

# Support Vector Machines (SVM)

Jérémie Gince

Département de physique  
Université de Sherbrooke

September 12, 2023

# Contents

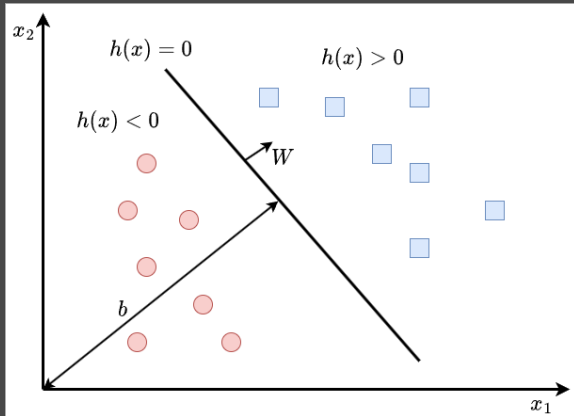
1 Binary classification

2 SVM

3 SVM in Python

4 SVM - The origin

# Context



**Figure:** Two classes being separate by a line with the parameters  $W$  and  $b$  and the estimator  $h(\cdot)$ .

# Hinge Loss

The hinge loss used to maximize the margins of a binary classifier is the following:

$$E_{\text{hinge}}(\theta|\mathcal{D}) = \sum_{\{x^t, y^t\} \in \mathcal{D}} \max(0, 1 - y^t h(x^t|\theta)) + E_{\text{regularization}}(\theta). \quad (1)$$

- if  $y^t = 1$  and  $h^t > 0 \rightarrow$  good classification with  $E \in [0, 1)$
- if  $y^t = 1$  and  $h^t < 0 \rightarrow$  bad classification with  $E > 0$
- if  $y^t = -1$  and  $h^t < 0 \rightarrow$  good classification with  $E \in [0, 1)$
- if  $y^t = -1$  and  $h^t > 0 \rightarrow$  bad classification with  $E > 0$

# Estimator

The linear separator is defined by the decision function

$$h^t(x^t|W, b) = \sum_{j=1}^M W_j x_j^t + b \quad (2)$$

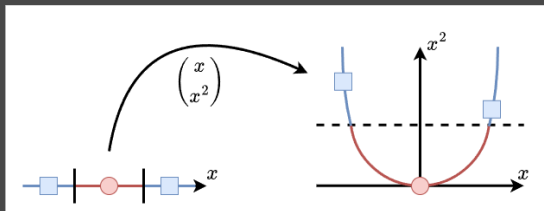
$$\hat{y}^t = \phi(h^t(x^t|W, b)) \quad (3)$$

with a training dataset  $\mathcal{D} = \{x^t, y^t\}$  where

- $x_j^t$  is  $j$ -th entry of the input  $t$  of size  $M$ ;
- $W_j$  is the  $j$ -th weight from the vector  $W$  of size  $M$ ;
- $b$  is the bias as a scalar;
- $\phi(\cdot)$  is the activation function which is often  $\text{sign}(\cdot)$  in a binary classifier;
- $y^t$  is the target class  $t$ ;
- $\hat{y}^t$  is the predicted class  $t$ .

# Changing the decision space

It's possible that the data  $\mathcal{D}$  are not linearly separable in the initial space  $\mathcal{S}_0$  but they are in another space  $\mathcal{S}_1$ . Then we can use a transformation  $\psi : \mathbb{R}^M \mapsto \mathbb{R}^N$ .



**Figure:** Two classes not separable in initial space but are separable after the transformation  $\psi(x) = (x \ x^2)^T$ .

- The linear separator will be

$$h(x) = W^T \psi(x) + b.$$

# SVM

In a SVM, the parameters  $W$  of the separator are

$$W = \sum_t \alpha^t y^t \psi(x^t) \quad (5)$$

with  $\alpha^t \geq 0$ , a parameter associated with  $x^t$ . This lead to

$$h(x) = \sum_t \alpha^t y^t (\psi(x^t))^T \psi(x) + b. \quad (6)$$

# The Kernel

The kernel function is defined as

$$K(x, y) = (\psi(x))^T \psi(y). \quad (7)$$

Using the kernel definition, the SVM decision function is

$$h(x) = \sum_t \alpha^t y^t K(x^t, x) + b. \quad (8)$$

The computing of  $\psi(\cdot)$  is now hidden in the new kernel function, so this  $\psi(\cdot)$  can be "unknown" from the SVM decision function.



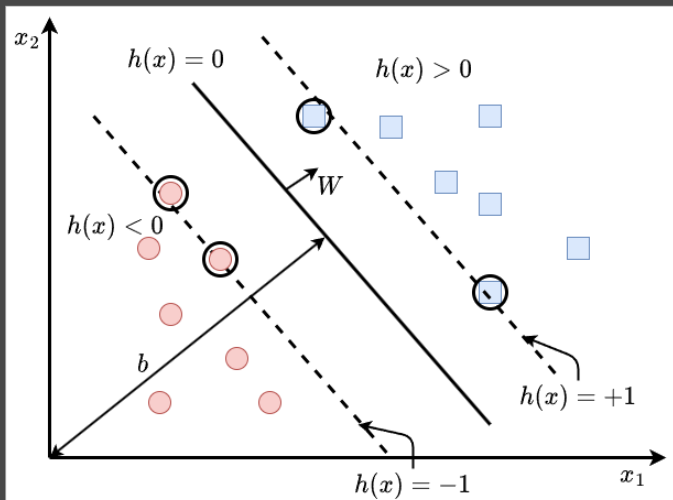
Considering that some  $\alpha^t$  are null, we can write the prediction function as

$$h(x) = \sum_s \alpha^s y^s K(x^s, x) + b, \quad (9)$$

$$\hat{y} = \phi(h(x)) \quad (10)$$

where

- the index  $s$  are associated with the non-zero  $\alpha^t$ . The  $x^s$  are what we called, the support vectors.



**Figure:** Binary classification. The dots lines represent the margins of the classifier and the circled data are the support vectors.

# Finding the parameters

To Learn the parameters  $\alpha^t$  and  $b$  of the SVM classifier, we can do a gradient descent in  $L$  iterations. The updates are

$$\blacksquare \Delta\alpha^t = -\eta \frac{\partial E(\alpha, b | \mathcal{D})}{\partial \alpha^t},$$

$$\blacksquare \Delta b = -\eta \frac{\partial E(\alpha, b | \mathcal{D})}{\partial b}$$

with the user-defined error function  $E(\cdot)$  and the learning rate  $\eta$ . The update of the parameters will be

$$\alpha_\kappa = \begin{cases} 0 & \text{if } \alpha_{\kappa-1} + \Delta\alpha^t < 0 \\ \alpha_{\kappa-1} + \Delta\alpha^t & \text{else,} \end{cases} \quad (11)$$

$$b_\kappa = b_{\kappa-1} + \Delta b, \quad \forall \kappa \in \{1, \dots, L\}. \quad (12)$$

# Hinge Loss

The following hinge loss is generally used for SVM:

$$E_{\text{hinge}}(\alpha, b|\mathcal{D}) = \sum_{\{x^t, y^t\} \in \mathcal{D}} \max(0, 1 - y^t h(x^t|\alpha, b)) + \lambda \frac{1}{2} \sum_{\alpha^s \in \alpha} (\alpha^s)^2, \quad (13)$$

where  $\lambda$  is the regularization coefficient and generally  $\lambda \in [0, 1]$ .

# Advantages/Disadvantages

## Advantages:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: user-defined Kernel for the decision function.

## Disadvantages:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates.

Advantages and disadvantages taken from [scikit-learn](#).

# Code

- Repository: [https://github.com/JeremieGince/Learning\\_SVM](https://github.com/JeremieGince/Learning_SVM)
- Google Colab: [Learning\\_SVM.google.colab](https://colab.research.google.com/github/JeremieGince/Learning_SVM/blob/main/01_SVM.ipynb)

# Support vector machines (SVM)

SVM is a special type of linear separator with following goals:

- Maximize the distance between the separating hyperplane and the training data:

$$d_{h^t, x^t} = \frac{|W^T x^t + b|}{\|W\|} = \frac{y^t (W^T x^t + b)}{\|W\|} \quad (14)$$

- Find the weights  $W$  and  $b$  in order to have

$$d_{h^t, x^t} \geq \rho, \forall t \quad (15)$$

where  $\rho$  is the margin.

# Lagrange multipliers

In order to achieve those goals, we have to use the Lagrange multipliers.

- It's a method used to optimize a function  $f(x)$  under a constraint  $g(x)$  written in a way that  $g(x) = 0$ ,
- with the Lagrangian

$$\mathcal{L}(x, \alpha) = f(x) + \alpha g(x). \quad (16)$$

- The stationary points of the Lagrangian and the optimal solution of  $f(x)$  is then found with

$$\nabla \mathcal{L}(x, \alpha) = 0. \quad (17)$$



# Lagrangian multipliers - Example

Suppose we want to maximize the function  $f(q_i) = q_1 + q_2$  with the constraint  $q_1^2 + q_2^2 = 1$ . The constraint will then be noted as

$$g(q_i) = q_1^2 + q_2^2 - 1 = 0, \quad (18)$$

and the Lagrangian as

$$\mathcal{L}(q_i, \alpha) = f(q_i) + \alpha g(q_i) \quad (19)$$

$$\implies \mathcal{L}(q_i, \alpha) = q_1 + q_2 + \alpha(q_1^2 + q_2^2 - 1). \quad (20)$$

The gradient of the Lagrangian will be

$$\nabla \mathcal{L}(q_i, \alpha) = \left[ \frac{\partial \mathcal{L}(q_i)}{\partial q_1}, \quad \frac{\partial \mathcal{L}(q_i)}{\partial q_2}, \quad \frac{\partial \mathcal{L}(q_i)}{\partial \alpha} \right], \quad (21)$$

$$= [1 + 2\alpha q_1, \quad 1 + 2\alpha q_2, \quad q_1^2 + q_2^2 - 1], \quad (22)$$

$$= \begin{bmatrix} 1 + 2\alpha q_1 = 0, \\ 1 + 2\alpha q_2 = 0, \\ q_1^2 + q_2^2 - 1 = 0 \end{bmatrix}, \quad (23)$$

$$q_1 = q_2 = -\frac{1}{2\alpha}, \quad (24)$$

$$\Rightarrow \alpha^{(1)} = +\frac{1}{\sqrt{2}}, \quad \alpha^{(2)} = -\frac{1}{\sqrt{2}}. \quad (25)$$

Since  $f(-\frac{1}{2\alpha^{(1)}}, -\frac{1}{2\alpha^{(1)}}) = -\sqrt{2}$  and  $f(-\frac{1}{2\alpha^{(2)}}, -\frac{1}{2\alpha^{(2)}}) = \sqrt{2}$ , the Lagrange multiplier that maximize our function  $f(q_i)$  is  $\alpha^{(2)}$ .

# Lagrange multipliers with inequality

If the constraint is  $g(x) \geq 0$ , the conditions for optimality will be

- $g(x) \geq 0$ ;
- $\alpha \geq 0$ ;
- $\alpha g(x) = 0$ .

Then to minimize  $f(x)$  with  $g(x) \geq 0$ , we have to optimize the Lagrangian

$$\mathcal{L}(x, \alpha) = f(x) - \alpha g(x) \quad (26)$$

with  $\alpha \geq 0$ .

# SVM

The problem:

- Minimize  $\frac{1}{2} \|W\|^2$
- with the constraint  $y^t (W^T x^t + b) \geq 1 \ \forall t$

Re-writing using the Lagrangian multiplier:

$$\mathcal{L}_p = \frac{1}{2} \|W\|^2 - \sum_t \alpha^t [y^t (W^T x^t + b) - 1] \quad (27)$$

$$= \frac{1}{2} \|W\|^2 - \sum_t \alpha^t y^t (W^T x^t + b) + \sum_t \alpha^t \quad (28)$$

with  $\mathcal{L}_p$ , the primal form of the Lagrangian.

# Simplification - dual form

Since we want to find an stationary point of  $\mathcal{L}_p$ :

$$\frac{\partial \mathcal{L}_p}{\partial W} = 0 = W - \sum_t \alpha^t y^t x^t, \quad (29)$$

$$\implies W = \sum_t \alpha^t y^t x^t, \quad (30)$$

and

$$\frac{\partial \mathcal{L}_p}{\partial b} = 0 = \sum_t \alpha^t y^t. \quad (31)$$

The dual form can be written as

$$\mathcal{L}_d = \frac{1}{2}(W^T W) - \underbrace{W^T \sum_t \alpha^t y^t x^t}_W - \underbrace{b \sum_t \alpha^t y^t}_0 + \sum_t \alpha^t \quad (32)$$

$$= -\frac{1}{2}(W^T W) + \sum_t \alpha^t \quad (33)$$

$$\implies \mathcal{L}_d = -\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s y^t y^s (x^t)^T x^s + \sum_t \alpha^t \quad (34)$$

# Dual form

Re-writing the problem using the dual form.

- Maximize  $-\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s y^t y^s (x^t)^T x^s + \sum_t \alpha^t$
- With the constraints  $\sum_t \alpha^t y^t = 0$  and  $\alpha^t \geq 0 \forall t$
- In practice, a majority of  $\alpha^t$  will be zero;
- The data  $x^t$  with  $\alpha^t > 0$  are called the support vectors.

# Changing the decision space

It's possible that the data  $\mathcal{D}$  are not linearly separable in the initial space  $\mathcal{S}_0$  but are in another space  $\mathcal{S}_1$ . Then we can use a transformation  $\psi : \mathbb{R}^M \mapsto \mathbb{R}^N$  that  $x \mapsto \psi(x)$ .

- The linear separator will be

$$h(x) = W^T \psi(x) + b \quad (35)$$

- In the dual form:

$$W = \sum_t \alpha^t y^t \psi(x^t) \quad (36)$$

$$h(x) = \sum_t W^T \psi(x) + b = \sum_t \alpha^t y^t (\psi(x^t))^T \psi(x) + b \quad (37)$$



# The Kernel

The kernel function is defined as

$$K(x, y) = (\psi(x))^T \psi(y). \quad (38)$$

Using the kernel definition, the SVM decision function is

$$h(x) = \sum_t \alpha^t y^t K(x^t, x) + b. \quad (39)$$

The computing of  $\psi(\cdot)$  is now hidden in the new kernel function, so this  $\psi(\cdot)$  can be "unknown" from the SVM decision function.

The resulting prediction function of the SVM:

$$h(x) = \sum_{x^s} \alpha^s y^s K(x^s, x) + b, \quad (40)$$

$$\hat{y} = \phi(h(x)). \quad (41)$$

- Reminder:  $x^s$  is the support vector  $s$  associated with the non-zero Lagrangian multiplier  $\alpha^s$ .

# The End