# Data Science for Actuaries (ACT6100)

Arthur Charpentier

Supervisé # 2 (régularisation - 3)

automne 2020
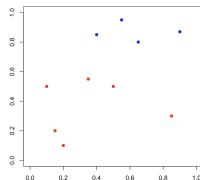
⬥ https://github.com/freakonometrics/ACT6100/

# SVM : Support Vector Machine

> **Linearly Separable sample [regression notations]**
>
> Data $(y_1, \boldsymbol{x}_1), \cdots, (y_n, \boldsymbol{x}_n)$ - with $y \in \{0, 1\}$ - are linearly separable if there are $(\beta_0, \boldsymbol{\beta})$ such that
> - $y_i = 1$ if $\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} > 0$
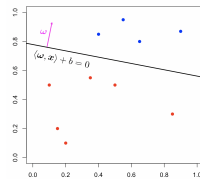> - $y_i = 0$ if $\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} < 0$



> **Linearly Separable sample [ML notations]**
>
> Data $(y_1, \boldsymbol{x}_1), \cdots, (y_n, \boldsymbol{x}_n)$ - with $y \in \{-1, +1\}$ - are linearly separable if there are $(b, \boldsymbol{\omega})$ such that
> - $y_i = +1$ if $b + \langle \boldsymbol{x}_i, \boldsymbol{\omega} \rangle > 0$
> - $y_i = -1$ if $b + \langle \boldsymbol{x}_i, \boldsymbol{\omega} \rangle < 0$
> or equivalently $y_i \cdot (b + \langle \boldsymbol{x}_i, \boldsymbol{\omega} \rangle) > 0, \ \forall i$.

# SVM : Support Vector Machine

$$(b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle) = b + \boldsymbol{x}^\top \boldsymbol{\omega} = 0$$

is an hyperplane (in $\mathbb{R}^p$) orthogonal with $\boldsymbol{\omega}$

Use $m(\boldsymbol{x}) = \mathbf{1}_{b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle \geq 0} - \mathbf{1}_{b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle < 0}$ as classifier
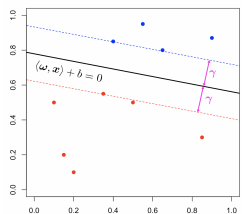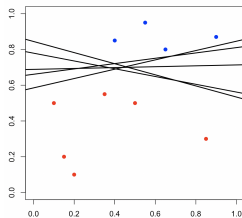
Problem : equation (i.e. $(b, \boldsymbol{\omega})$) is not unique !

Canonical form : $\min_{i=1,\cdots,n} \left\{ |b + \langle \boldsymbol{x}_i, \boldsymbol{\omega} \rangle| \right\} = 1$

Problem : solution here is not unique !

Idea : use the widest (safety) margin $\gamma$

Vapnik & Lerner (1963) and or Cover (1965).



The distance from point $\boldsymbol{x}_i$ to $\Delta$ is $d(\boldsymbol{x}_i, \Delta) = \dfrac{\omega^\top \boldsymbol{x}_i + b}{\|\omega\|}$. Consider

$$\max_{\omega, b} \left\{ \min_{i=1,\cdots,n} \left\{ d(\boldsymbol{x}_i, \Delta) \right\} \right\}$$

# SVM : Support Vector Machine

Consider two points, $\boldsymbol{x}_{-1}$ and $\boldsymbol{x}_{+1}$

$$\gamma = \frac{1}{2} \frac{\langle \omega, \boldsymbol{x}_{+1} - \boldsymbol{x}_{-1} \rangle}{\|\omega\|}$$



It is minimal when
$b + \langle \boldsymbol{\omega}_i, \boldsymbol{x}_{-1} \rangle = -1$ and
$b + \langle \boldsymbol{\omega}_i, \boldsymbol{x}_{+1} \rangle = +1$, and therefore

$$\gamma^{\star} = \frac{1}{\|\boldsymbol{\omega}\|}$$

Optimization problem $\max\{\gamma\}$ becomes

$$\min_{(b\,\boldsymbol{\omega})} \left\{ \frac{1}{2} \|\boldsymbol{\omega}\|_{\ell_2}^2 \right\} \text{ s.t. } y_i \cdot (b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle) > 0, \ \forall i.$$



convex optimization problem with linear constraints
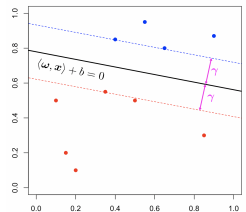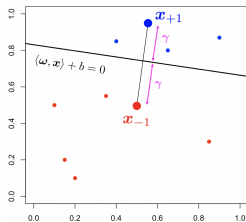
# SVM : Support Vector Machine

Here, $L(b, \boldsymbol{\omega}, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \sum_{i=1}^{n} \alpha_i \cdot \left(y_i \cdot (b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle) - 1\right)$

From the first order conditions,

$$\frac{\partial L(b, \boldsymbol{\omega}, \boldsymbol{\alpha})}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} - \sum_{i=1}^{n} \alpha_i \cdot y_i \boldsymbol{x}_i = \boldsymbol{0}, \text{ i.e. } \boldsymbol{\omega}^\star = \sum_{i=1}^{n} \alpha_i^\star y_i \boldsymbol{x}_i$$

$$\frac{\partial L(b, \boldsymbol{\omega}, \boldsymbol{\alpha})}{\partial b} = -\sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \text{ i.e. } \sum_{i=1}^{n} \alpha_i^\star \cdot y_i = 0$$

and

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

i.e. $\Lambda(\boldsymbol{\alpha}) = \boldsymbol{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{Q} \boldsymbol{\alpha}$

# SVM : Support Vector Machine

The dual problem is

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \right\} \text{ s.t. } \left\{ \begin{array}{l} \alpha_i \geq 0, \ \forall i \\ \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \end{array} \right.$$

where $\boldsymbol{Q} = [\boldsymbol{Q}_{i,j}]$ and $\boldsymbol{Q}_{i,j} = y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, and then

$$\boldsymbol{\omega}^\star = \sum_{i=1}^n \alpha_i^\star y_i \boldsymbol{x}_i \text{ and } b^\star = -\frac{1}{2} \left[ \min_{i:y_i=+1} \{ \langle \boldsymbol{x}_i, \boldsymbol{\omega}^\star \rangle \} + \min_{i:y_i=-1} \{ \langle \boldsymbol{x}_i, \boldsymbol{\omega}^\star \rangle \} \right]$$

Points $\boldsymbol{x}_i$ such that $\alpha_i^\star > 0$ are called support

$$y_i \cdot \left( b^\star + \langle \boldsymbol{x}_i, \boldsymbol{\omega}^\star \rangle \right) = 1$$

Classifier $m^\star(\boldsymbol{x}) = \mathbf{1}_{b^\star + \langle \boldsymbol{x}, \boldsymbol{\omega}^\star \rangle \geq 0} - \mathbf{1}_{b^\star + \langle \boldsymbol{x}, \boldsymbol{\omega}^\star \rangle < 0}$

Observe that $\gamma^\star = \left( \sum_{i=1}^n \alpha_i^{\star 2} \right)^{-1/2}$
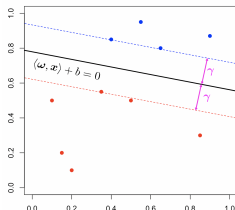
# SVM : Support Vector Machine

Consider here the more general case where the space is not linearly separable

$$(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b)y_i \geq 1$$

becomes

$$(\langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b)y_i \geq 1 - \xi_i$$



for some slack variables $\xi_i$'s.
and penalize large slack variables $\xi_i$ (when $> 0$) by solving (for some cost $C$)

$$\min_{\boldsymbol{\omega}, b} \left\{ \frac{1}{2} \boldsymbol{\omega}^\top \boldsymbol{\omega} + C \sum_{i=1}^{n} \xi_i \right\}$$

subject to $\forall i$, $\xi_i \geq 0$ and $(\boldsymbol{x}_i^\top \boldsymbol{\omega} + b)y_i \geq 1 - \xi_i$.
This is the soft-margin extension, see
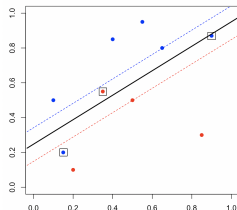
```
1 > e1071::svm()
2 > kernlab::ksvm()
```

# SVM : Support Vector Machine

The dual optimization problem is now

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{1}^\top \boldsymbol{\alpha} \right\} \text{ s.t. } \begin{cases} 0 \leq \alpha_i \leq C, \ \forall i \\ \boldsymbol{y}^\top \mathbf{1} = 0 \end{cases}$$

where $\boldsymbol{Q} = [\boldsymbol{Q}_{i,j}]$ and $\boldsymbol{Q}_{i,j} = y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, and then

$$\boldsymbol{\omega}^\star = \sum_{i=1}^n \alpha_i^\star y_i \boldsymbol{x}_i \text{ and } b^\star = -\frac{1}{2} \left[ \min_{i:y_i=+1} \{\langle \boldsymbol{x}_i, \boldsymbol{\omega}^\star \rangle\} + \min_{i:y_i=-1} \{\langle \boldsymbol{x}_i, \boldsymbol{\omega}^\star \rangle\} \right]$$

Note further that the (primal) optimization problem can be written

$$\min_{(b,\boldsymbol{\omega})} \left\{ \frac{1}{2} \|\omega\|_{\ell_2}^2 + \sum_{i=1}^n \left( 1 - y_i \cdot (b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle) \right)_+ \right\},$$

where $(1-z)_+$ is a convex upper bound for empirical error $\mathbf{1}_{z \leq 0}$

# SVM : Support Vector Machine, with R

The dual optimization problem is now

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{Q} \boldsymbol{\alpha} - \boldsymbol{1}^\top \boldsymbol{\alpha} \right\} \text{ s.t. } \left\{ \begin{array}{l} 0 \leq \alpha_i \leq C, \ \forall i \\ \boldsymbol{y}^\top \boldsymbol{\alpha} = 0 \end{array} \right.$$

where $\boldsymbol{Q} = [\boldsymbol{Q}_{i,j}]$ and $\boldsymbol{Q}_{i,j} = y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$

```
1 > library ( quadprog )
2 > C = .5
3 > y = ( myocarde [ ," PRONO "]==" SURVIE ") *2 -1
4 > X = as . matrix ( cbind (1 , myocarde [ ,1:7]) )
5 > n = length ( y )
6 > Q = sapply (1: n , function ( i ) y [ i ]* t ( X ) [ , i ])
7 > D = t ( Q ) %*% Q
8 > d = matrix (1 , nrow = n )
9 > A = rbind ( y , diag ( n ) , - diag ( n ) )
10 > b = c (0 , rep (0 , n ) , rep ( -C , n ) )
```
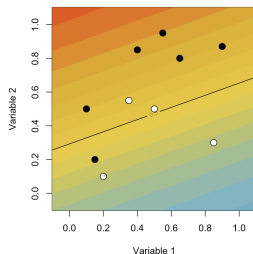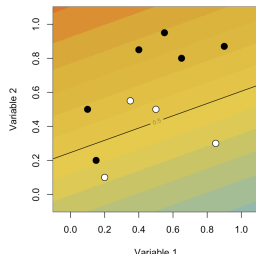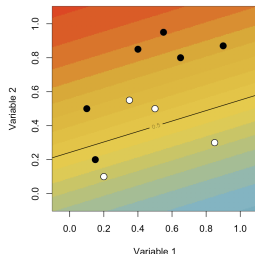
# SVM : Support Vector Machine, with R

```
1 > eps = 5e-4
2 > sol = solve.QP(D+eps*diag(n), d, t(A),b, meq=1,
    factorized=FALSE)
3 > qpsol = sol$solution
4 > omega = apply(qpsol*y*X,2,sum)
5 > omega
6     1   FRCAR   INCAR   INSYS   PRDIA   PAPUL   PVENT   REPUL
7 0.000   0.055  -0.092   0.361  -0.109  -0.049  -0.066   0.001
```

car $\boldsymbol{\omega}^{\star} = \sum_{i=1}^{n} \alpha_i^{\star} y_i \boldsymbol{x}_i$

# SVM : Support Vector Machine, with R

```
1  x1 = c(.4,.55,.65,.9,.1,.35,.5,.15,.2,.85)
2  x2 = c(.85,.95,.8,.87,.5,.55,.5,.2,.1,.3)
3  y = c(1,1,1,1,1,0,0,1,0,0)
4  df = data.frame(x1=x1,x2=x2,y=2*y-1)
5  library(kernlab)
6  SVM2 = ksvm(y ~ x1 + x2, data = df, C=2.5, kernel = "
       vanilladot" , prob.model=TRUE , type="C-svc")
```

# SVM : Support Vector Machine

One can also consider the kernel trick : $\mathbf{x}_i^\top \mathbf{x}_j$ is replace by $\varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)$ for some mapping $\varphi$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)$$

For instance $K(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^\top \mathbf{b})^3 = \varphi(\mathbf{a})^\top \varphi(\mathbf{b})$
where $\varphi(a_1, a_2) = (a_1^3 \ , \ \sqrt{3}a_1^2 a_2 \ , \ \sqrt{3}a_1 a_2^2 \ , \ a_2^3)$
Consider polynomial kernels

$$K(\mathbf{a}, \mathbf{b}) = (1 + \mathbf{a}^\top \mathbf{b})^p$$

or a Gaussian kernel

$$K(\mathbf{a}, \mathbf{b}) = \exp(-(\mathbf{a} - \mathbf{b})^\top (\mathbf{a} - \mathbf{b}))$$

and solve $\displaystyle\max_{\alpha_i \geq 0} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right\}$

# SVM : Support Vector Machine

The radial kernel is formed by taking an infinite sum over polynomial kernels...

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma\|\boldsymbol{x} - \boldsymbol{y}\|^2\right) = \langle\psi(\boldsymbol{x}), \psi(\boldsymbol{y})\rangle$$

where $\psi$ is some $\mathbb{R}^n \to \mathbb{R}^\infty$ function, since

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma\|\boldsymbol{x} - \boldsymbol{y}\|^2\right) = \underbrace{\exp(-\gamma\|\boldsymbol{x}\|^2 - \gamma\|\boldsymbol{y}\|^2)}_{=\text{constant}} \cdot \exp\left(2\gamma\langle\boldsymbol{x}, \boldsymbol{y}\rangle\right)$$

i.e.

$$K(\boldsymbol{x}, \boldsymbol{y}) \propto \exp\left(2\gamma\langle\boldsymbol{x}, \boldsymbol{y}\rangle\right) = \sum_{k=0}^{\infty} 2\gamma\frac{\langle\boldsymbol{x}, \boldsymbol{y}\rangle^k}{k!} = \sum_{k=0}^{\infty} 2\gamma K_k(\boldsymbol{x}, \boldsymbol{y})$$

where $K_k$ is the polynomial kernel of degree $k$.
If $K = K_1 + K_2$ with $\psi_j : \mathbb{R}^n \to \mathbb{R}^{d_j}$ then $\psi : \mathbb{R}^n \to \mathbb{R}^d$ with $d \sim d_1 + d_2$

# SVM : Support Vector Machine

A kernel is a measure of similarity between vectors.
The smaller the value of $\gamma$ the narrower the vectors should be to have a small measure

Is there a probabilistic interpretation ?
Platt (2000, Probabilities for SVM) suggested to use a logistic function over the SVM scores,

$$p(\boldsymbol{x}) = \frac{\exp[b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle]}{1 + \exp[b + \langle \boldsymbol{x}, \boldsymbol{\omega} \rangle]}$$

# SVM : Support Vector Machine, with R

```r
x1 = c(.4,.55,.65,.9,.1,.35,.5,.15,.2,.85)
x2 = c(.85,.95,.8,.87,.5,.55,.5,.2,.1,.3)
y = c(1,1,1,1,1,0,0,1,0,0)
df = data.frame(x1=x1,x2=x2,y=2*y-1)
library(kernlab)
SVM2 = ksvm(y ~ x1 + x2, data = df, C=1, kernel = "
    rbfdot" , prob.model=TRUE, type="C-svc")
```