



Machine Learning Models





Machine Learning Models

- In the design of any supervised machine learning model, the model design can be organized into three well-defined parts.
 - **Model:** Refers to the equations or structure of the mathematical model, which requires obtaining parameters to replicate the behavior of the data.
 - **Objective function or cost function:** This refers to the equation that is required to minimize or maximize, which represents the objective that is required to be achieved. Defining this function can totally change the way your model works.
 - **Training algorithm:** This refers to the method used to solve the optimization problem described by the cost function.
 - **Evaluation metrics:** This refers to the method used to evaluate the trained model performance in different scenarios.





Support Vector Machines

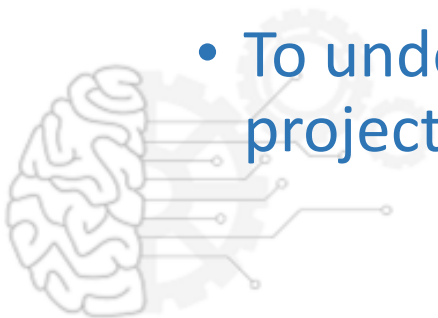
Dr. Riemann Ruiz Cruz





Support Vector Machines

- The term SVM is typically used in both cases for regression and classification problems. Typically SVC to refer classification with support vector methods, and SVR to refer a support vector regression.
- Support vectors were originally proposed to solve a classification problem based on maximizing the existing margin at a decision frontier.
- To understand the margin one must remember the theory of vector projection.





Support Vector Machines

- Scalar product

- The dot product or scalar product is an algebraic operation that takes two equal-length sequences of numbers (usually coordinate vectors), and returns a single number.

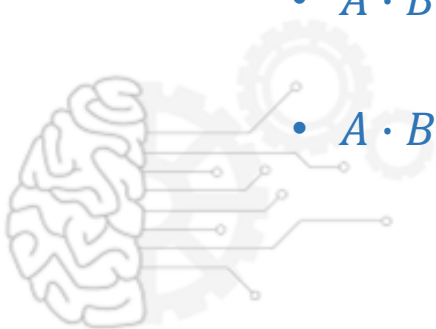
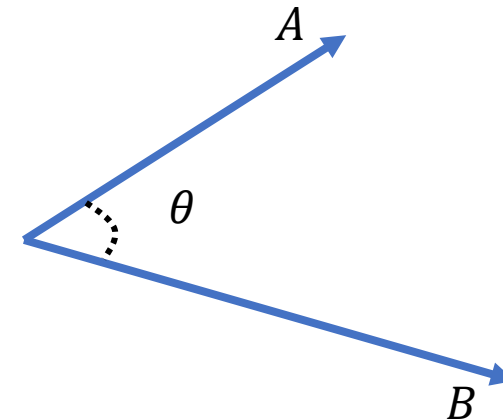
- $A = [a_1 \ \cdots \ a_n]^T; B = [b_1 \ \cdots \ b_n]^T$

- $A \cdot B = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n;$

- $A \cdot B = A^T B$

- $A \cdot B = \|A\| \|B\| \cos(\theta)$

- $A \cdot B = \langle A, B \rangle$

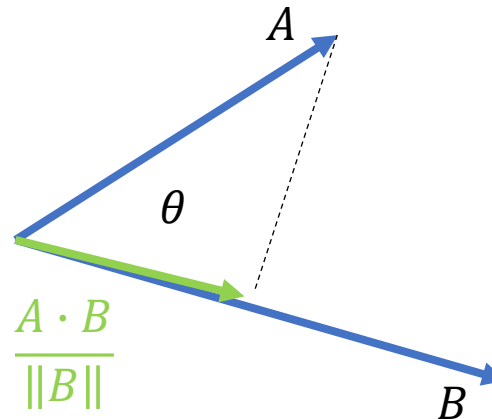
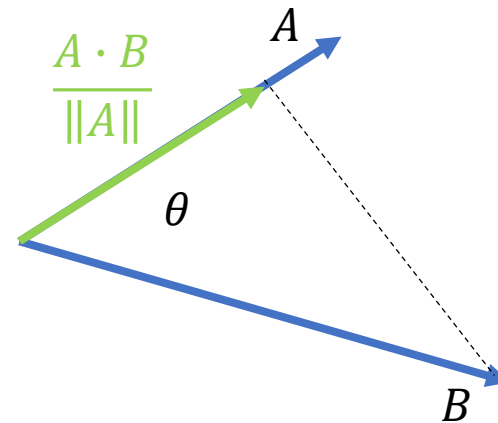




Support Vector Machines

- Scalar product
 - $A \cdot B = \|A\| \|B\| \cos(\theta)$

- $\frac{A \cdot B}{\|A\|} = \|B\| \cos(\theta)$





- $\frac{A \cdot B}{\|B\|} = \|A\| \cos(\theta)$



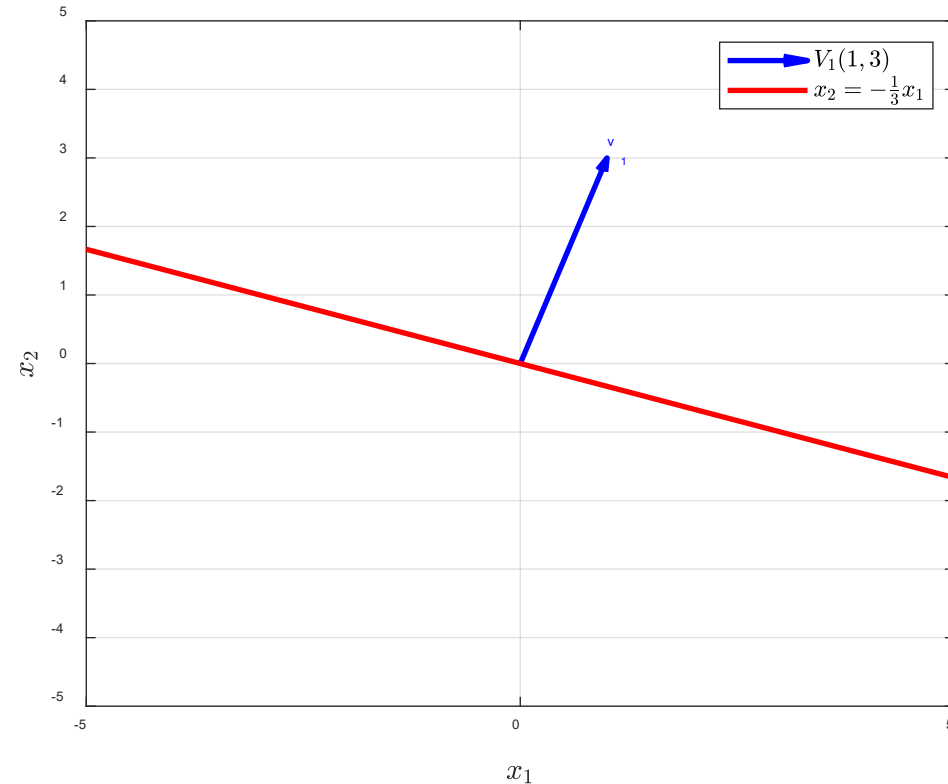
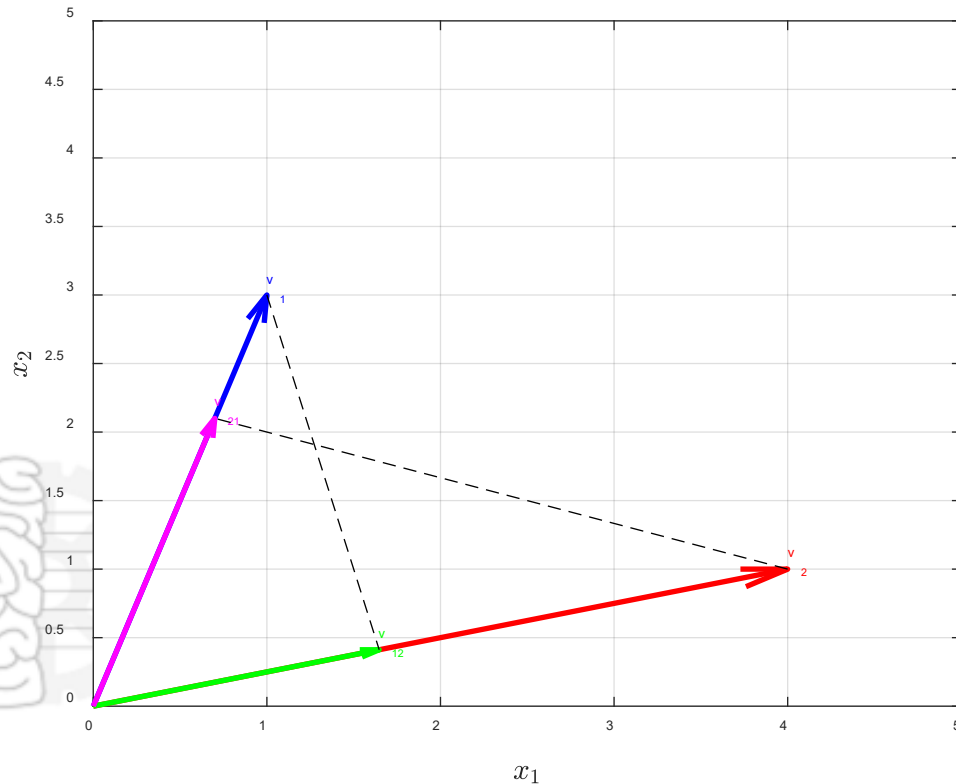
Support Vector Machines

$$\underline{y} = w_1 x_1 + w_2 x_2 = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1$$

Vector: $V_1(\overset{w_1}{\underbrace{v_1(x_1)}}, \overset{w_2}{\underbrace{v_1(x_2)}})$

Plano: $x_2 = -\frac{v_1(x_1)}{v_1(x_2)} x_1$





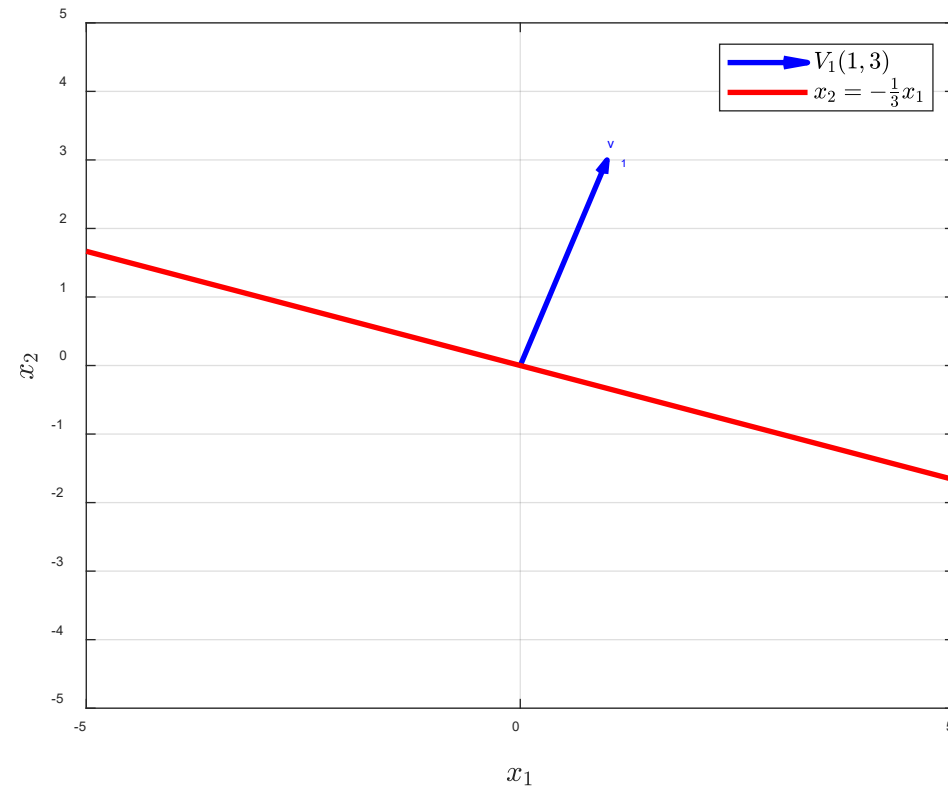
Support Vector Machines

modelo $y = w_0 + w_1x_1 + w_2x_2$

vector W $W(w_1, w_2)$

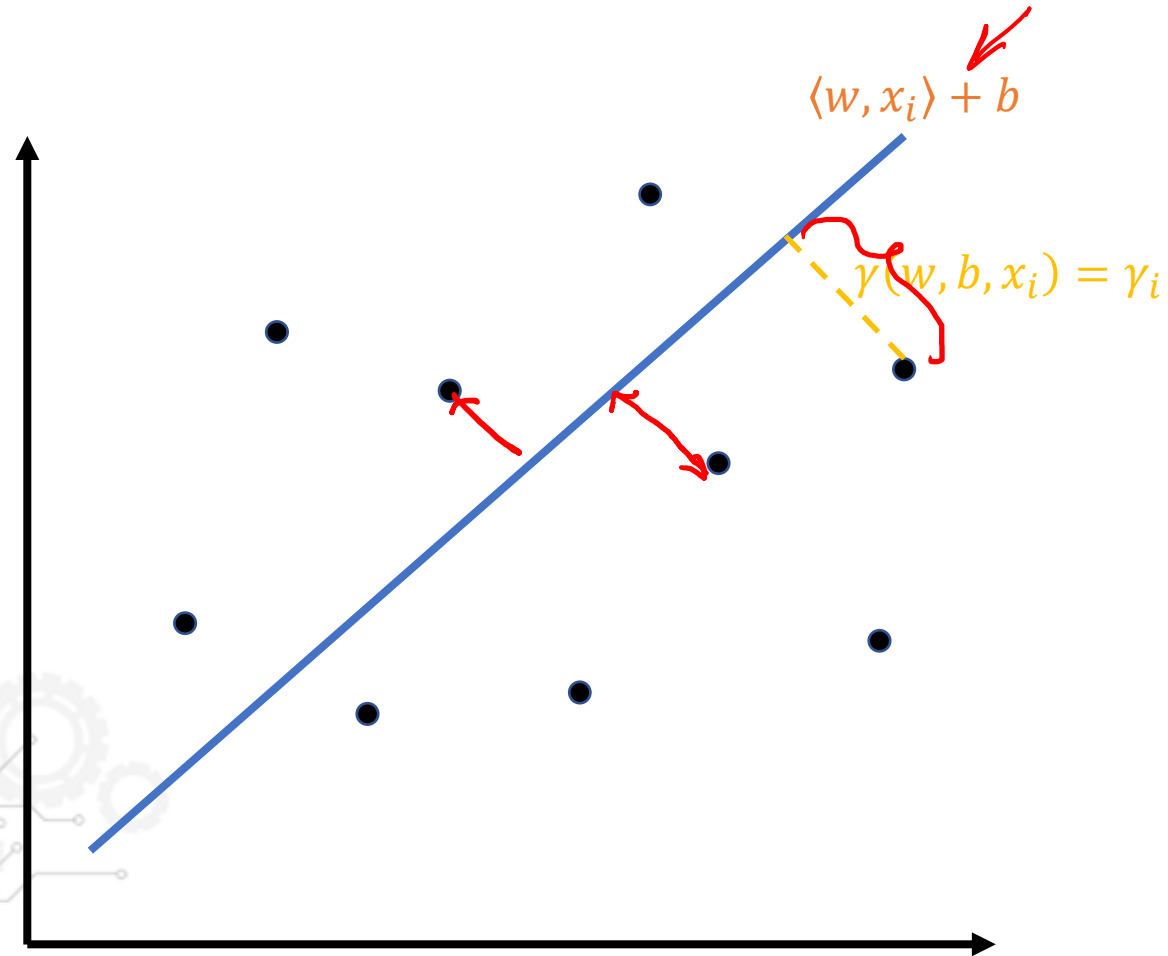
Intersección
con plano x_1, x_2

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{w_0}{w_2}$$





Support Vector Machines



$$\gamma_i = \frac{|\langle w, x_i \rangle + b|}{\|w\|}$$

$$\gamma_i = \left| \frac{w \cdot x_i}{\|w\|} + \frac{b}{\|w\|} \right|$$

margin $\rho(\underline{w}, \underline{b}) = \min_{x_i} \gamma(w, b, x_i)$



Support Vector Machines

- The margin is the distance from the points x_i closest to the surface $w \cdot x_i + b$.
- The optimal hyperplane is the one that maximizes the margin for all x_i , and this can be found by optimizing the w and b parameters.

- Optimal hyperplane: $\max_{w,b} \rho(w, b) = \max_{w,b} \min_{x_i} \left| \frac{w \cdot x_i}{\|w\|} + \frac{b}{\|w\|} \right|$

- So to maximize the margin, it is equivalent to minimize only the denominator w , because the margin increases if the denominator is small.

- $\max_{w,b} \rho(w, b) = \min_w \|w\|$

- To reduce the complexity of the problem, the square of the modulus of w can be optimized.

- $\min_w \frac{1}{2} \|w\|^2$





Support Vector Regression (SVR)





Regression problem

- This problem requires finding a function that maps two data sets.

$$Y = f(X)$$

- X , are vectors with the input data, where each column is the predictor p .

$$X = [x_1, x_2, \dots, x_p], x_p \in R_e$$

- Y , are vectors with the output data, where each column is a desired output.

$$Y = [y_1, y_2, \dots, y_s], y_s \in R_e$$





Support Vector Regression

- In general, the solution to this problem of maximum margin is the trivial solution. But it is important to consider the conditions that warrant the regression, which we have not mentioned yet.
- In a regression problem, the estimation error is sought to be within an acceptable error interval.
- $f(x) = \langle w, x \rangle + b$
- $|\hat{y}_i - y_i| = |\langle w, x_i \rangle + b - y_i| \leq \varepsilon$

$$|\langle w, x_i \rangle + b - y_i| \leq \varepsilon \longrightarrow \begin{cases} \langle w, x_i \rangle + b - y_i \leq \varepsilon \\ y_i - \langle w, x_i \rangle - b \leq \varepsilon \end{cases}$$





Support Vector Regression

- So to solve a regression problem using the support vector scheme it can be written as an optimization problem with constraints.

- Optimization objective: $\min_w \frac{1}{2} \|w\|^2$

- Constraints: $\begin{cases} \langle w, x_i \rangle + b - y_i - \varepsilon \leq 0 \\ y_i - \langle w, x_i \rangle - b - \varepsilon \leq 0 \end{cases}$

$$\textcircled{1} \quad \hat{y} = w_0 + w_1 x + w_2 x^2 + \dots + w_n x^n$$

$$\textcircled{2} \quad J = \frac{1}{n} \sum (\hat{y} - y)^2 + \lambda \|w\|^2$$

$J(w, b)$

$d'?$

- So the optimization problem using Lagrange multipliers can be written as:

- $\min_{w, b, \alpha_i, \alpha_i^*} L(w, b, \alpha_i, \alpha_i^*);$

- $L(w, b, \alpha_i, \alpha_i^*) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \underline{\alpha_i} (\langle w, x_i \rangle + b - y_i - \varepsilon) + \sum_{i=1}^m \underline{\alpha_i^*} (y_i - \langle w, x_i \rangle - b - \varepsilon)$



Support Vector Regression

- To solve the optimization problem, the gradients are calculated for each variable to be optimized.

- $L(\underline{w}, b, \underline{\alpha_i}, \underline{\alpha_i^*}) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (\langle \underline{w}, \underline{x_i} \rangle + b - y_i - \varepsilon) + \sum_{i=1}^m \alpha_i^* (\underline{y_i} - \langle \underline{w}, \underline{x_i} \rangle - b - \varepsilon)$

- $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial w} = w + \sum_{i=1}^m \alpha_i x_i + \sum_{i=1}^m -\alpha_i^* x_i = w + \sum_{i=1}^m (\alpha_i - \alpha_i^*) x_i = \emptyset$

- $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial b} = \sum_{i=1}^m \alpha_i + \sum_{i=1}^m -\alpha_i^* = \sum_{i=1}^m (\alpha_i - \alpha_i^*) = \emptyset$

- $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial \alpha_i} = \langle w, x_i \rangle + b - y_i - \varepsilon = \emptyset$

- $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial \alpha_i^*} = y_i - \langle w, x_i \rangle - b - \varepsilon = \emptyset$



Support Vector Regression

- $L(w, b, \alpha_i, \alpha_i^*) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (\langle w, x_i \rangle + b - y_i - \varepsilon) + \sum_{i=1}^m \alpha_i^* (y_i - \langle w, x_i \rangle - b - \varepsilon)$
- $L(w, b, \alpha_i, \alpha_i^*) = \frac{1}{2} w^T w + \sum_{i=1}^m \alpha_i (\langle w, x_i \rangle) + \sum_{i=1}^m \alpha_i (b - y_i - \varepsilon) + \sum_{i=1}^m \alpha_i^* (-\langle w, x_i \rangle) + \sum_{i=1}^m \alpha_i^* (y_i - b - \varepsilon)$
- $L(w, b, \alpha_i, \alpha_i^*) = \frac{1}{2} w^T w + \sum_{i=1}^m \alpha_i (\langle w, x_i \rangle) - \sum_{i=1}^m \alpha_i^* (\langle w, x_i \rangle) + \sum_{i=1}^m (\alpha_i - \alpha_i^*) b + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i + \sum_{i=1}^m (\alpha_i^* + \alpha_i) \varepsilon$





Support Vector Regression

- From $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial b} = \sum_{i=1}^m (\alpha_i - \alpha_i^*) = 0$
- $L(w, \alpha_i, \alpha_i^*) = \frac{1}{2} w^T w + \sum_{i=1}^m \alpha_i (\langle w, x_i \rangle) - \sum_{i=1}^m \alpha_i^* (\langle w, x_i \rangle) + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i + \sum_{i=1}^m (\alpha_i^* + \alpha_i) \varepsilon$
- From $\frac{\partial L(w, b, \alpha_i, \alpha_i^*)}{\partial w} = 0$, the parameters w can be written as: $w = \sum_{i=1}^m (\alpha_i^* - \alpha_i) x_i$
- $L(\alpha_i, \alpha_i^*) = \frac{1}{2} (\sum_{i=1}^m (\alpha_i^* - \alpha_i) x_i)^T (\sum_{i=1}^m (\alpha_i^* - \alpha_i) x_i) + \sum_{j=1}^m \alpha_j (\sum_{i=1}^m (\alpha_i^* - \alpha_i) x_i)^T x_j - \sum_{j=1}^m \alpha_j^* (\sum_{i=1}^m (\alpha_i^* - \alpha_i) x_i)^T x_j + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i + \sum_{i=1}^m (\alpha_i^* + \alpha_i) \varepsilon$



Support Vector Regression

$$J = (\hat{y} - y)^2$$

$$J = (wx + b - y)^2$$

$$(w^* x^2 + w^{*x} x + b)$$

- $$L(\alpha_i, \alpha_i^*) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) x_i^T x_j - \sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) x_i^T x_j + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i + \sum_{i=1}^m (\alpha_i^* + \alpha_i) \varepsilon$$

- Finally

- $$L(\alpha_i, \alpha_i^*) = -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) x_i^T x_j + \sum_{i=1}^m (\alpha_i^* - \alpha_i) y_i + \sum_{i=1}^m (\alpha_i^* + \alpha_i) \varepsilon$$

$\underbrace{\hspace{1.5cm}}_w \quad \underbrace{\hspace{1.5cm}}_w \quad \underbrace{\hspace{1.5cm}}_a$

$\underbrace{aw^2 + bw}_{\text{final}}$

- where $\sum_{i=1}^m (\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0, C]$



Support Vector Regression

- Solving for α_i and α_i^* from the previous optimization problem, the values of x_i for which their respective value of α_i, α_i^* is different from zero are the **support vectors** of the function.
- Then the function ($f(x)$) initially proposed to perform the regression is a function of the alpha parameters found when solving the dual optimization problem.

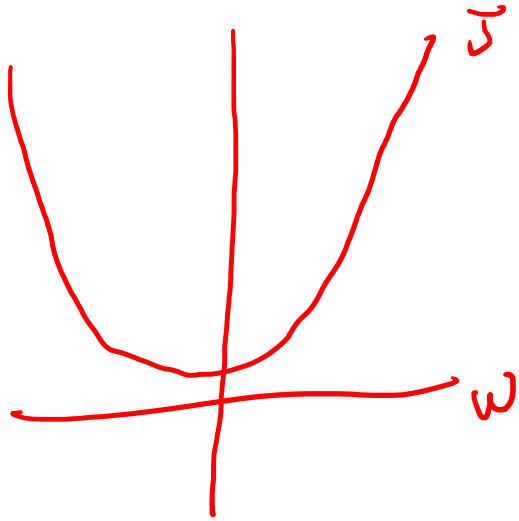
$$y = w x_i + b$$

$$f(x) = \langle w, x \rangle + b = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \underbrace{x_i^T x}_{\alpha_i \quad k(x)} + b$$

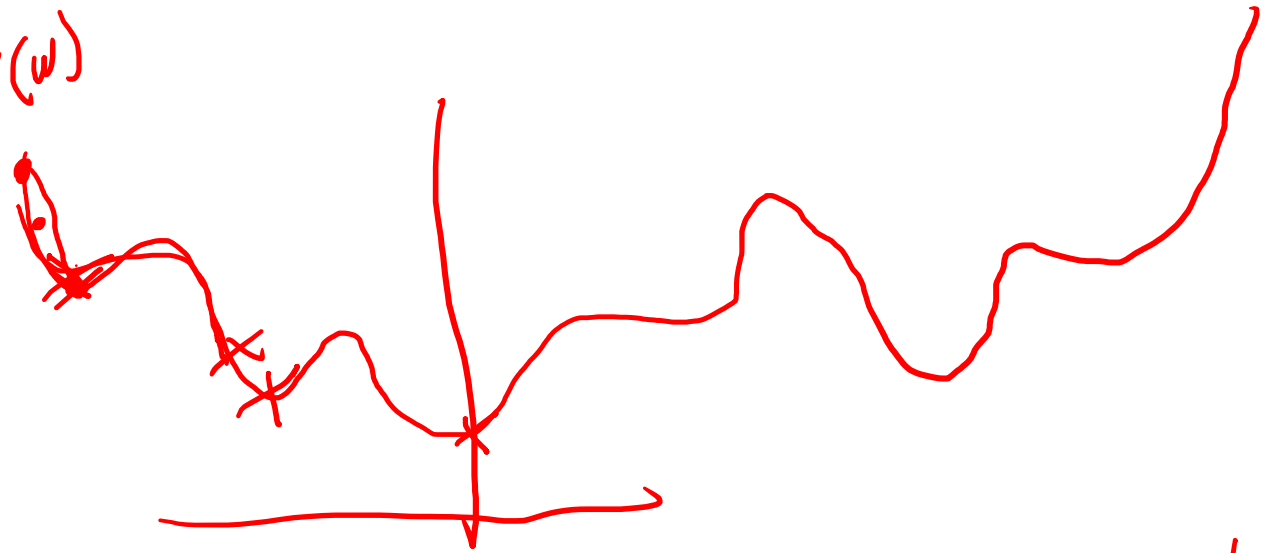
$$y = \alpha \tilde{x}^* + b$$



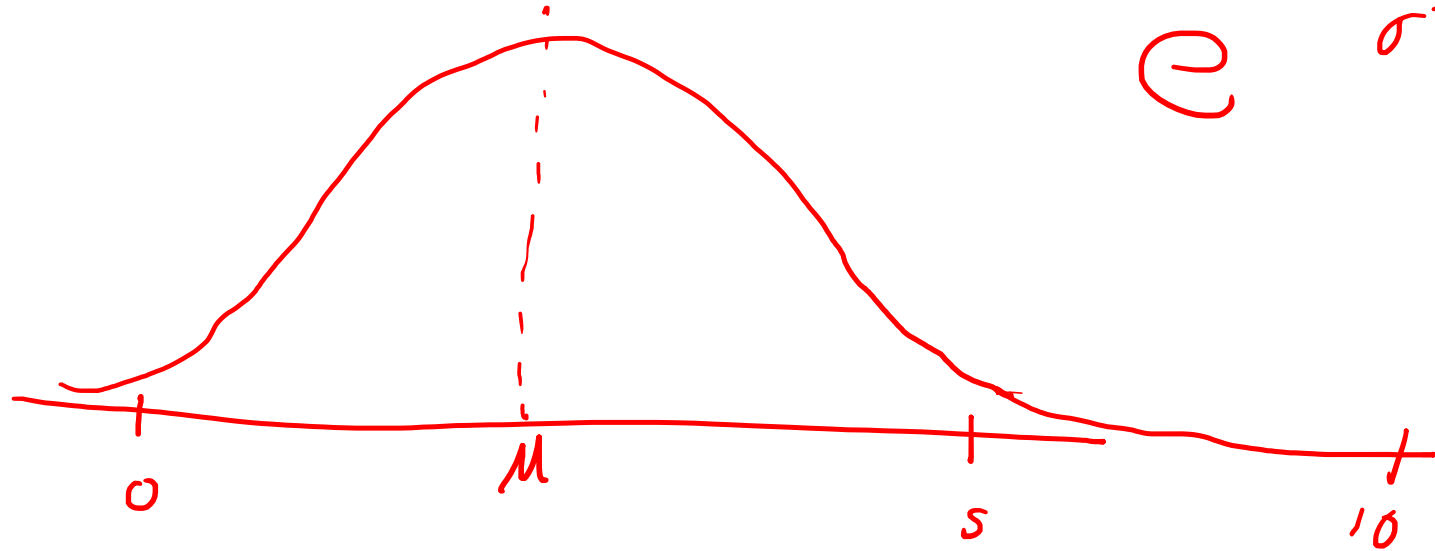
$J(w)$



$J(w)$



$$e^{-\frac{\|x - \mu\|^2}{\sigma^2}}$$





Support Vector Regression

- If $\varphi(\cdot)$ is defined as a transformation that is applied to the data before modeling. For the linear case we can say that $\varphi(x_i) = x_i$, then the complete problem can be written as

↙ no overfit

- $\min_w \frac{1}{2} \|w\|^2$

- subject to constraints $\begin{cases} \langle w, x_i \rangle + b - y_i \leq \varepsilon \\ y_i - \langle w, x_i \rangle - b \leq \varepsilon \end{cases}$

↙ regression

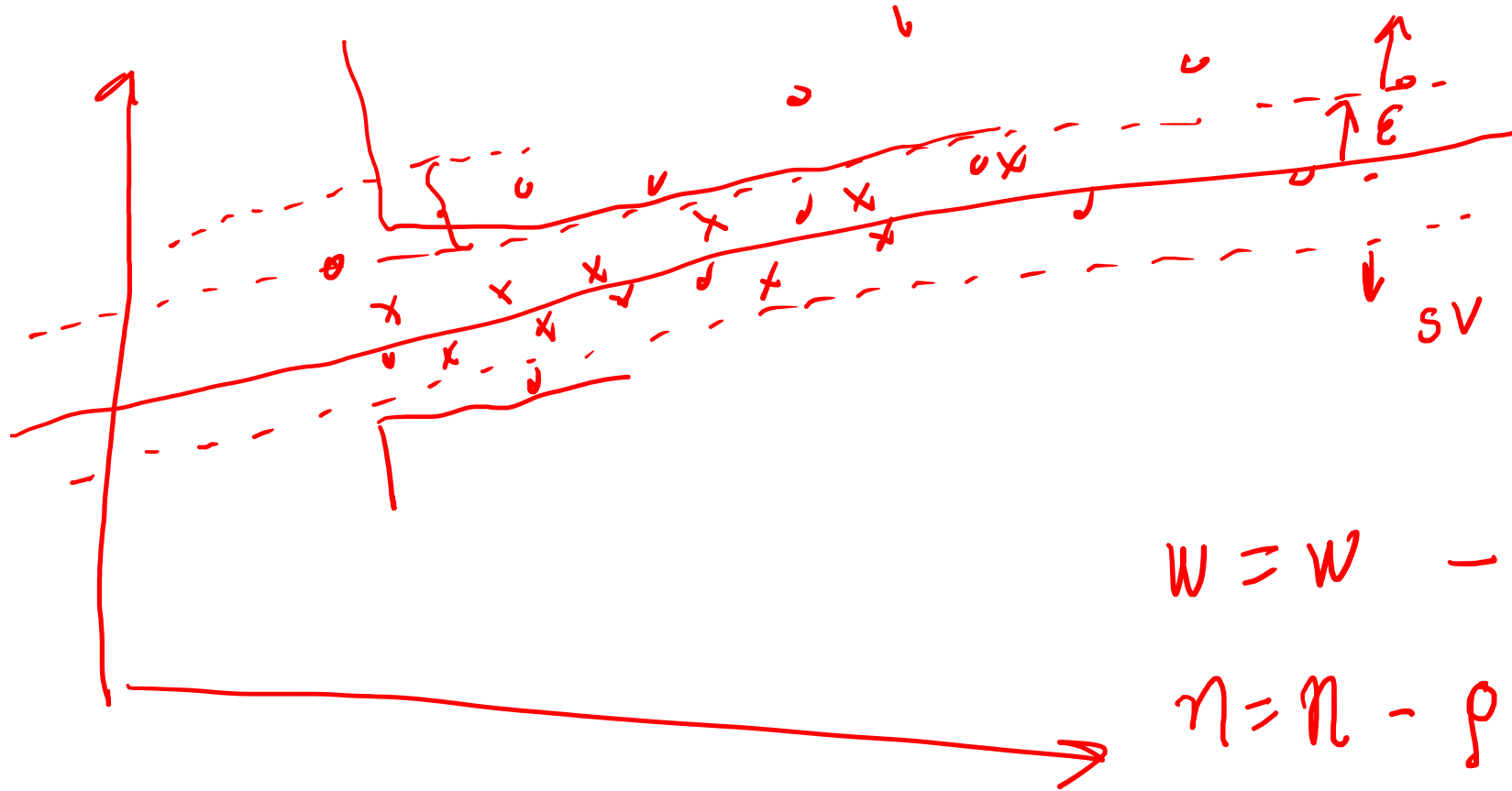
- It results in

- $w = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \varphi(x_i)$

- $f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \varphi(x_i) \varphi(x) = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) K(x_i, x)$

- $K(x_i, x) = \varphi(x_i) \varphi(x)$

Kernel transformation ↙



$$w = w - \eta \frac{\partial J}{\partial w}$$
$$\eta = \eta - \rho \frac{\partial J}{\partial \eta}$$





Support Vector Regression

Cost functions





Cost Functions

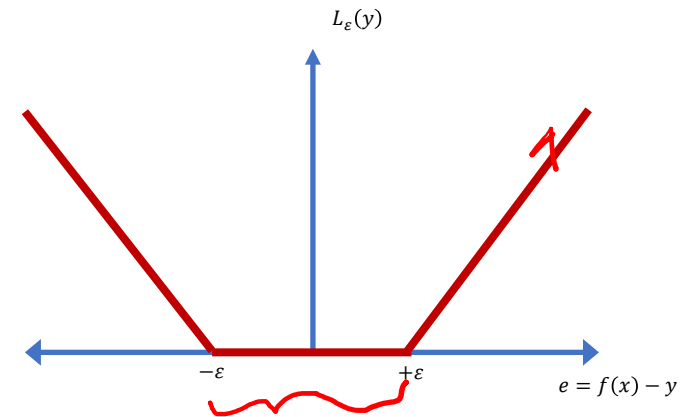
- In general, the optimization problem consists of an objective function with constraints.

- $\min_w \frac{1}{2} \|w\|^2$, with constraints
$$\begin{cases} \langle w, x_i \rangle + b - y_i \leq \varepsilon \\ y_i - \langle w, x_i \rangle - b \leq \varepsilon \end{cases}$$

- **Cost function: ε -insensitive**



- $$L_\varepsilon(y) = \begin{cases} 0 & \text{if } |f(x) - y| \leq \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases}$$





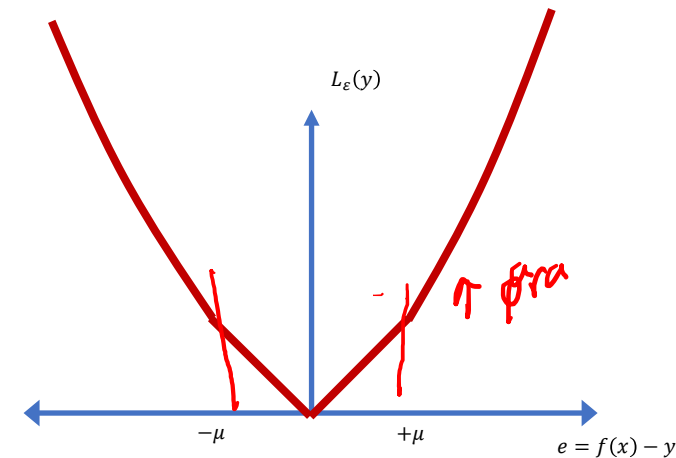
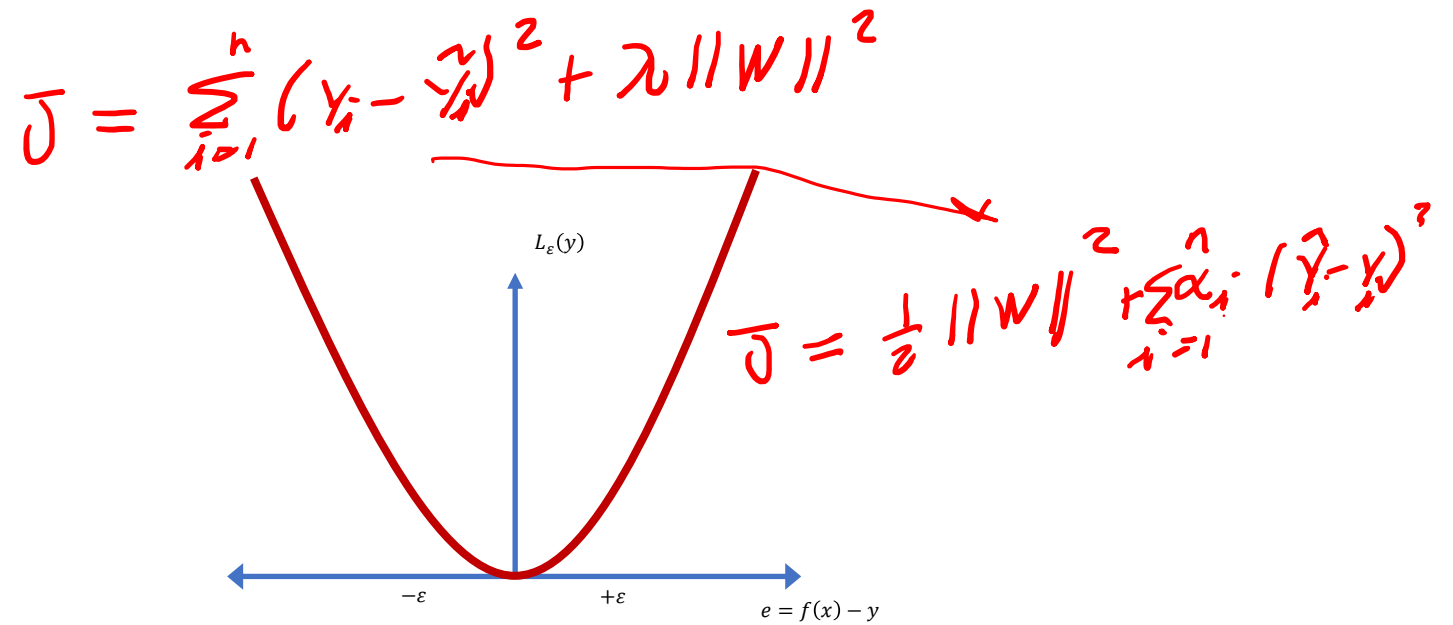
Cost Functions

- Cost function cuadratic.

- $L_{quad}(y) = (f(x) - y)^2$

- Hubber cost function.

- $L_{hubber} = f(x) = \begin{cases} \frac{1}{2} (f(x) - y)^2 & |f(x) - y| < \mu \\ \mu |f(x) - y| - \frac{\mu^2}{2} & \text{otherwise} \end{cases}$





Kernel transformations





Kernel transformations

- Kernel definition. $K(x_i, x) = \varphi(x_i) \varphi(x)$

- Linear Kernel.

$$K(x^*, x) = x^* \cdot x$$

- Polynomial Kernel.

$$K(x^*, x) = (x^* \cdot x)^d$$
$$K(x^*, x) = (x^* \cdot x + 1)^d$$

coef

- Radial basis function Kernel.

$$K(x^*, x) = e^{-\frac{\|x - x_c\|^2}{2\sigma^2}} = e^{-\gamma \|x - x_c\|^2}$$

✓
↓
↑

$\gamma = \frac{1}{2\sigma^2}$





Kernel transformations

- Exponential radial base

$$K(x^*, x) = e^{-\frac{\|x - x_c\|}{2\sigma^2}}$$

- Multilayer Perceptron: A neural network with a single hidden layer can also be considered as a valid Kernel transformation.



$$K(x^*, x) = \tanh(\overset{\downarrow}{\alpha}(x^* \cdot x) + \overset{\downarrow}{\beta})$$

$\tanh(w_i x + w_o)$



Example

P4_SVR.py





Support Vector Classification

SVC





Classification problem

- This problem requires finding a function that maps two data sets.

$$Y = f(X)$$

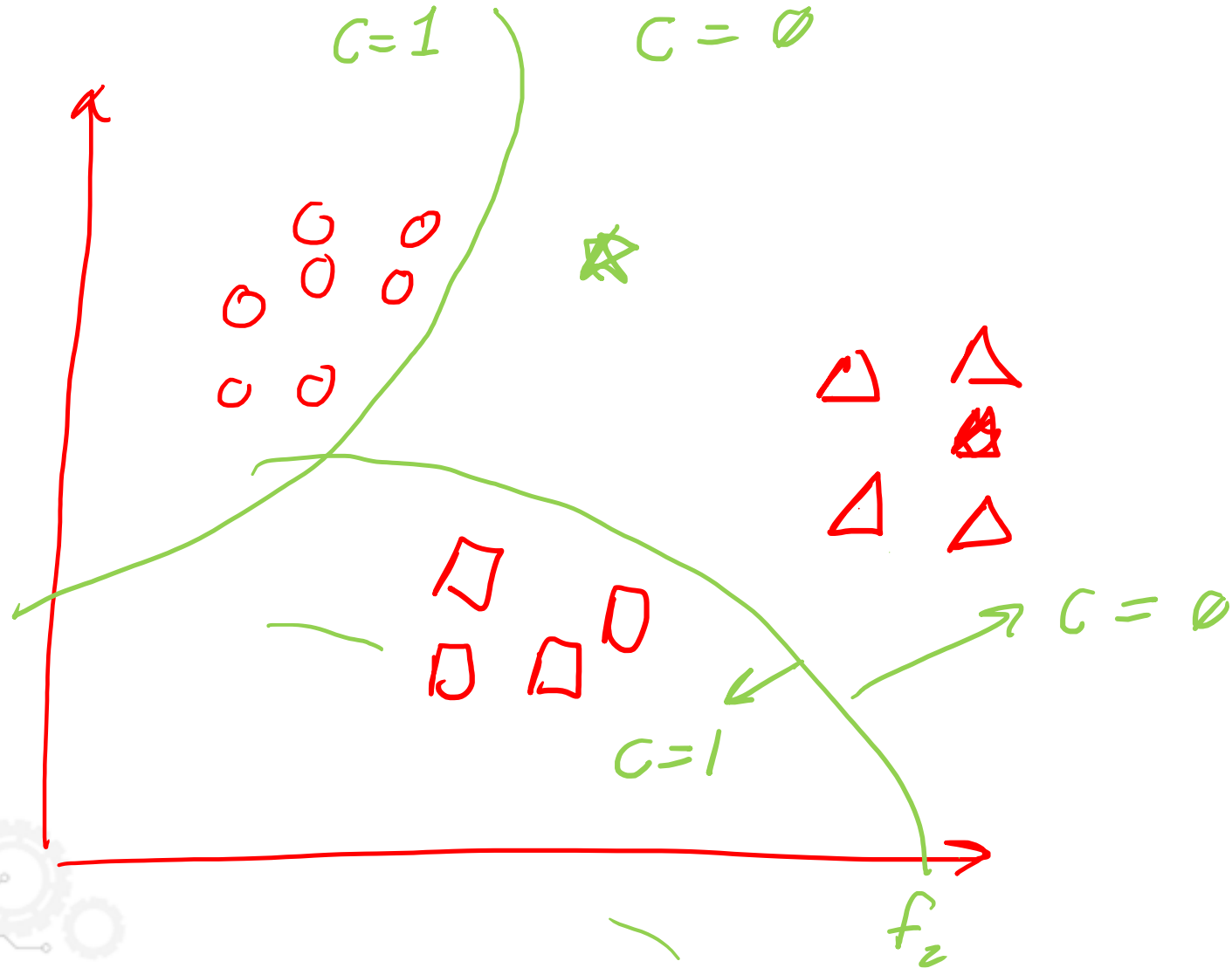
- X , are vectors with the input data, where each column is the predictor p .

$$X = [x_1, x_2, \dots, x_p], x_p \in R$$

- Y , are vectors with the output data, where each column is a desired output.

$$Y = [y_1, y_2, \dots, y_s], y_s \in \{0,1\}$$







Classification with support vector machines

- This problem requires finding a function that maps two data sets.

$$Y = f(X)$$

- X , are vectors with the input data, where each column is the predictor p .

$$X = [x_1, x_2, \dots, x_p], x_p \in R$$

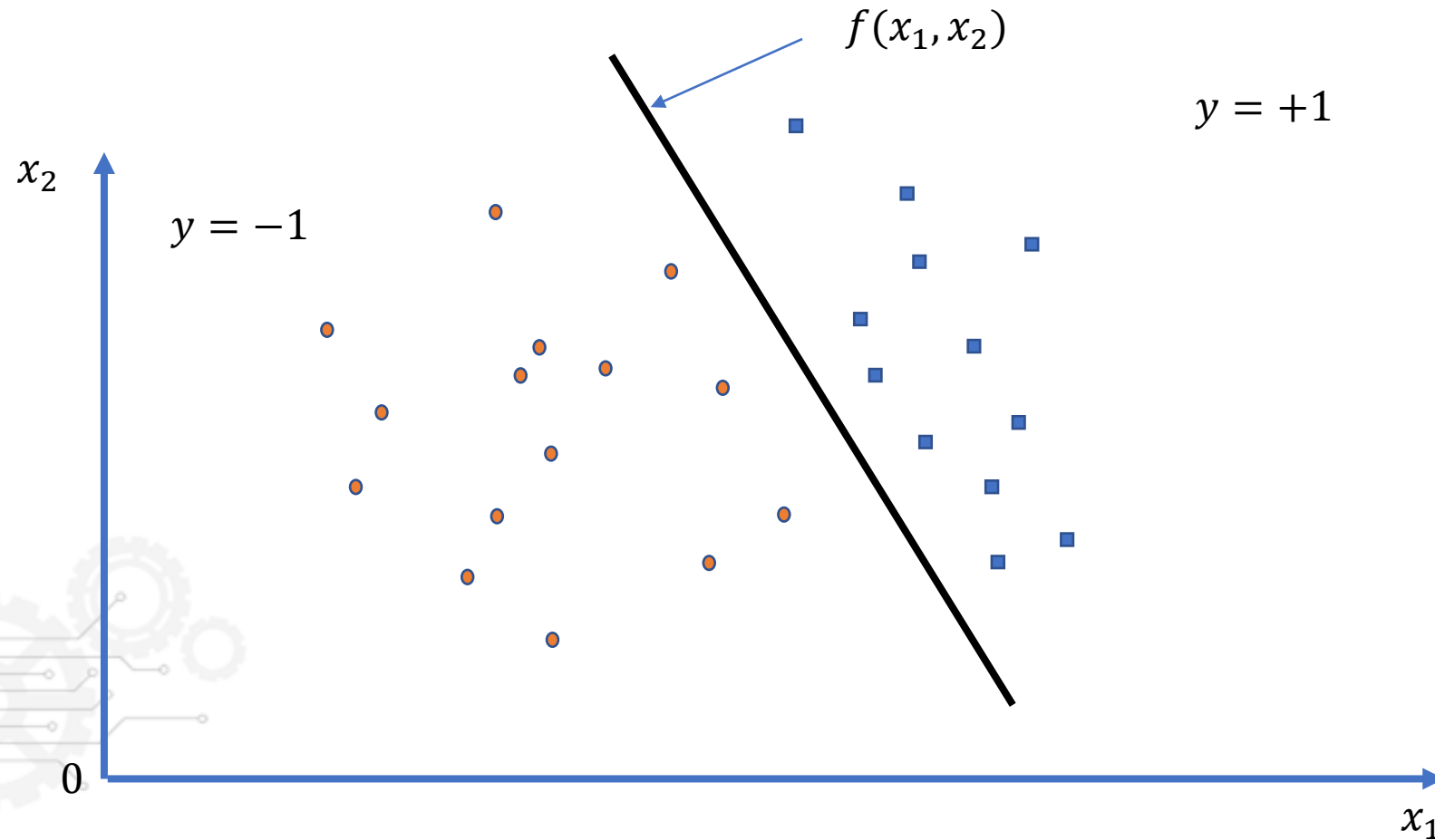
- Y , are vectors with the output data, where each column is a desired output.

$$Y = [y_1, y_2, \dots, y_s], y_s \in \{-1, 1\}$$





Classification with support vector machines





Classification with support vector machines

- Generalizing, for all samples in the data set the following condition must be met.

$$\underbrace{y_i(w^T x_i + b)}_{> 0} > 0$$

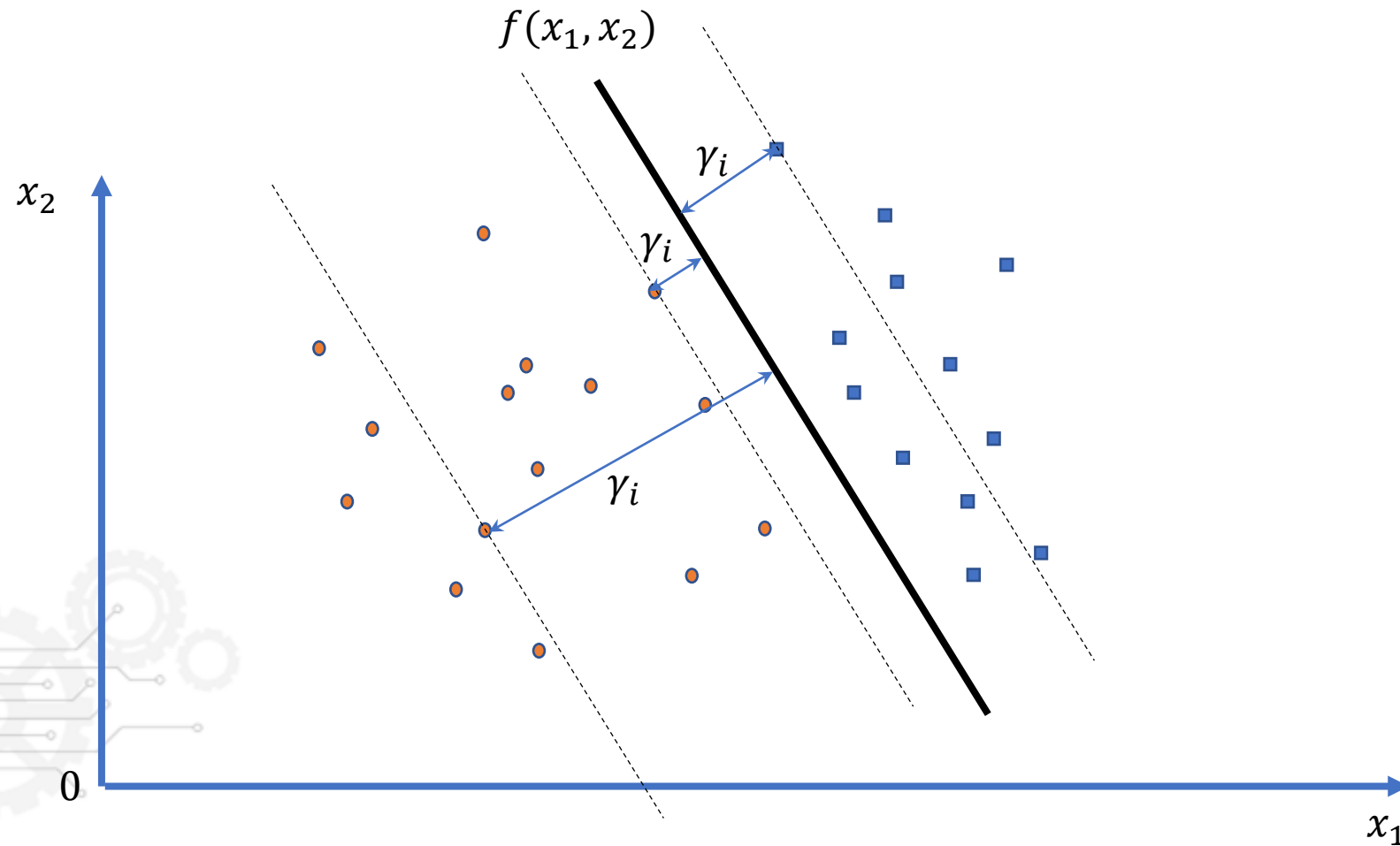
- To make compliance with the classification stricter, the restriction is reconsidered as

$$y_i(w^T x_i + b) \geq C$$





Classification with support vector machines





Classification with support vector machines

- The perpendicular distance of a sample x_i to the plane of the function or separation boundary is defined as

$$\gamma_i = \frac{w^T}{\|w\|} x_i + \frac{b}{\|w\|}$$

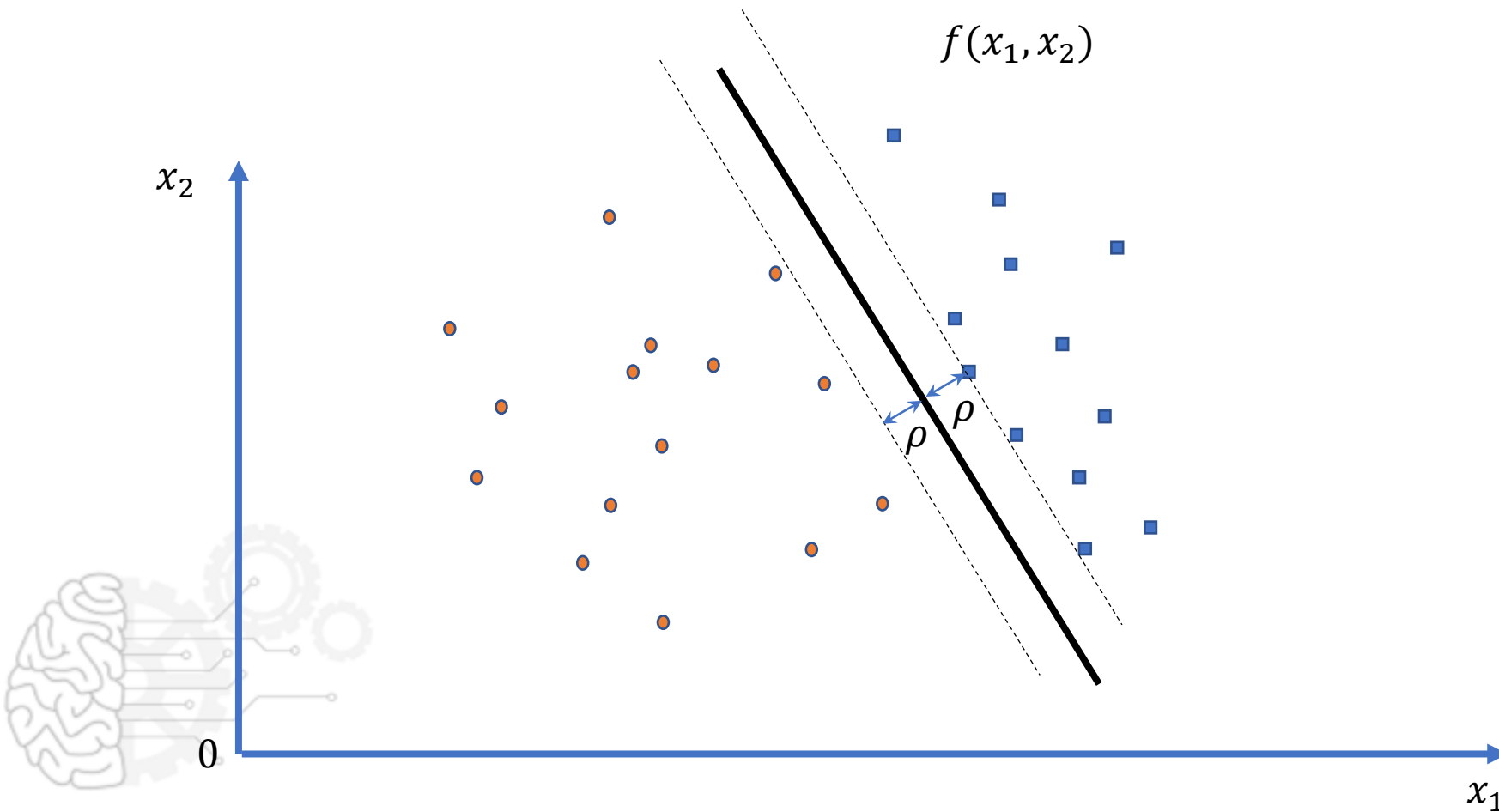
- The margin γ of the boundary, is the minimum perpendicular distance to the plane of the function or separation boundary. That is, the distance of the sample or samples closest to the separation plane

$$\rho = \min_{i=1, \dots, m} \gamma_i$$





Classification with support vector machines





Classification with support vector machines

- To achieve the best separation, a separation plane is sought that has the maximum margin.
- To find the maximum margin, the following optimization problem must be solved

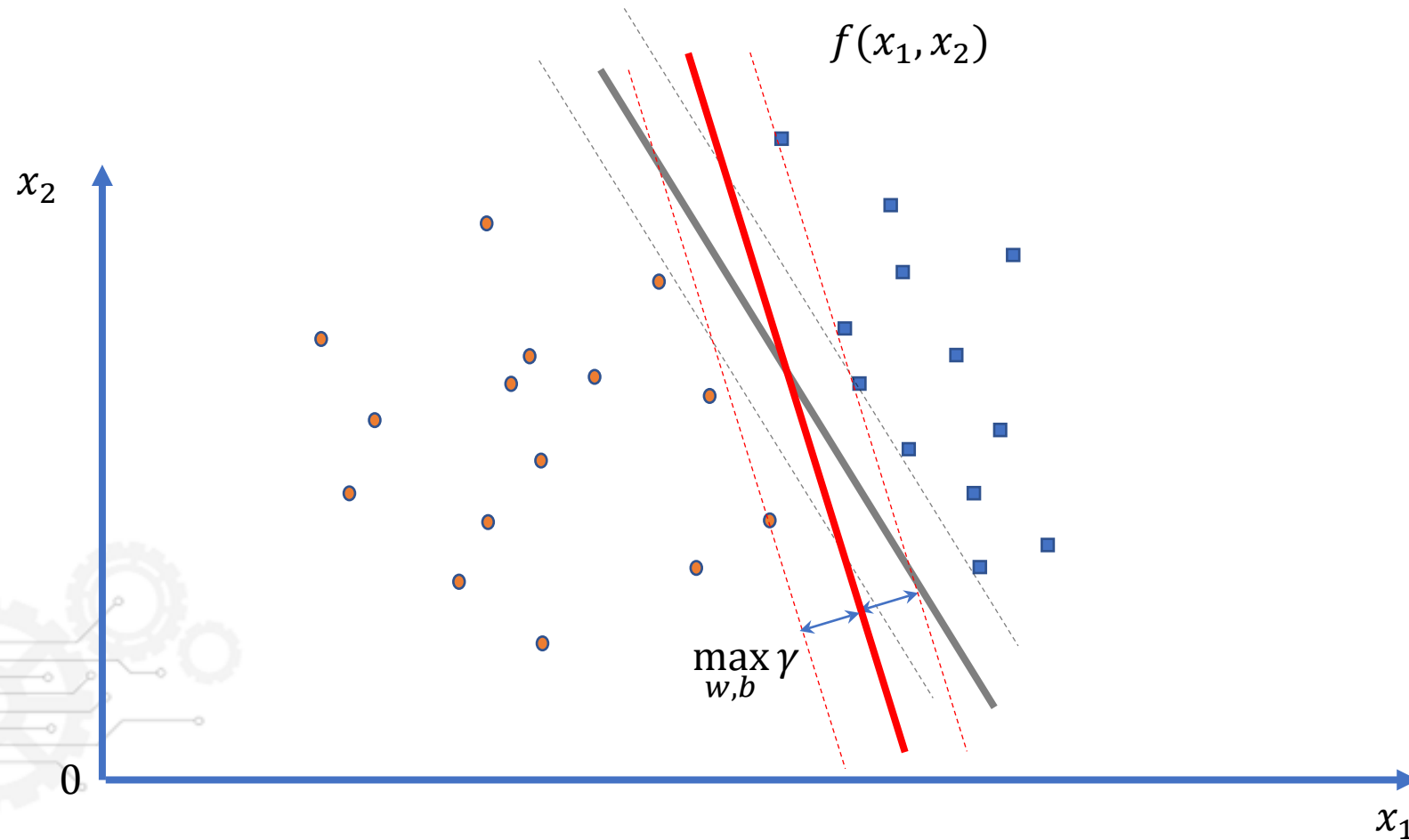
$$\max_{w,b} \rho$$

- subject to $y_i(w^T x_i + b) \geq \gamma$





Classification with support vector machines





Classification with support vector machines

- The complete optimization problem can be stated as:

$$\max_{w,b} \rho = \max_{w,b} \min_{i=1,\dots,m} \gamma_i$$
$$\max_{w,b} \min_{i=1,\dots,m} \frac{w^T x_i}{\|w\|}$$

- To simplify the optimization problem, the optimization objective is reformulated as:

$$\min_{w,b} \|w\|, \text{ subject to } y_i (w^T x_i + b) \geq \gamma, \gamma > 0$$





Classification with support vector machines

- To reduce computational load

$$\min_{w,b} \frac{1}{2} \|w\|^2, \text{ subject to } y_i(w^T x_i + b) \geq C$$

- The final optimization problem can be posed by making use of Lagrange multipliers.

$$\min_{w,b} J = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i [y_i(w^T x_i + b) - C]$$





Classification with support vector machines

- Defining $\varphi(x_i)$ as the data transformation applied before modeling. For the linear case we can say that $\varphi(x_i) = x_i$.
- The classification problem using SVM is stated as

$$\min_{w,b} \frac{1}{2} \|w\|^2, \text{ subject to } y_i(w^T x_i + b) \geq C$$

$$w = \sum_{i=1}^{N_{sv}} \alpha_i y_i x_i = \sum_{i=1}^{N_{sv}} \alpha_i y_i \varphi(x_i)$$

$$f(x) = \sum_{i=1}^{N_{sv}} \alpha_i y_i \varphi(x_i) \varphi(x) + b$$

- where, $K(x_i, x) = \varphi(x_i) \varphi(x)$ is the Kernel transformation.





Resume





Resume

	Model	Cost Function	Training Algorithm
Regression	$w = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \varphi(x_i)$ $f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) \varphi(x_i) \varphi(x) + b$ $= \sum_{i=1}^{n_{sv}} (\alpha_i^* - \alpha_i) K(x_i, x) + b$	$\min_{w,b} \frac{1}{2} \ w\ ^2,$ <p>subject to</p> <ol style="list-style-type: none"> 1. $L_\varepsilon(y) = \begin{cases} 0 & \text{if } f(x) - y \leq \varepsilon \\ f(x) - y - \varepsilon & \text{otherwise} \end{cases}$ 2. $L_{quad}(y) = (f(x) - y)^2$ 3. $L_{hubber} = \begin{cases} \frac{1}{2} (f(x) - y)^2 & f(x) - y < \mu \\ \mu f(x) - y - \frac{\mu^2}{2} & \text{otherwise} \end{cases}$ 	<ol style="list-style-type: none"> 1. Descending gradient 2. Convex optimization algorithm 3. etc.
Classification	$w = \sum_{i=1}^{n_{sv}} \alpha_i y_i \varphi(x_i)$ $f(x) = \sum_{i=1}^{n_{sv}} \alpha_i y_i \varphi(x_i) \varphi(x) + b$ $= \sum_{i=1}^{n_{sv}} \alpha_i y_i K(x_i, x) + b$	$\min_{w,b} \frac{1}{2} \ w\ ^2,$ <p>subject to</p> <ol style="list-style-type: none"> 1. $y_i (w^T x_i + b) \geq C$ 	<ol style="list-style-type: none"> 1. Descending gradient 2. Convex optimization algorithm 3. etc.

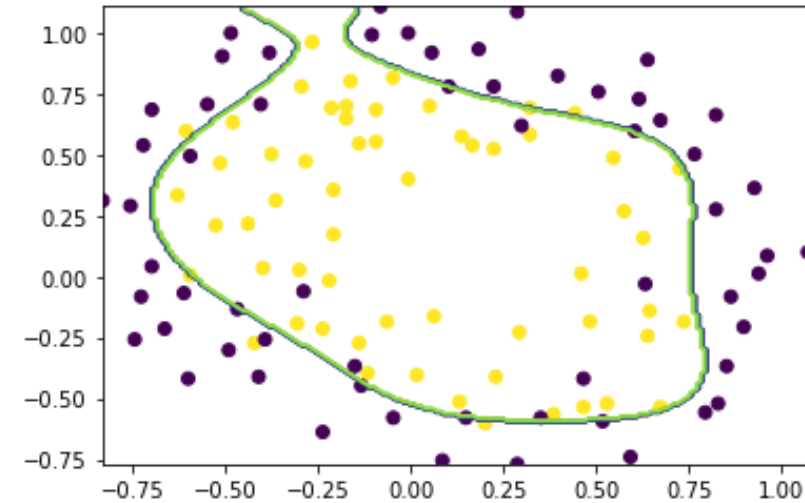
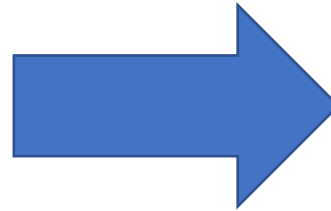
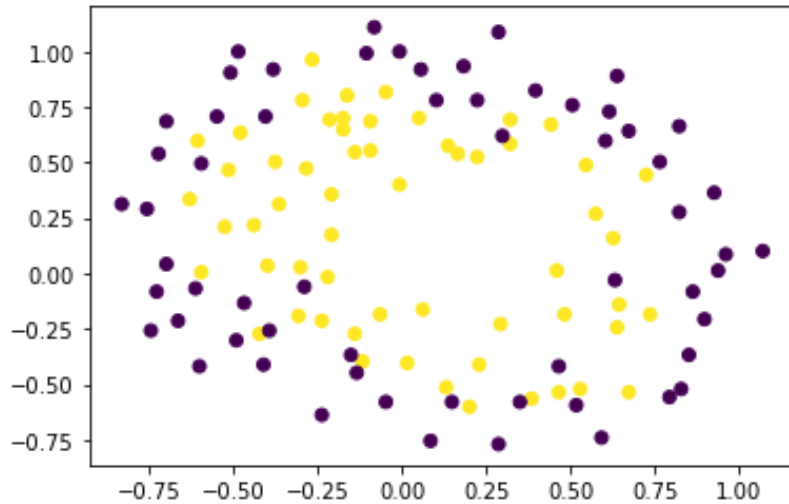



Logistic Regression vs Support Vector Classifier



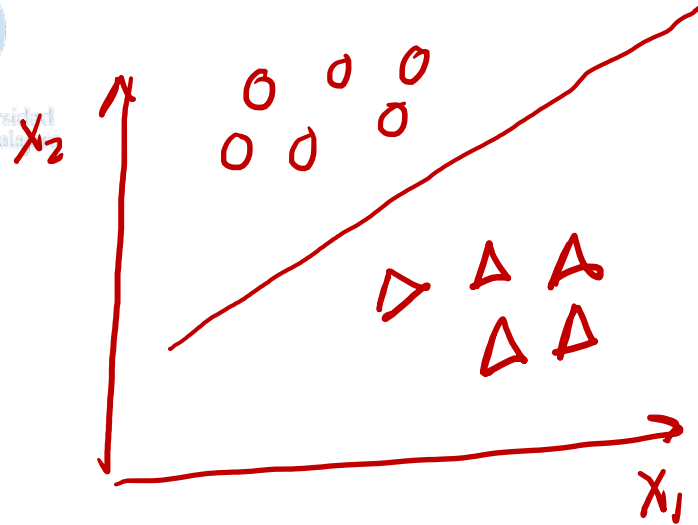


Logistic Regression




$$[x_1, x_2] \rightarrow \{0,1\}$$

$$\hat{y} = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + \dots + w_n x_1^p + w_m x_2^p)}}$$



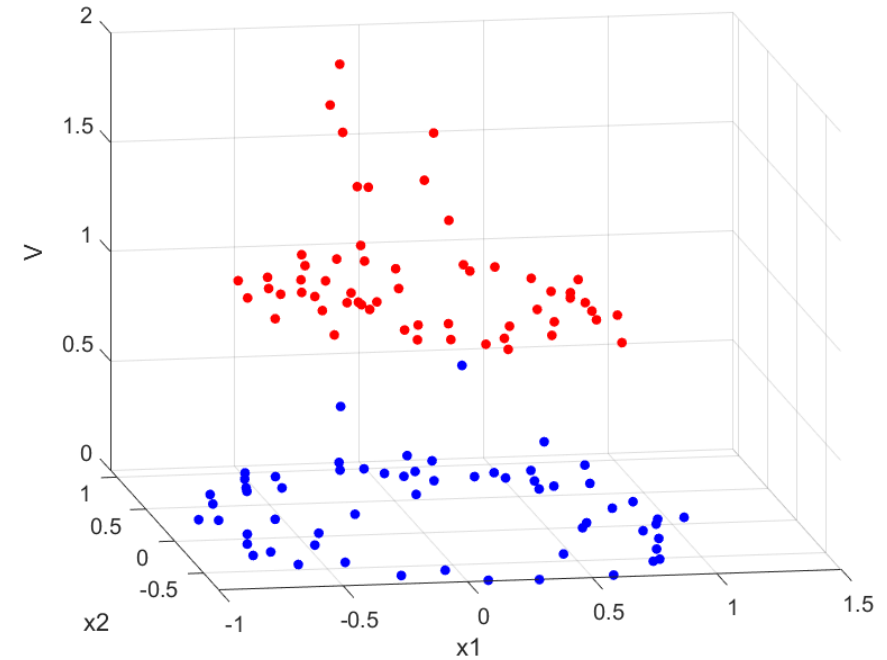
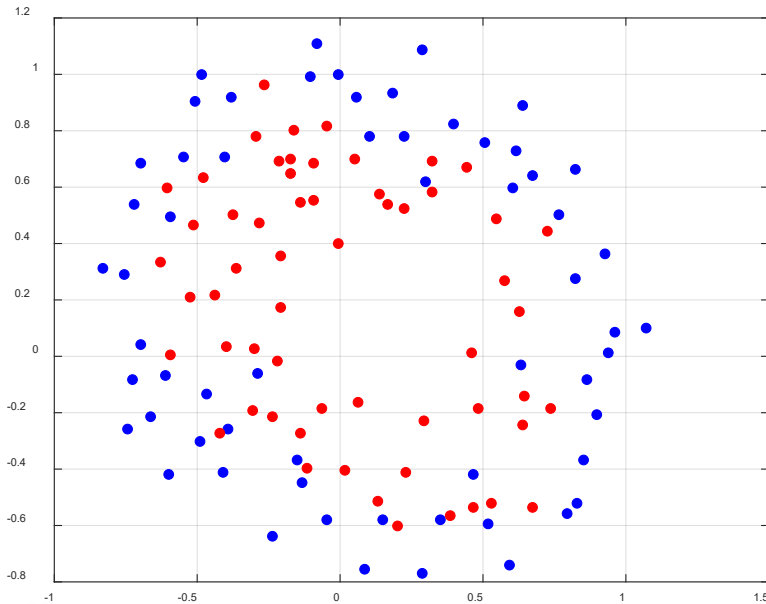
$$z = w_0 + w_1 x_1 + w_2 x_2$$


$$f(x) = \frac{1}{1 + e^{-z}}$$





Support Vector Classifier




$$[x_1, x_2] \rightarrow \{0,1\}$$

$$\hat{y} = \sum_{i=1}^{n_{sv}} \alpha_i y_i K(x_i, x) + b$$



Example

P5_SVM_SV.py





Model Optimization

High Bias vs High Variance





High Bias vs High Variance

- High Bias. It refers to when the model is not capable of having a good generalization because it is a very simple model or the model is over-regularized. It is commonly known as **underfitting**.
- High Variance. It refers to when the model is not capable of having a good generalization because it is a very complex model or the model is not sufficiently regularized. Commonly known as **overfitting**.





High Bias vs High Variance

