

# Supervised learning

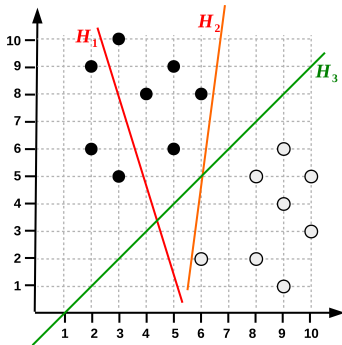
## (VII) Support Vector Machine (SVM)

Séverine Affeldt

Université Paris Cité  
Centre Borelli, UFR Sciences Fondamentales et Biomédicales

2023 – 2024

	x		
	$x_1$	$x_2$	y
i	1	9	-1
ii	2	6	-1
iii	2	8	-1
iv	3	10	-1
v	4	9	-1
vi	5	3	1
vii	5	8	-1
viii	5	10	-1
ix	6	2	1
x	6	5	1
xi	6	9	-1
xii	8	4	1
xiii	8	6	1
xiv	9	2	1
xv	9	5	1
xvii	10	3	1

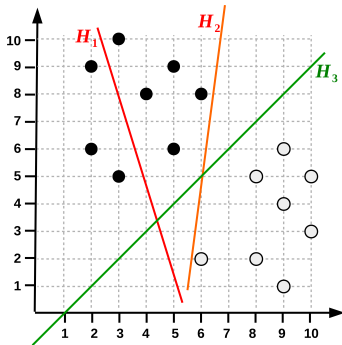


## Objective

Find the decision boundary that will indicate the class of a new *data point*  $x$  ?

- data point  $x$ :  $p$ -dimensional vector (eg.  $x_1 = (x_{1,1}, x_{1,2})$ )
- decision boundary  $H$ :  $(p - 1)$ -dimensional hyperplane (eg.  $H_1, H_2, H_3$ )

	x		
	$x_1$	$x_2$	y
i	1	9	-1
ii	2	6	-1
iii	2	8	-1
iv	3	10	-1
v	4	9	-1
vi	5	3	1
vii	5	8	-1
viii	5	10	-1
ix	6	2	1
x	6	5	1
xi	6	9	-1
xii	8	4	1
xiii	8	6	1
xiv	9	2	1
xv	9	5	1
xvii	10	3	1



## Objective

Find the decision boundary that will indicate the class of a new *data point*  $x$  ?

- data point  $x$ :  $p$ -dimensional vector (eg.  $x_1 = (x_{1,1}, x_{1,2})$ )
- decision boundary  $H$ :  $(p - 1)$ -dimensional hyperplane (eg.  $H_1, H_2, H_3$ )

## SVM

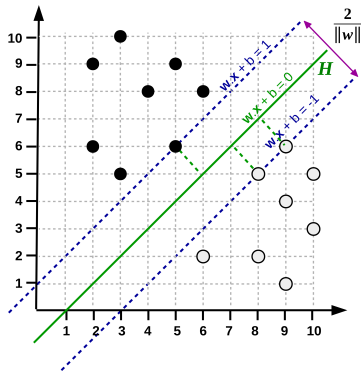
$H_3$  is a reasonable choice: maximises the **margin** between the two classes

(Vapnik & Chervonenkis, 1963)

## Decision boundary

Let us consider a training dataset of the form  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ , where  $\mathbf{x}_i$  is a  $p$  dimensional **real** vector and  $y_i \in \{-1, 1\}$ . We want to find a hyperplane that maximizes the distance to the nearest  $\mathbf{x}_i$ .

Hyperplane  $H$  with parameters  $\{w_i\} \Leftrightarrow \{\mathbf{x}\}$ ,  $\sum_{i=1}^n w_i x_i + b = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0$



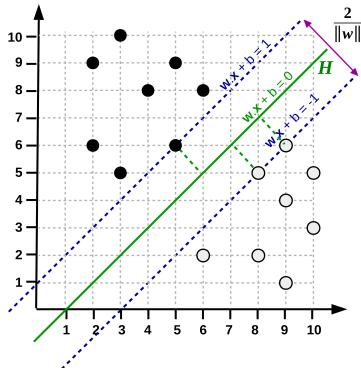
## Linearly separable data

∃ **two parallel** hyperplanes that separate the data, such that the **distance** between them is as **large** as possible. This distance is the **margin**.

$$(1) \langle \mathbf{w}, \mathbf{x} \rangle + b = 1$$

$$(2) \langle \mathbf{w}, \mathbf{x} \rangle + b = -1$$

Where does the **margin** come from?



## Objective

Maximizing the margin  $\Leftrightarrow$  find  $\mathbf{w}$  s.t.  $\max(\frac{2}{\|\mathbf{w}\|}) \Leftrightarrow$  find  $\mathbf{w}$  s.t.  $\min(\|\mathbf{w}\|)$ ,  $\|\mathbf{w}\| = \sqrt{w_1^2 + w_2^2 + \dots}$

## Supplementary constraint

$\forall i,$

$$\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b \geq 1, \text{ if } y_i = 1$$

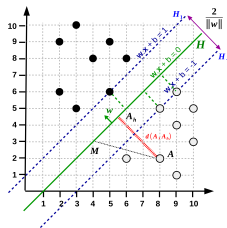
or

$$\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b \leq -1, \text{ if } y_i = -1$$

All in all,

$$\forall 1 \leq i \leq n, y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1$$

*No data point within the margin!*



## Full SVM objective

Minimizing  $\|\mathbf{w}\|$ , where  $\|\mathbf{w}\| = \sqrt{w_1^2 + w_2^2 + \dots}$ , and subject to  $y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1, \forall 1 \leq i \leq n$

Our classifier is defined by  $\mathbf{w}$  and  $b$  that satisfy this requirements. It is **fully** determined by the closest samples  $\mathbf{x}^{s_i}$ , which are named **support vectors**.

## Primal form of the optimization problem

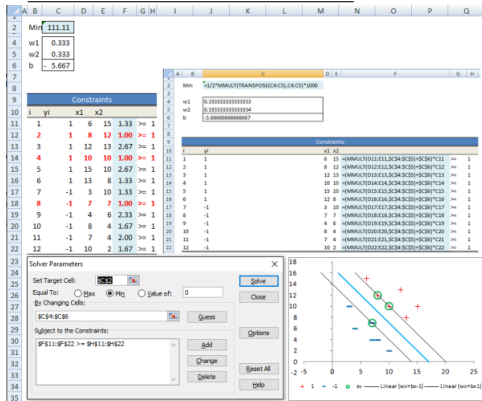
Minimizing  $\frac{1}{2}(w_1^2 + w_2^2 + \dots + w_p^2)$  subject to  $y_i(\langle \mathbf{w} \cdot \mathbf{x}^i \rangle + b) \geq 1, \forall 1 \leq i \leq n$

⇒ a well known quadratic optimization problem, where we have a **quadratic objective function** subject to **linear constraints**.

⇒ not possible for large  $p$  ( $>$  few hundreds)...

⇒ not applicable to non linearly separable data...

### Toy example with the Excel Solver(12 constraints)



## Dual form of the optimization problem

Minimizing  $\frac{1}{2}(w_1^2 + w_2^2 + \dots + w_p^2)$  subject to  $y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1, \forall 1 \leq i \leq n$

Set up a **Lagrange** function and derive the solution **analytically**

---

<sup>1</sup>Karush, Kuhn & Tucker



## Dual form of the optimization problem

Minimizing  $\frac{1}{2}(w_1^2 + w_2^2 + \dots + w_p^2)$  subject to  $y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1, \forall 1 \leq i \leq n$

Set up a **Lagrange** function and derive the solution **analytically**

Rewrite the constraint,

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1 \geq 0$$

Multiple by **Lagrange multipliers** ( $\alpha_i$ ) and subtract from the objective function,

$$\operatorname{argmin}_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w}^T \cdot \mathbf{w} \rangle - \sum_{i=1}^n \alpha_i (y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1), \text{ where } \alpha_i \geq 0$$

NB: minimizing with respect to  $\mathbf{w}$  and  $b$ ...but maximizing with respect to  $\{\alpha_i\}$

---

<sup>1</sup>Karush, Kuhn & Tucker

## Dual form of the optimization problem

Minimizing  $\frac{1}{2}(w_1^2 + w_2^2 + \dots + w_p^2)$  subject to  $y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1, \forall 1 \leq i \leq n$

Set up a **Lagrange** function and derive the solution **analytically**

Rewrite the constraint,

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1 \geq 0$$

Multiple by **Lagrange multipliers** ( $\alpha_i$ ) and subtract from the objective function,

$$\operatorname{argmin}_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w}^T \cdot \mathbf{w} \rangle - \sum_{i=1}^n \alpha_i (y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1), \text{ where } \alpha_i \geq 0$$

NB: minimizing with respect to  $\mathbf{w}$  and  $b$ ...but maximizing with respect to  $\{\alpha_i\}$

A quadratic optimization problem **must** satisfied the KKT<sup>1</sup> conditions,

$$(1) \nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \ \& \ \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$(2) \alpha_i \geq 0$$

$$(3) \alpha_i (y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1) = 0$$

$$(4) y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) - 1 \geq 0$$

Caution:  $\alpha_i > 0 \Rightarrow \mathbf{x}_i$  is a support vector

<sup>1</sup>Karush, Kuhn & Tucker

## Conditions

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

$$(1) \quad \nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \quad \& \quad \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$(2) \quad \alpha_i \geq 0$$

$$(3) \quad \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1) = 0$$

$$(4) \quad y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1 \geq 0$$

## Lagrangian expansion

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

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

## Conditions

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

$$(1) \quad \nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \quad \& \quad \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$(2) \quad \alpha_i \geq 0$$

$$(3) \quad \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1) = 0$$

$$(4) \quad y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1 \geq 0$$

## Getting the dual formulation

$$(1) \quad \nabla_{\mathbf{w}} = \mathbf{w} - \sum_{i=1}^n \alpha_i y_i \mathbf{x}^{S_i} \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}^{S_i}$$

$$(2) \quad \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = - \sum_{i=1}^n \alpha_i y_i \Rightarrow - \sum_{i=1}^n \alpha_i y_i = 0$$

## Lagrangian expansion

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

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

...substituting in expanded Lagrangian...

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{S_i} \mathbf{x}^{S_j} \rangle$$

## Conditions

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

$$(1) \quad \nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \quad \& \quad \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$(2) \quad \alpha_i \geq 0$$

$$(3) \quad \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1) = 0$$

$$(4) \quad y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1 \geq 0$$

## Lagrangian expansion

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

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

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{S_i} \mathbf{x}^{S_j} \rangle, \quad \text{where} \quad \alpha_i \geq 0 \quad \& \quad \sum_{i=1}^n \alpha_i y_i = 0$$

## Conditions

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

$$(1) \quad \nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \quad \& \quad \frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$(2) \quad \alpha_i \geq 0$$

$$(3) \quad \alpha_i (y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1) = 0$$

$$(4) \quad y_i (\langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + b) - 1 \geq 0$$

## Lagrangian expansion

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

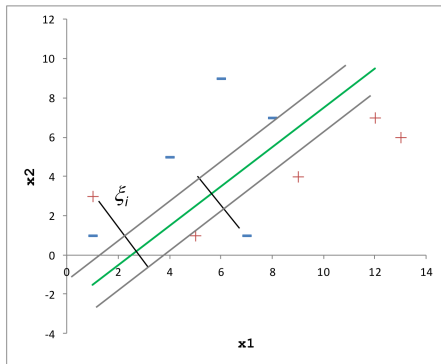
$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle - \sum_{i=1}^n \alpha_i y_i \langle \mathbf{w}, \mathbf{x}^{S_i} \rangle + \sum_{i=1}^n \alpha_i y_i b - \sum_{i=1}^n \alpha_i$$

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{S_i} \mathbf{x}^{S_j} \rangle, \quad \text{where} \quad \alpha_i \geq 0 \quad \& \quad \sum_{i=1}^n \alpha_i y_i = 0$$

NB: Instead of minimizing over  $\mathbf{w}$  and  $b$  subject to linear constraints, we can maximize over  $\alpha$

## In practice, there is no perfect separation!



- $\xi$  est un vecteur de taille  $n$
- $\xi_i \geq 0$  matérialise l'erreur de classement pour chaque observation
- $\xi_i = 0$ , elle est nulle lorsque l'observation est du bon côté de la droite « marge » associée à sa classe
- $\xi_i < 1$ , le point est du bon côté de la frontière, mais déborde de la droite « marge » associée à sa classe
- $\xi_i > 1$ , l'individu est mal classé

In practice, there is no perfect separation!

Formulation  
primale

$$\min_{\beta, \beta_0, \xi_i} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \xi_i$$

s.t.

$$y_i \times (x_i^T \beta + \beta_0) \geq 1 - \xi_i, \forall i = 1, \dots, n$$

$$\xi_i \geq 0, \forall i$$

La tolérance aux erreurs est plus ou moins accentuée avec le paramètre **C** ("cost" parameter)

→ C trop élevé, danger de sur-apprentissage

→ C trop faible, sous-apprentissage

Formulation  
duale

$$\max_{\alpha} L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{i'=1}^n \alpha_i \alpha_{i'} y_i y_{i'} \langle x_i, x_{i'} \rangle$$

s.t.

$$\sum_{i=1}^n \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \forall i$$



In practice, there is no perfect separation!

### 'Soft' Primal formulation

$\operatorname{argmin}_{\mathbf{w}, b, \xi^{s_i}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$ ,  $\xi^{s_i}$  tolerance or classification error,  $C$  cost

with constraints,

$$\forall i, 1 \leq i \leq n, \quad y_i(\langle \mathbf{w} \cdot \mathbf{x}^{s_i} \rangle + b) \geq 1 - \xi^{s_i} \quad \& \quad \xi^{s_i} \geq 0$$

### 'Soft' Primal formulation

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{s_i} \mathbf{x}^{s_j} \rangle$$

with constraints,

$$\forall i, 1 \leq i \leq n, \quad 0 \leq \alpha_i \leq C \quad \& \quad \sum_{i=1}^n \alpha_i y_i = 0$$

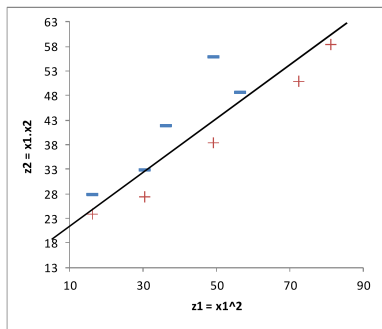
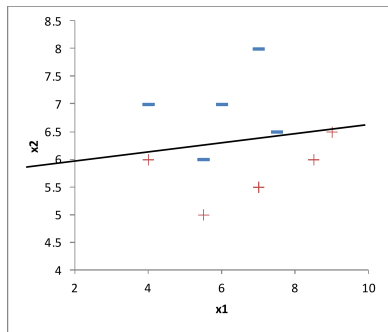
### About the cost $\xi^{s_i}$ and $C$

OK  $\Rightarrow \xi^{s_i} = 0$ ; OK but within margin  $\Rightarrow \xi^{s_i} < 1$ ; not OK  $\Rightarrow \xi^{s_i} > 1$

$C$  too high  $\Rightarrow$  overfitting;  $C$  too low  $\Rightarrow$  underfitting

With *appropriate* variable transformations, we can turn a non linearly separable problem into a linearly separable one!

An example:  $\Phi(\mathbf{x} = (x_1, x_2)) = (x_1^2, x_1x_2)$



## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{s_i} \mathbf{x}^{s_j} \rangle, \quad \text{where } \alpha_i \geq 0 \text{ \& } \sum_{i=1}^n \alpha_i y_i = 0$$

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{s_i} \mathbf{x}^{s_j} \rangle, \quad \text{where } \alpha_i \geq 0 \text{ \& } \sum_{i=1}^n \alpha_i y_i = 0$$

## Directly transforming the variables

An example:  $\Phi(\mathbf{x} = (x_1, x_2)) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

We need to compute all the  $\langle \Phi(\mathbf{x}^{s_i}) \Phi(\mathbf{x}^{s_j}) \rangle$ , and we need to manipulate 3 variables instead of 2... This is time and memory costly...

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{s_i} \mathbf{x}^{s_j} \rangle, \quad \text{where } \alpha_i \geq 0 \text{ \& } \sum_{i=1}^n \alpha_i y_i = 0$$

## Directly transforming the variables

An example:  $\Phi(\mathbf{x} = (x_1, x_2)) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

We need to compute all the  $\langle \Phi(\mathbf{x}^{s_i}) \Phi(\mathbf{x}^{s_j}) \rangle$ , and we need to manipulate 3 variables instead of 2... This is time and memory costly...

## Using a Kernel function

$$K(\mathbf{x}^{s_i}, \mathbf{x}^{s_j}) = \langle \Phi(\mathbf{x}^{s_i}), \Phi(\mathbf{x}^{s_j}) \rangle$$

Here, we compute the scalar product (as before), and convert only the result with K!  
We manipulate only 2 variables, but still, we are in a higher dimension.

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{S_i} \mathbf{x}^{S_j} \rangle, \quad \text{where } \alpha_i \geq 0 \text{ \& } \sum_{i=1}^n \alpha_i y_i = 0$$

Let us consider two data points/vectors,

$$\mathbf{u} = (4, 7) \text{ \& } \mathbf{v} = (2, 5) \Rightarrow \langle \mathbf{u}, \mathbf{v} \rangle = 4 \times 2 + 5 \times 7 = 43$$

## Example (a)

$$\begin{aligned} \Phi(\mathbf{x} = (x_1, x_2)) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \\ \Phi(\mathbf{u} = (16, 39.6, 49)) \text{ \& } \Phi(\mathbf{v} = (4, 14.1, 25)) \\ \Rightarrow \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle &= 1849 \end{aligned}$$

Using the corresponding  $K$ ,  $K_1(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle)^2 = 43^2 = 1849$

## Dual formulation of the optimization problem

$$\operatorname{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{S_i} \mathbf{x}^{S_j} \rangle, \quad \text{where } \alpha_i \geq 0 \text{ \& } \sum_{i=1}^n \alpha_i y_i = 0$$

Let us consider two data points/vectors,

$$\mathbf{u} = (4, 7) \text{ \& } \mathbf{v} = (2, 5) \Rightarrow \langle \mathbf{u}, \mathbf{v} \rangle = 4 \times 2 + 5 \times 7 = 43$$

### Example (a)

$$\begin{aligned} \Phi(\mathbf{x} = (x_1, x_2)) &= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \\ \Phi(\mathbf{u} = (16, 39.6, 49)) \text{ \& } \Phi(\mathbf{v} = (4, 14.1, 25)) \\ \Rightarrow \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle &= 1849 \end{aligned}$$

Using the corresponding  $K$ ,  $K_1(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle)^2 = 43^2 = 1849$

### Example (b)

$$\begin{aligned} \Phi(\mathbf{x} = (x_1, x_2)) &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2, x_2^2) \\ \Phi(\mathbf{u} = (1, 5.7, 9.9, 16, 49, 39; 6)) \text{ \& } \Phi(\mathbf{v} = (1, 2.8, 7.1, 4, 25, 14.1)) \\ \Rightarrow \langle \Phi(\mathbf{u}), \Phi(\mathbf{v}) \rangle &= 1936 \end{aligned}$$

Using the corresponding  $K$ ,  $K_2(\mathbf{u}, \mathbf{v}) = (1 + \langle \mathbf{u}, \mathbf{v} \rangle)^2 = (1 + 43)^2 = 1936$

With a kernel function  $K$ , computations are equivalent, but we project all data points into a higher dimensional space without explicitly transforming the data points.

'Soft' Dual formulation of the optimization problem with a kernel function

$$\text{argmax}_{\alpha} L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}^{S_i} \mathbf{x}^{S_j}), \quad \text{where } 0 \leq \alpha_j \leq C \ \& \ \sum_{i=1}^n \alpha_i y_i = 0$$

Classification with a kernel function

$$f(\mathbf{x}_i) = \sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i \cdot \mathbf{x}) + b$$



## Polynomial Kernel

$$K(\mathbf{u}, \mathbf{v}) = (coef0 + \langle \mathbf{u}, \mathbf{v} \rangle)$$

## Radial basis Kernel

$$K(\mathbf{u}, \mathbf{v}) = \exp(-\gamma \times \|\mathbf{u} - \mathbf{v}\|^2)$$

## Sigmoid Kernel

$$K(\mathbf{u}, \mathbf{v}) = \tanh(-\gamma \times \langle \mathbf{u}, \mathbf{v} \rangle + coef0)$$

NB: More kernel functions in LIBSVM, available in `scikit-learn` or `e1071`

## Bibliography

- Ng, Andrew. "CS 229 Lecture Notes: Support Vector Machines.", cs229. stanford.edu/notes (2012)
- Rakotomalala, Ricco. "Machines à Vecteurs de Support."

## A reminder on normal vector and direction vector

### Normal vector

If  $(d)$  is a line defined by the equation  $ax + by + c = 0$ , then  $v = (a, b)$  is a vector normal to  $(d)$ .

Let's choose two points,  $A = (x_A, y_A)$  and  $M = (x, y)$  that belong to  $(d)$ . If  $v$  is normal to  $(d)$ , then  $\langle AM, v \rangle = \langle (x_A - x, y_A - y), (a, b) \rangle = 0$ .

Hence,  $A$  and  $M$  are such that  $a(x - x_A) + b(y - y_A) = 0 \Leftrightarrow ax + by + c = 0$ , with  $c = -(ax_A + by_A)$

### Direction vector

A direction vector of  $(d)$  is  $u = (-b, a)$ .

## Distance $d(A, A_h)$

$\mathbf{w} = (w_1, w_2)^T$  is a vector **normal** to  $H$ .

$$\forall M \in H, \quad \langle \mathbf{AM}, \mathbf{w} \rangle = \langle (\mathbf{AA}_h + \mathbf{A}_h \mathbf{M}), \mathbf{w} \rangle = \langle \mathbf{AA}_h, \mathbf{w} \rangle$$

$$\langle \mathbf{AA}_h, \mathbf{w} \rangle = w_1(x_{A_h} - x_A) + w_2(y_{A_h} - y_A)$$

$$\langle \mathbf{AA}_h, \mathbf{w} \rangle = w_1 x_{A_h} + w_2 y_{A_h} - w_1 x_A - w_2 y_A$$

$$\langle \mathbf{AA}_h, \mathbf{w} \rangle = -w_1 x_A - w_2 y_A - b \quad (A_h \in H)$$

...and...

$$\langle \mathbf{AA}_h, \mathbf{w} \rangle = \|\mathbf{AA}_h\| \|\mathbf{w}\|$$

$$\Rightarrow d(A, A_h) = \|\mathbf{AA}_h\| = \frac{|w_1 x_A + w_2 y_A + b|}{\|\mathbf{w}\|}$$

$$\text{If } A \in H_1 \text{ or } A \in H_{-1}, \quad \|\mathbf{AA}_h\| = \frac{1}{\|\mathbf{w}\|}$$

...hence...

$$\text{Margin} = \frac{2}{\|\mathbf{w}\|}$$

