+1\right) - \ln\frac{\varepsilon}{4}}{n}}.$$

Instead of minimizing $P_{emp}(d)$,
the structural risk minimization principle suggests to minimize, with respect to h, the upper bound of the guaranteed risk $P_{emp}(d) + \Phi(n, h, \varepsilon)$.


$P_{emp}(d) + \Phi(n, h, \varepsilon)$
$\Phi(n, h, \varepsilon)$
$P_{emp}(d)$
h_1
h^*
h_k
VC-dim
$\mathcal{D}_1$
$\mathcal{D}^*$
$\mathcal{D}_k$

Perceptron algorithm

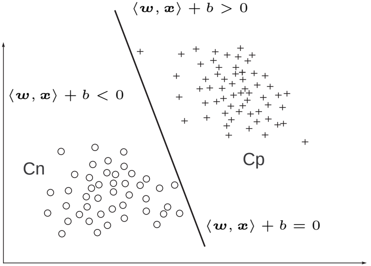
The Perceptron algorithm is meant to produce a solution of minimum training error by minimizing the following empirical risk :

$$(\boldsymbol{w}^*, b^*) = \arg \min_{(\boldsymbol{w},b)} \sum_{i=1}^{n} |y_i - d(\boldsymbol{x}_i; \boldsymbol{w}, b)|.$$

▷ *Why the obtained solution would have the best performance ?*

▷ *Is the minimization of the empirical error a good idea ?*

▷ *Is there any other possible approach ?*


$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0$
Cn
Cp
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$

Considering the two classes classification problem, using a set of n data $\boldsymbol{x}_i \in \mathbb{R}^l$ associated with n labels y_i, we have a data set :

$$\mathcal{A}_n = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}.$$

Assuming $y_i = (-1)$ if $\boldsymbol{x}_i \in \omega_0$, and $y_i = (+1)$ if $\boldsymbol{x}_i \in \omega_1$.

A **linear classifier** is defined as :

$$d(\boldsymbol{x}; \boldsymbol{w}, b) = \mathsf{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b).$$

Considering the two classes classification problem, using a set of n data $\boldsymbol{x}_i \in \mathbb{R}^l$ associated with n labels y_i, we have a data set :

$$\mathcal{A}_n = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}.$$

Assuming $y_i = (-1)$ if $\boldsymbol{x}_i \in \omega_0$, and $y_i = (+1)$ if $\boldsymbol{x}_i \in \omega_1$.

The equation of a hyperplane is defined to a multiplicative constant :

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \quad \Longleftrightarrow \quad \langle \gamma\, \boldsymbol{w}, \boldsymbol{x} \rangle + \gamma\, b = 0, \quad \gamma \in \mathbb{R}^*$$

The classes ω_0 and ω_1 are **linearly separable** if there exists $\boldsymbol{w}$ and b such that :

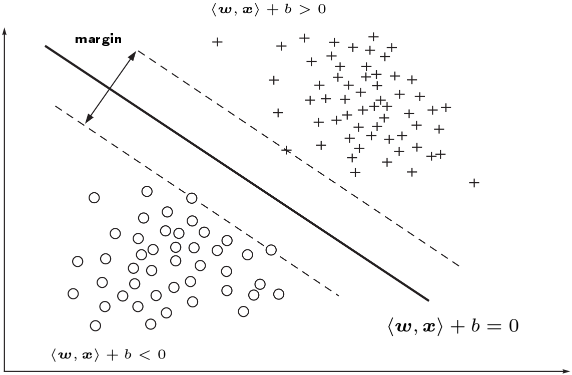
$$\begin{aligned} \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b &\geq +1 \qquad \forall \boldsymbol{x}_i \in \omega_1 \\ \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b &\leq -1 \qquad \forall \boldsymbol{x}_i \in \omega_0 \end{aligned}$$

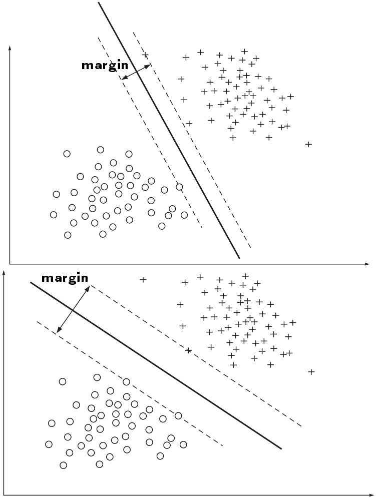
In what follows, this separability criteria is summarized as :

$$y_i\,(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \geq 0 \qquad \forall (\boldsymbol{x}_i, y_i) \in \mathcal{A}_n$$

A new induction principle

Among separators leading to a minimum empirical error, you should choose the one that *maximizes the margin* (Vapnik 1965, 1992).

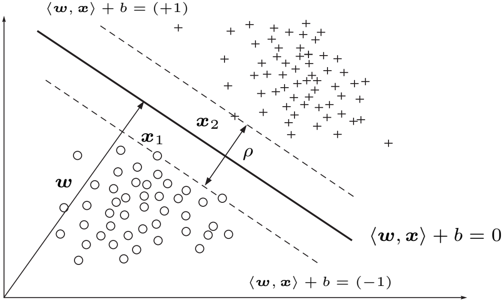

$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0$
margin
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0$



Weak margin *: probably low generalization performance*

Large margin *: probably good generalization performance*

This will be justified more rigorously now.


$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = (+1)$
$\boldsymbol{x}_2$
$\boldsymbol{x}_1$
ρ
$\boldsymbol{w}$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = (-1)$

We have $\langle \boldsymbol{w}, \boldsymbol{x}_2 \rangle + b = (+1)$ and $\langle \boldsymbol{w}, \boldsymbol{x}_1 \rangle + b = (-1)$.
It directly follows that :

$$\rho = \left\langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \boldsymbol{x}_2 - \boldsymbol{x}_1 \right\rangle = \frac{2}{\|\boldsymbol{w}\|}$$

The justification for maximizing the margin ρ, basic principle of SVM, is based on the following results from the statistical learning theory.

Theorem

Consider the hyperplanes of the form $\langle \boldsymbol{w}, \boldsymbol{x} \rangle = 0$, *where* $\boldsymbol{w}$ *is normalized so that it is in canonical form with respect to* $\mathcal{A}_n$. *Thus :*

$$\min_{\boldsymbol{x} \in \mathcal{A}_n} |\langle \boldsymbol{w}, \boldsymbol{x} \rangle| = 1.$$

The set of decision functions $d(\boldsymbol{x}; \boldsymbol{w}) = \mathit{sgn}\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ *defined based on* $\mathcal{A}_n$ *and satisfying the constraint* $\|\boldsymbol{w}\| \leq \Lambda$ *has a VC-dimension* h *which satisfies :*

$$h \leq R^2 \Lambda^2,$$

where R *is the radius of the smallest sphere centered on the origin that contains* $\mathcal{A}_n$.

Therefore, the larger $\rho = 2/\|\boldsymbol{w}\|$ is, the smaller h is ; which is better.

Maximizing the margin, defined by $\rho = \frac{2}{\|\boldsymbol{w}\|}$, is equivalent to minimizing $\|\boldsymbol{w}\|^2$.
To implement the SRM principle we have to solve the following optimization problem :

Minimize $\frac{1}{2}\|\boldsymbol{w}\|^2$

under the constrains $y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\right) \geq 1, \qquad 1 \leq i \leq n.$

Remark. This formulation is valid for linearly separable classes.

Optimization problems

Problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

Assume that :

- $f(\boldsymbol{x})$ is continu,
- $\frac{\partial f}{\partial x_i}$ is continu $\forall i$ and $\boldsymbol{x} \in \mathcal{X}$
- $\frac{\partial^2 f}{\partial x_i x_j}$ is continu $\forall i, j$ and $\boldsymbol{x} \in \mathcal{X}$

Local optimality

A necessary and suffisant condition for $\boldsymbol{x}^*$ to be a local minima is that :

1. $\nabla f(\boldsymbol{x}^*) = 0$ - stationarity
2. the hessian $\nabla^2 f(\boldsymbol{x}^*) = [\frac{\partial^2 f}{\partial x_i x_j}(\boldsymbol{x}^*)]$ is a semi-definite positive matrix

Proof :

Second order Taylor development near $\boldsymbol{x}^*$:
$f(\boldsymbol{x}) = f(\boldsymbol{x}^*) + \nabla f^T(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) + (\boldsymbol{x} - \boldsymbol{x}^*)^T \nabla^2 f(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) + \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \mathcal{O}(\boldsymbol{x} - \boldsymbol{x}^*)$.
Use $\boldsymbol{x} = \boldsymbol{x}^* - \theta \nabla f(\boldsymbol{x}^*)$.

Principle

Choose $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{d}_k$
and choose $d_k = -\nabla f(\boldsymbol{x}_k)$

Convergence

$\lambda_k \to 0$ when $k \to \infty$
$\sum_{k=0}^{\infty} \lambda_k \to +\infty$

Problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

under the constraints :

$$g_i(\boldsymbol{x}) \leq 0 \qquad i = 1..m$$

We define $\mathcal{X}_{sol}$ the solution domain :

$$\mathcal{X}_{sol} = \{\boldsymbol{x} \in \mathcal{X} | g_i(\boldsymbol{x}) \leq 0 \quad \forall i = 1..m\}$$

We assume that $\mathcal{X}_{sol}$ is not empty.

$\boldsymbol{x}^0 \in \mathcal{X}_{sol}$ is a local minima if $f(\boldsymbol{x})$ cannot decrease when $\boldsymbol{x}$ moves on any curve in $\mathcal{X}_{sol}$ that begins in $\boldsymbol{x}^0$.

Definition

This curve is defined by a differentiable function $\varphi(\theta)$ with $\theta \in \mathbb{R}^+$ such that :

- $\varphi(0) = \boldsymbol{x}^0$
- for θ small enough $\varphi(\theta) \in \mathcal{X}_{sol}$

An admissible direction in $\boldsymbol{x}^0$ is any vecteur y tangent to a curve $\varphi(\theta)$.

$$y = \frac{d\varphi}{d\theta}(0)$$

Constraint optimization

Lemme

Let y be an admissible direction, then :

$$\nabla g_i^T(\boldsymbol{x}^0).y \leq 0 \qquad \forall i \in \mathcal{I}^0$$

$\mathcal{I}^0$ is the set of indexes of saturated constraints

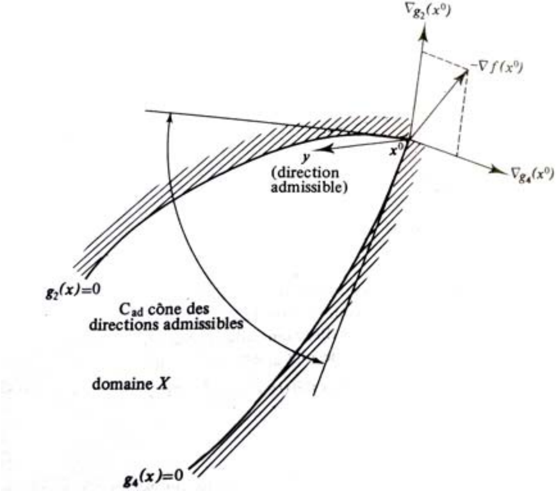
Expliquer

Theorem

A necessary condition for $\boldsymbol{x}^0$ to be a local minima is that there exist λ_i positive reals such that :

$$\left\{\begin{array}{c} \nabla f\left(\boldsymbol{x}^0\right) + \sum_{i=1}^{m} \lambda_i \nabla g_i\left(\boldsymbol{x}^0\right) = 0 \\ \text{and} \\ \lambda_i g_i\left(\boldsymbol{x}^0\right) = 0 \qquad \forall i = 1..m \end{array}\right.$$

Proof :
Consequence of Farkas and minkowski theorem.


$\nabla g_2(x^0)$
$-\nabla f(x^0)$
$\nabla g_4(x^0)$
x^0
y
(direction admissible)
$g_2(x)=0$
C_{ad} cône des directions admissibles
domaine X
$g_4(x)=0$

Saddle point

Definition

Given the following problem :

$$\begin{cases} \min\limits_{\boldsymbol{x}} f(\boldsymbol{x}) \\ g_i(\boldsymbol{x}) \leq 0 \quad i \in \mathcal{I} \\ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d \end{cases}$$

Associate to each constraint a positive real λ_i
The Lagrange function $L(\boldsymbol{x}, \lambda)$ *is define as :*

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \sum_{i \in \mathcal{I}} \lambda_i g_i(\boldsymbol{x})$$

Definition

Let $\boldsymbol{x}^* \in \mathcal{X}$ *and* $\lambda^* \geq 0$.
$(\boldsymbol{x}^*, \lambda^*)$ *is called a saddle point of* $L(\boldsymbol{x}, \lambda)$ *if :*

- $L(\boldsymbol{x}^*, \lambda^*) \leq L(\boldsymbol{x}, \lambda^*)$
- $L(\boldsymbol{x}^*, \lambda) \leq L(\boldsymbol{x}^*, \lambda^*)$

Saddle point

Theorem

Properties of saddle points
Given $\boldsymbol{x} \in \mathcal{X}$ *and* $\lambda^* \geq 0$ *;* $(\boldsymbol{x}, \lambda)$ *is a saddle point of* $L(\boldsymbol{x}, \lambda)$ *if and only if :*

- $L(\boldsymbol{x}^*, \lambda^*) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda^*)$
- $g_i(\boldsymbol{x}^*) \leq 0 \qquad \forall i = 1..m$
- $\lambda_i^* g_i(\boldsymbol{x}^*) = 0 \qquad \forall i = 1..m$

Theorem

Sufficiency of saddle point
If $(\boldsymbol{x}^*, \lambda^*)$ *is a saddle point of* $L(\boldsymbol{x}, \lambda)$ *then* $\boldsymbol{x}^*$ *is the global minima of the constraint optimisation problem*

Proof : $L(\boldsymbol{x}^*, \lambda^*) \leq L(\boldsymbol{x}, \lambda^*)$ and $\lambda_i^* g_i(\boldsymbol{x}^*) = 0$

Properties of saddle points - Proof :

Direct statement

If $(\boldsymbol{x}^*, \lambda^*)$ is a saddle point $L(\boldsymbol{x}^*, \lambda^*) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda^*)$ is true.
By definition of a saddle point, $\forall \lambda \geq 0 : L(\boldsymbol{x}^*, \lambda^*) \geq L(\boldsymbol{x}^*, \lambda)$ thus :

$$f(\boldsymbol{x}^*) + \sum_{i \in \mathcal{I}} \lambda_i^* g_i(\boldsymbol{x}^*) \geq f(\boldsymbol{x}^*) + \sum_{i \in \mathcal{I}} \lambda_i g_i(\boldsymbol{x}^*)$$

then :

$$\forall \lambda \geq 0, \quad \sum_{i \in \mathcal{I}} (\lambda_i - \lambda_i^*) g_i(\boldsymbol{x}^*) \leq 0$$

if $g_i(\boldsymbol{x}^*) > 0 \ldots$
if $\lambda = 0$

Reciprocal

If $L(\boldsymbol{x}^*, \lambda^*) = \min_{\boldsymbol{x}} L(\boldsymbol{x}, \lambda^*)$ then $L(\boldsymbol{x}^*, \lambda^*) \leq L(\boldsymbol{x}, \lambda^*) \quad \forall \boldsymbol{x} \in \mathcal{X}$.
Since $\lambda_i^* g_i(\boldsymbol{x}^*) = 0$ then $L(\boldsymbol{x}^*, \lambda^*) = f(\boldsymbol{x}^*)$ and
$L(\boldsymbol{x}^*, \lambda) = f(\boldsymbol{x}^*) + \sum_{i \in \mathcal{I}} \lambda_i g_i(\boldsymbol{x}^*) \leq f(\boldsymbol{x}^*) = L(\boldsymbol{x}^*, \lambda^*)$ so...

Saddle point and convex problem

Theorem

Given f and g_i convexes functions, $\mathcal{X} \subseteq \mathbb{R}^l$ not empty and that ($\exists \boldsymbol{x}$ such that $g_i(\boldsymbol{x}) < 0 \qquad \forall i = 1..m$) then if the optimisation problem has a solution $\boldsymbol{x}^*$, there exists a vector λ^* such that $(\boldsymbol{x}^*, \lambda^*)$ is a saddle point of $L(\boldsymbol{x}, \lambda)$.

Lagrangian duality

Define $w(\lambda) = \min_{x \in \mathcal{X}} L(\boldsymbol{x}, \lambda)$
The problem :

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

under the constraints :

$$g_i(\boldsymbol{x}) \leq 0 \qquad i = 1..m$$

is tackled by solving :

$$\begin{cases} \max\limits_{\lambda} \; w\left(\lambda\right) = \max\limits_{\lambda} \; \min\limits_{\boldsymbol{x}} \; L\left(\boldsymbol{x}, \lambda\right) \\ \lambda \in \mathbb{R}^{m+} \end{cases}$$

This is the dual problem.

Property

If $w(\lambda^*)$ is the optimal value of the dual problem, $\forall \lambda \in \mathbb{R}^{m+}$ *:*

$$w(\lambda) \leq w(\lambda^*) \leq f(\boldsymbol{x}^*).$$

Saddle point and convex problem

Property

If the optimisation problem admits a saddle point $(\boldsymbol{x}^*, \lambda^*)$ *then .*

$$w(\lambda^*) = f(\boldsymbol{x}^*).$$

If $(\boldsymbol{x}^*, \lambda^*)$ is a saddle point :

$$\begin{aligned} L(\boldsymbol{x}^*, \lambda^*) &= f(\boldsymbol{x}^*) + \lambda^* g(\boldsymbol{x}^*) = f(\boldsymbol{x}^*) \\ &= \min_{\boldsymbol{x} \in \mathcal{X}} L(\boldsymbol{x}, \lambda^*) = w(\lambda^*) \end{aligned}$$

Back to Support vector machines

Minimizing a convex function $f(\boldsymbol{x})$ under the constraints $g_i(\boldsymbol{x}) \leq 0$, $i = 1, \ldots, n$, is equivalent to the search of the saddle point of the Lagrangian :

$$L(\boldsymbol{x}; \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_{i=1}^{n} \alpha_i \, g_i(\boldsymbol{x}).$$

The minimum is searched over $\boldsymbol{x}$. The maximum is over the n Lagrangian multipliers α_i, which must be positive or zero.

$$\max_{\alpha} \min_{x} L\left(\boldsymbol{x}; \boldsymbol{\alpha}\right)$$

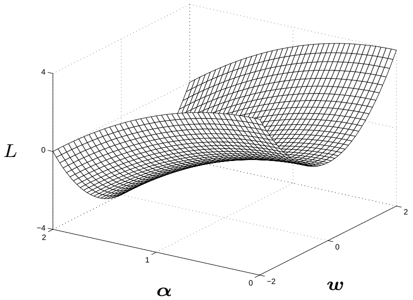
Conditions, called Karush-Kuhn-Tucker conditions, are satisfied at the optimum :

$$\alpha_i^* \, g_i(\boldsymbol{x}^*) = 0, \qquad i = 1, \ldots, n.$$

We solve the above problem using the method of Lagrangian

$$L(\boldsymbol{w}, b; \boldsymbol{\alpha}) = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1\}, \quad \alpha_i \geq 0.$$

The function L must be minimized with respect to primal variables $\boldsymbol{w}$ et b and maximized with respect to dual variables α_i.



Optimality conditions made with respect to the Lagrangian

$$L(\boldsymbol{w}, b; \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1\}$$

result in nul derivatives with respect to the primal and dual variables :

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b; \boldsymbol{\alpha}) = 0 \qquad \frac{\partial}{\partial b} L(\boldsymbol{w}, b; \boldsymbol{\alpha}) = 0.$$

A quick calculation leads to the following relations, injected into the expression of the Lagrangian, provide the dual problem to solve :

$$\sum_{i=1}^{n} \alpha_i^* \, y_i = 0 \qquad \boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* \, y_i \, \boldsymbol{x}_i.$$

The dual optimization problem is finally expressed as :

Maximize $W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$

under the contraints $\sum_{i=1}^{n} \alpha_i \, y_i = 0, \quad \alpha_i \geq 0, \qquad \forall i = 1, \ldots, n.$

The normal vector to the hyperplane optimum separator is expressed as :

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* \, y_i \, \boldsymbol{x}_i$$

According to the Karush-Kuhn-Tucker conditions, the following relation is satisfied at the optimum :

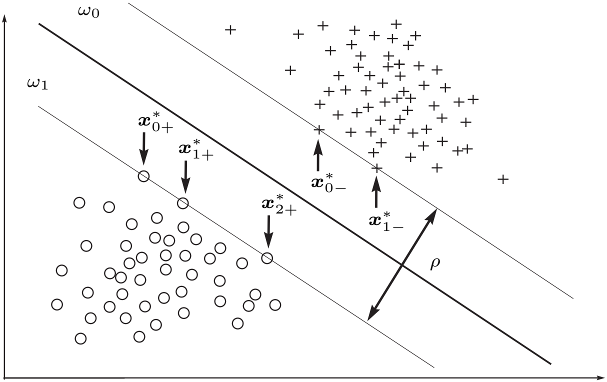
$$\alpha_i^* \{ y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*) - 1 \} = 0, \forall i.$$

Case 1 *:* $y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*) > 1$
We have $\alpha_i^* = 0$*, meaning that* $\boldsymbol{x}_i$ *does not appear in the expression of*$\boldsymbol{w}^*$*.*

Case 2 *:* $y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*) = 1$
We have $\alpha_i^* \neq 0$ *and* $\boldsymbol{x}_i$ *is on the margin. The value of* b^* *is deduced from such samples.*

The vector $\boldsymbol{w}^*$ is defined using only the samples $\boldsymbol{x}_i$ located on the margin : these samples are called *Support Vectors*.

The support vectors are shown below by arrows.



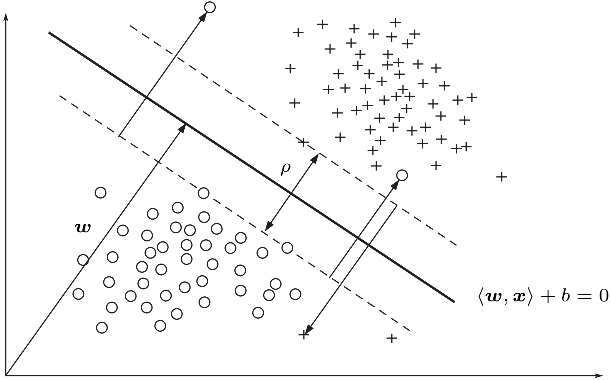
Performances en generalization of SVM

The fact that the optimal hyperplane is expressed only using support vectors is notable because, in general, their number is small.

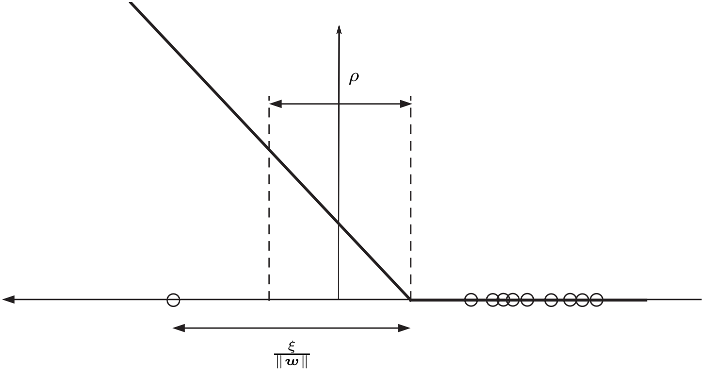
The number n_{sv} of support vectors enables to estimate the generalization performance of the classifier :

$$\mathrm{E}\{P_e\} \leq \frac{\mathrm{E}\{n_{sv}\}}{n}$$

When classes in competition are not linearly separable, the problem formulation has to be modified to penalize misclassified data.



The most common way to penalize the errors is to relate the cost to the distance from the sample to the wrong-sized margin. Sometimes the square of the latter is considered.



The previous scheme leads to formulate the optimization problem as follows :

Minimize $\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \quad C \geq 0$

under the contraints $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \qquad 1 \leq i \leq n.$

The term $C\sum_{i=1}^{n}\xi_i$ penalizes misclassified samples.
Other penalty functions exist.

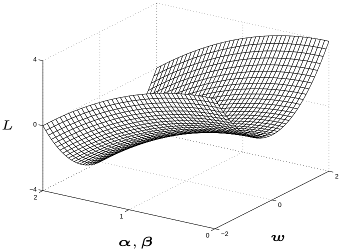
Support vector machines (Soft margin)

Resolution by Lagrange's method

We solve the above problem using the Lagrangian method

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1 + \xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i,$$

where α_i and β_i are nonnegative Lagrangian multipliers.
The function L must be minimized with respect to the primal variables $\boldsymbol{w}$ and b and maximized with respect to dual variables α_i et β_i.



Optimality conditions with respect to the Lagrangian :

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1 + \xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i,$$

result in zero derivatives with respect to primal and dual variables :

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0 \quad &\Longrightarrow \quad \boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* \, y_i \, \boldsymbol{x}_i \\
\frac{\partial}{\partial b} L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0 \quad &\Longrightarrow \quad \sum_{i=1}^{n} \alpha_i^* \, y_i = 0 \\
\frac{\partial}{\partial \boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0 \quad &\Longrightarrow \quad \beta_i^* = C - \alpha_i^*
\end{aligned}$$

Injected into the expression of the Lagrangian, after simplification these relationships provide the dual problem to solve.

Support vector machines (soft margin)

Dual problem formulation and solution

Finally the dual optimization problem can be written as follow :

Maximize $W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$

under the contraints $\sum_{i=1}^{n} \alpha_i \, y_i = 0, \quad 0 \leq \alpha_i \leq C, \qquad \forall i = 1, \ldots, n.$

The solution is finally written :

$$d(\boldsymbol{x}; \boldsymbol{\alpha}^*, b^*) = \text{sign} \left(\sum_{sv} \alpha_i^* \, y_i \, \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b^* \right)$$

In order to determine b^*, the conditions of Karush-Kuhn-Tucker are used :

$$\alpha_i^* \{ y_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b^*) - 1 + \xi_i^* \} = 0, \qquad \beta_i^* \, \xi_i^* = 0.$$

For any vector $\boldsymbol{x}_i$ such that $0 < \alpha_i < C$, we have $\xi_i = 0$ and thus $b^* = y_i - \langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle$.

Minimize $\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \quad C \geq 0$

under the contraints $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \qquad 1 \leq i \leq n.$

The parameter C enable to tune the tradeoff between the width of the margin, which has a regulating role, and the number of misclassified samples.

C ***Large*** *: small margin, less training errors*

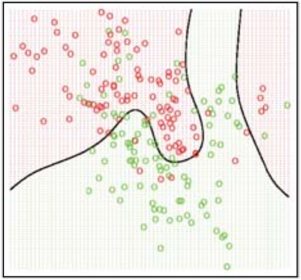
C ***Small*** *: large margin, more training errors*

The choice of the parameter C can be optimized by cross-validation.

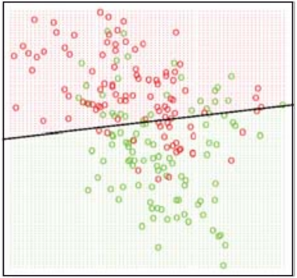
Implementation Examples

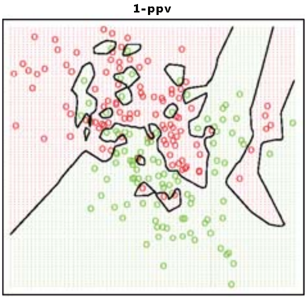
Bayes' detector and Fisher discriminant

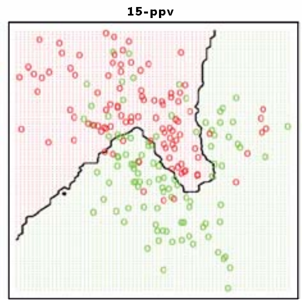
Bayes Classifier



Fisher discriminant



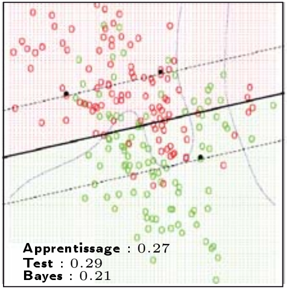

1-ppv


15-ppv

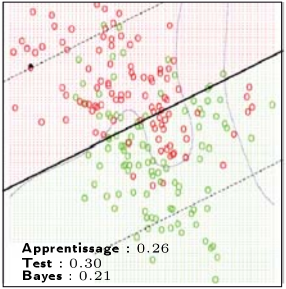
Implementation Examples

Support vector machines

$C = 10^4$



$C = 10^{-2}$



Linear classifiers have limited classification capabilities. To remedy this, one can implement them after non-linear transformation of the data :

$$\boldsymbol{x} \longrightarrow \boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots]^t$$

where $\varphi_i(\boldsymbol{x})$ are the chosen non-linear functions.

A linear classifier with respect to $\phi(x)$ is non-linear with respect to x

Let $\boldsymbol{x} = [x(1)\; x(2)\; x(3)]^t \in \mathbb{R}^3$. Consider the following transformation :

$$\begin{array}{lll} \phi_1(\boldsymbol{x}) = x(1) & \phi_4(\boldsymbol{x}) = x(1)^2 & \phi_7(\boldsymbol{x}) = x(1)\,x(2) \\ \phi_2(\boldsymbol{x}) = x(2) & \phi_5(\boldsymbol{x}) = x(2)^2 & \phi_8(\boldsymbol{x}) = x(1)\,x(3) \\ \phi_3(\boldsymbol{x}) = x(3) & \phi_6(\boldsymbol{x}) = x(3)^2 & \phi_9(\boldsymbol{x}) = x(2)\,x(3) \end{array}$$

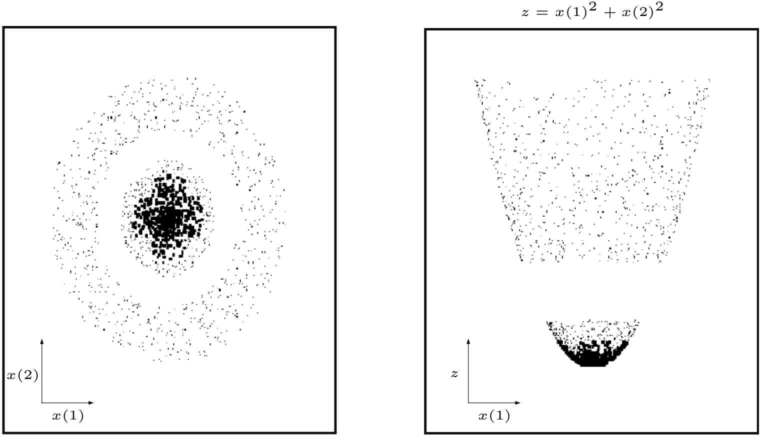
A linear classifier in the transformed space $\{\boldsymbol{\phi}(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathbb{R}^3}$, namely :

$$d(\boldsymbol{x}; \boldsymbol{w}, b) = \mathsf{sign}(\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x})\rangle + b),$$

is a polynomial classifier of order 2 with respect to $\boldsymbol{x}$.

Support vector machines

Example : polynomial classifier



The polynomial transformation makes data linearly separable !

The dual optimization problem is expressed as :

Maximize $W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle$

under the constraints $\sum_{i=1}^{n} \alpha_i \, y_i = 0, \quad 0 \leq \alpha_i \leq C, \qquad \forall i = 1, \ldots, n.$

The solution can be written :

$$d(\boldsymbol{x}; \boldsymbol{\alpha}^*, b^*) = \mathsf{sign} \left(\sum_{sv} \alpha_i^* \, y_i \, \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}_i) \rangle + b^* \right).$$

Note that :

- we never need to explicitly calculate $\boldsymbol{\phi}(\boldsymbol{x})$;
- the dimension of $\boldsymbol{x}$ can be large, the dimension of $\boldsymbol{\phi}(\boldsymbol{x})$ is even larger, sometimes infinite.

If it is possible to define a kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle$ such that :

- the associated decision surface is performant

$$d(\boldsymbol{x}; \boldsymbol{\alpha}^*, b^*) = \text{sign}\left(\sum_{sv} \alpha_i^* \, y_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + b^* \right)$$

- it is easy to calculate $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, even for high-dimensional data...

That's it ! Voila !

In the case of second order polynomial transformation, it is easily shown that :

$$\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle = (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^2 \triangleq \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

▷ **The dot product computation can be performed in $\mathbb{R}^2$!**

More generally, we are interested in : $\kappa(\boldsymbol{x}, \boldsymbol{x'}) = (1 + \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x'}) \rangle)^q$, with $\boldsymbol{x} \in \mathbb{R}^l$.

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^q = \sum_{j=0}^{q} \binom{q}{j} \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^j .$$

Each component $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^j = [x(1)\, x'(1) + \ldots + x(l)\, x'(l)]^j$ of this expression can be developed into a weighted sum of monomials of degree j of the form :

$$[x(1)\, x'(1)]^{j_1} \, [x(2)\, x'(2)]^{j_2} \; \ldots \; [x(l)\, x'(l)]^{j_l}$$

with $\sum_{i=1}^{l} j_i = j$. This directly leads to the expression of $\boldsymbol{\phi}(\boldsymbol{x})$...

We consider the functions $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ that can act as a dot product in a space $\mathcal{H}$. We call *kernel* a symmetric function κ of $\mathcal{X} \times \mathcal{X}$ in $\mathbb{R}$

Theorem (Mercer)

If κ is a continuous positive defined kernel based on an integral operator, which means that :

$$\iint \varphi(\boldsymbol{x})\kappa(\boldsymbol{x}, \boldsymbol{x}')\varphi^*(\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}' \geq 0$$

For any $\varphi \in \mathcal{L}^2(\mathcal{X})$, it can be decomposed as :

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{\infty} \lambda_i \psi_i(\boldsymbol{x})\psi_i(\boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle,$$

where ψ_i and λ_i are the eigenfunctions (orthogonales) and eigenvalues (positives) of the kernel κ, respectively, such that :

$$\int \kappa(\boldsymbol{x}, \boldsymbol{x}')\psi_i(\boldsymbol{x})d\boldsymbol{x} = \lambda_i \psi_i(\boldsymbol{x}')$$

It is easy to see that a kernel κ satisfying Mercer's theorem can act as a scalar product in a transformed space $\mathcal{H}$.
Since :

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} \sqrt{\lambda_1}\,\psi_1(\boldsymbol{x}) \\ \sqrt{\lambda_2}\,\psi_2(\boldsymbol{x}) \\ \cdots \end{pmatrix}$$

Under these conditions, it is verified that :

$$\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}') \rangle = \kappa(\boldsymbol{x}, \boldsymbol{x}')$$

So, let define the space $\mathcal{H}$ as the space generated by the eigenfunctions ψ_i of kernel κ which means that :

$$\mathcal{H} = \{f(\cdot) \mid f(x) = \sum_{i=1}^{\infty} \alpha_i\ \psi_i(x),\ \alpha_i \in \mathbb{R}\}.$$

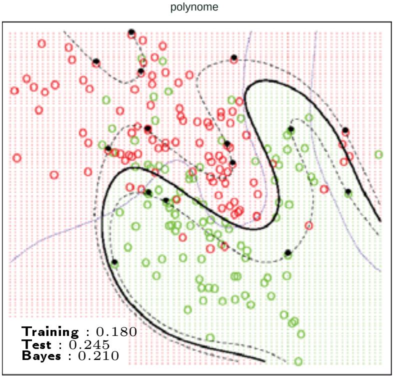
The kernel trick
Examples of Mercer kernel

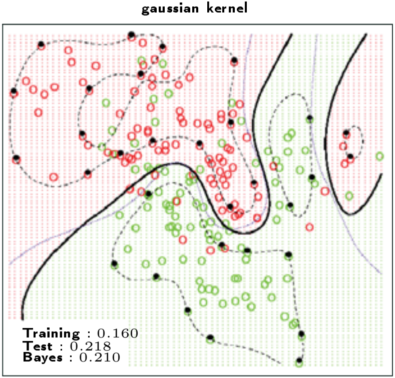
It can be shown that the following kernels verify the condition of Mercer, and thus correspond to a dot product in a space $\mathcal{H}$.

Projective kernels	
monomial of degree q	$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^q$
polynomial of degree q	$(1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^q$
sigmoidal	$\frac{1}{\eta_0} \tanh(\beta_0 \langle \boldsymbol{x}, \boldsymbol{x}' \rangle - \alpha_0)$

Radial kernels	
Gaussien	$\exp(-\frac{1}{2\sigma_0^2} \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert^2)$
exponential	$\exp(-\frac{1}{2\sigma_0^2} \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert)$
uniform	$\frac{1}{\eta_0} \mathbf{1}_{\lVert \boldsymbol{x} - \boldsymbol{x}' \rVert \leq \beta_0}$
Epanechnikov	$\frac{1}{\eta_0} (\beta_0^2 - \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert^2) \mathbf{1}_{\lVert \boldsymbol{x} - \boldsymbol{x}' \rVert \leq \beta_0}$
Cauchy	$\frac{1}{\eta_0} \frac{1}{1 + \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert^2 / \beta_0^2}$

. . . and also : $\kappa_1(\boldsymbol{x}, \boldsymbol{x}') + \kappa_2(\boldsymbol{x}, \boldsymbol{x}')$, $\kappa_1(\boldsymbol{x}, \boldsymbol{x}') \cdot \kappa_2(\boldsymbol{x}, \boldsymbol{x}')$, . . .


polynome
Training : 0.180
Test : 0.245
Bayes : 0.210


gaussian kernel
Training : 0.160
Test : 0.218
Bayes : 0.210

Among the possibilities offered by the kernel trick, we note in particular :

- Any algorithm that is based on scalar products can benefit from the kernel trick for an extension to the non-linear case.
- With the help of the kernel trick, any pattern recognition algorithm is able to process data other than numbers, such as alphabetical or text.

Compared to competing techniques such as artificial neural networks, SVM possess immense qualities :

1. Unique solution
   $\longrightarrow$ Quadratic problem and convex domaine

2. Integrated regularization process, sparse solution
   $\longrightarrow$ Cost function and induced inequality contraints

3. Easy extension to non-linear case, no black box solution
   $\longrightarrow$ Kernel trick

The implementation of the learning algorithm by a direct approach is difficult because the size of the problem is that of the learning set $\mathcal{A}_n$.

Minimize $\quad W(\boldsymbol{\alpha}) = \frac{1}{2}\,\boldsymbol{\alpha}^t \boldsymbol{H}\boldsymbol{\alpha} + \mathbf{1}_n^t\boldsymbol{\alpha}$

under the constraints $\quad \boldsymbol{y}^t\boldsymbol{\alpha} = 0, \quad 0 \cdot \mathbf{1}_n \leq \boldsymbol{\alpha} \leq C \cdot \mathbf{1}_n.$

Efficient algorithms exist, based on the decomposition of the optimization problem into sub-problems.

Minimize

$$W(\boldsymbol{\alpha}_B) = \frac{1}{2}\,(\boldsymbol{\alpha}_B \;\; \boldsymbol{\alpha}_N)^t \begin{pmatrix} \boldsymbol{H}_{BB} & \boldsymbol{H}_{BN} \\ \boldsymbol{H}_{NB} & \boldsymbol{H}_{NN} \end{pmatrix} (\boldsymbol{\alpha}_B \;\; \boldsymbol{\alpha}_N) - (\mathbf{1}_B \;\; \mathbf{1}_N) \begin{pmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N \end{pmatrix}$$

under the constraints $\boldsymbol{\alpha}_B^t\, \boldsymbol{y}_B + \boldsymbol{\alpha}_N^t\, \boldsymbol{y}_N = 0 \quad 0 \cdot \mathbf{1}_B \leq \boldsymbol{\alpha}_B \leq C \cdot \mathbf{1}_B.$

Several strategies have been proposed to decompose the problem. The most extreme is to limit B to two elements. The resolution is then analytic.

Formulation of the problem

Set $\gamma_i = y_i \alpha_i$
Solve :

$$\max_{\gamma} \; w(\gamma) = -\tfrac{1}{2} \sum_{i,j=1}^{N} \gamma_i \gamma_j K(x_i, x_j) + \sum_{i=1}^{N} \gamma_i y_i$$

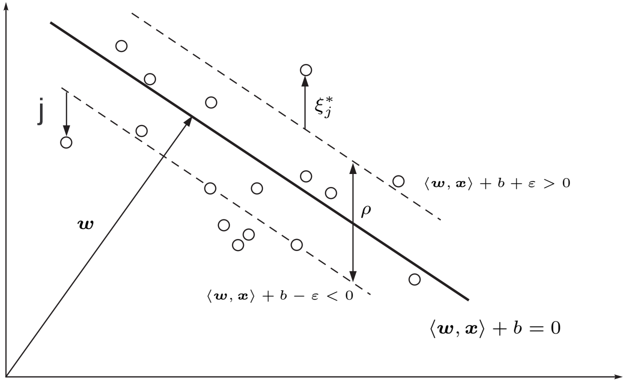
Subject to

$$\begin{cases} \sum\limits_{i=1}^{N} \gamma_i = 0 \\ 0 \leq \gamma_i \leq C \quad \text{for} \quad y_i = 1 \\ -C \leq \gamma_i \leq 0 \quad \text{for} \quad y_i = -1 \end{cases}$$

Find 2 components (i, j) of γ :

$$\gamma_i \leftarrow \gamma_i + \lambda$$
$$\gamma_j \leftarrow \gamma_j - \lambda$$

The idea of margin, on which almost all the qualities of SVM are based, can be applied to regression problem solving.


j
ξ_j^*
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b + \varepsilon > 0$
ρ
$\boldsymbol{w}$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b - \varepsilon < 0$
$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$

The corresponding optimization problem is expressed as follows :

Minimize $\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$

under the constraints $\quad y_i - (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq \varepsilon + \xi_i, \quad \xi_i \geq 0$

$$(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - y_i \geq \varepsilon + \xi_i^*, \quad \xi_i^* \geq 0$$

The kernel trick is applied in the same way as for the SVM, as the resolution algorithms.

We solve the above problem using the Lagrangian method. This leads to the following dual problem :

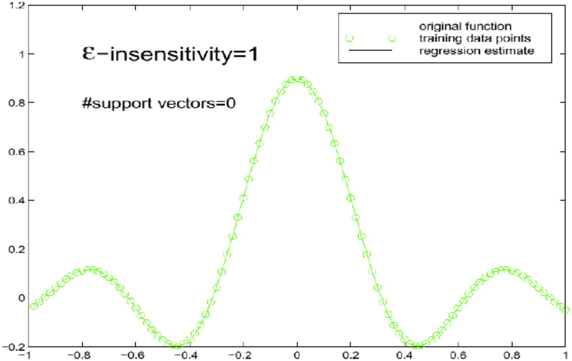
$$\textit{Minimize } W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) + \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$$
$$+ \varepsilon \sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i\,(\alpha_i - \alpha_i^*)$$

under the constraints

$$\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \alpha_i^*, \quad 0 \leq \alpha_i, \alpha_i^* \leq C, \quad \forall i = 1, \ldots, n.$$

The solution is finally written in the following form, enabling the implementation of the kernel trick :

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}(\alpha_i^* - \alpha_i)\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle.$$


ε−insensitivity=1
#support vectors=0
original function
training data points
regression estimate
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1