



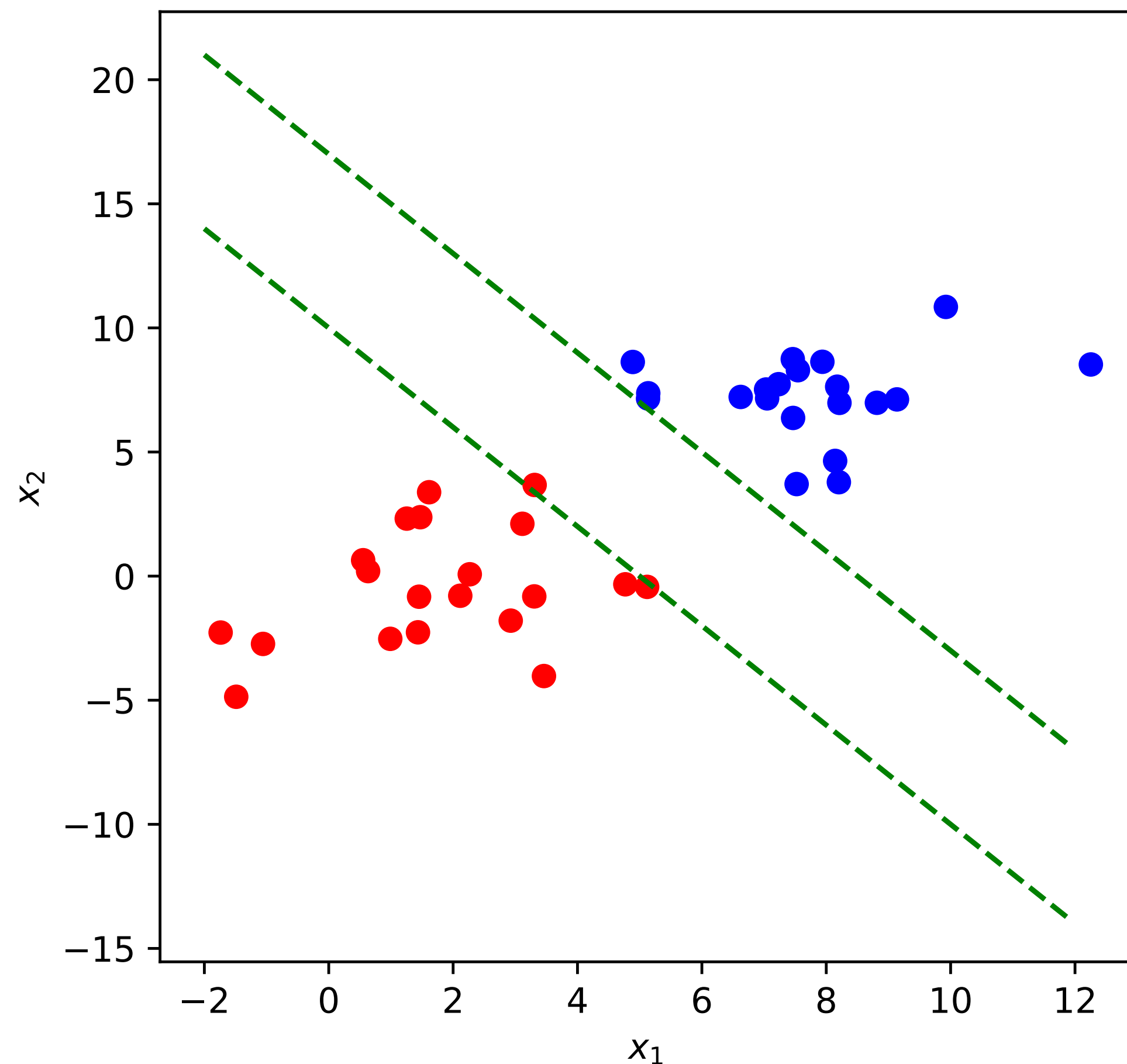
Support Vector Machines

Licenciatura en Inteligencia Artificial y Ciencia de Datos, CUGDL,
Universidad de Guadalajara.

Guadalajara, Jal., agosto de 2025

Support Vector Machines

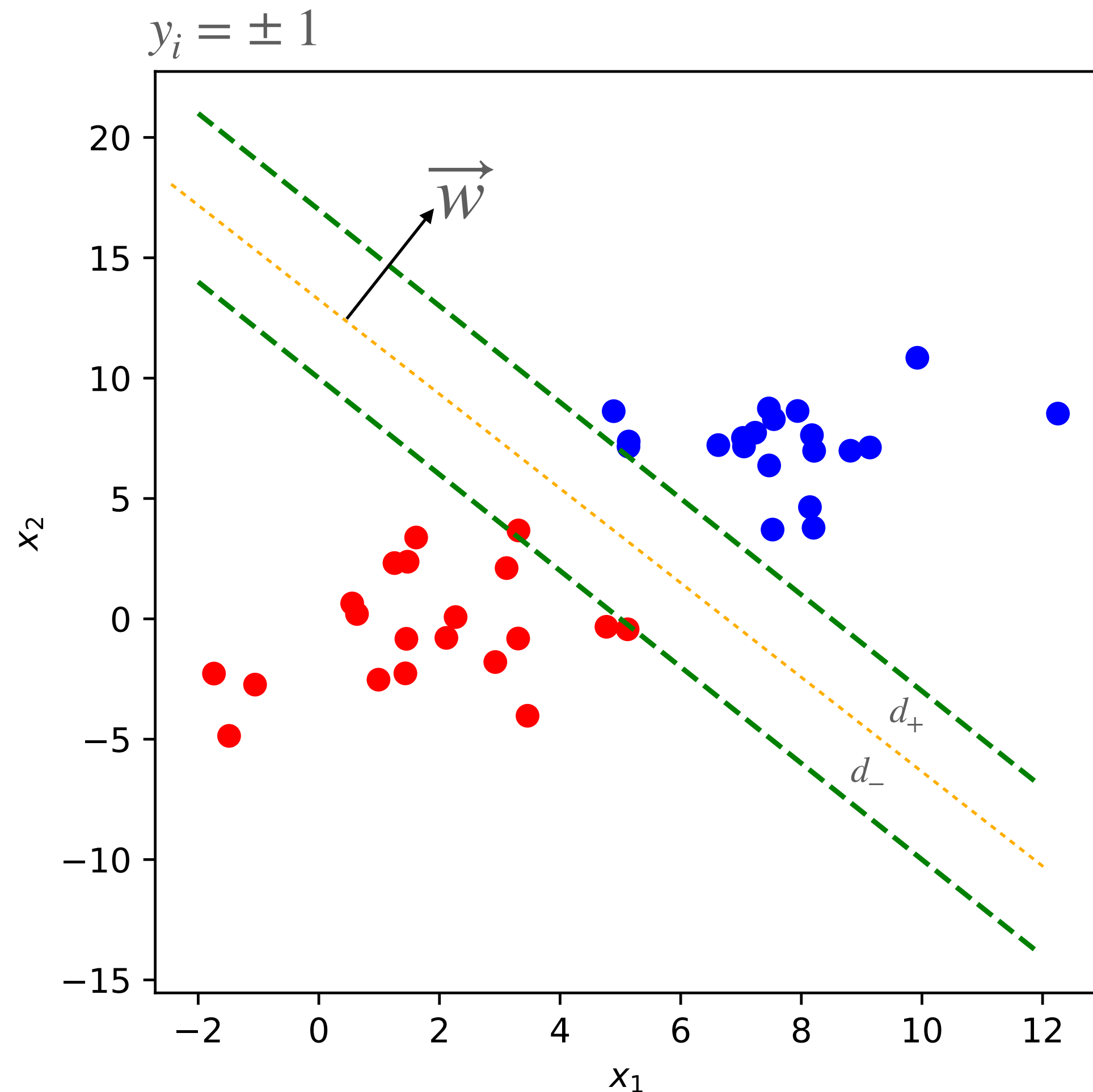
Introducción



- Algoritmo de ML supervisado de clasificación/regresión.
- El objetivo es optimizar un hiperplano para separar a los datos.
- El patrón que muestra este hiperplano no es necesariamente lineal. Se realizan transformaciones a los datos originales hacia otro espacio (kernel).
- Los **vectores de soporte** son asociados a los datos más cercanos a la superficie de decisión (hiperplano).
- El algoritmo de SVM busca maximizar el margen ("street") alrededor del hiperplano.

Support Vector Machines

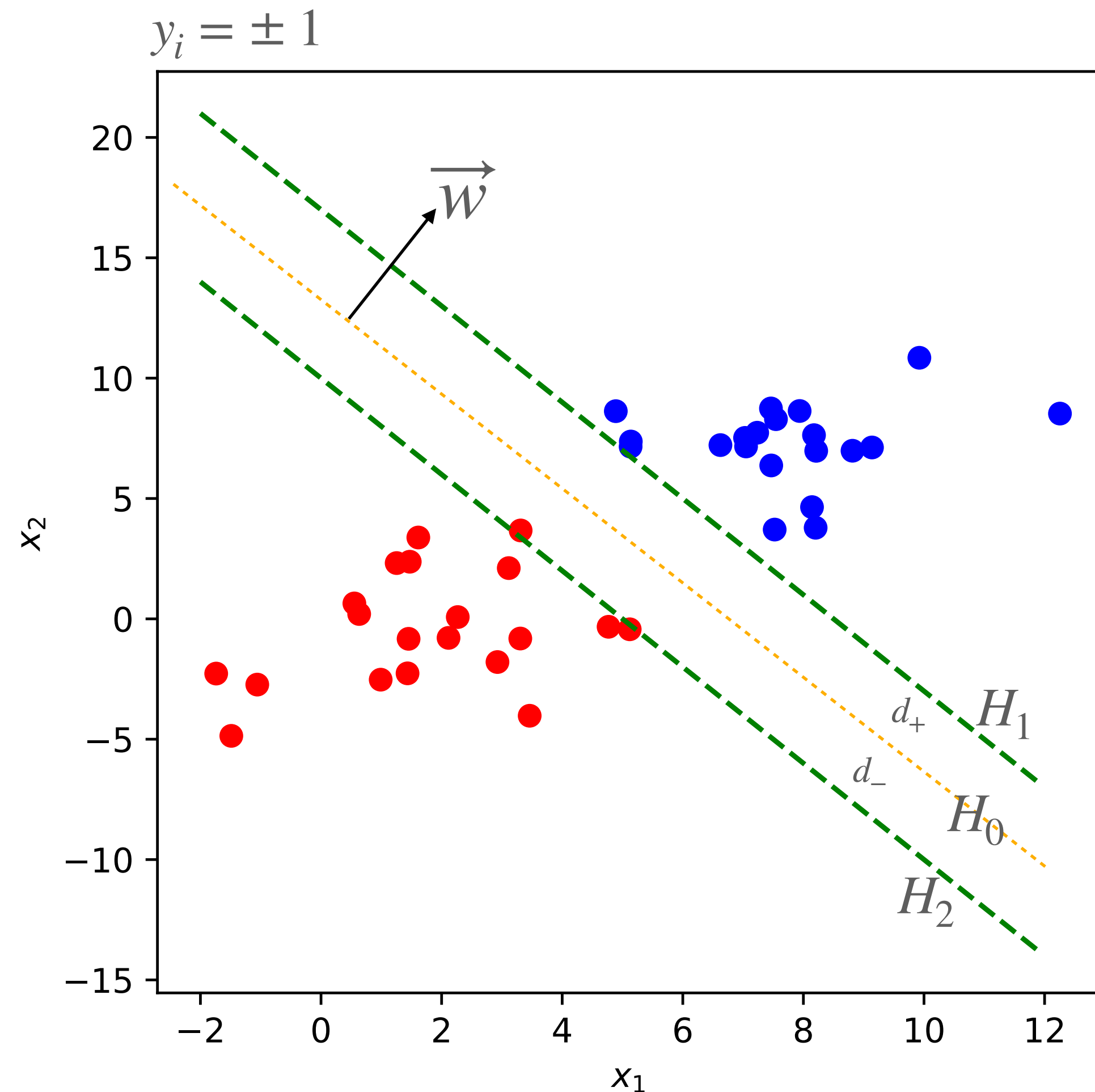
Separabilidad lineal



- Cualquier punto x que se encuentre en este hiperplano separador ($x \in H$) satisface la siguiente ecuación:
- $H_0 : x^T w + b = 0$
- w es un vector ortogonal a este hiperplano ($w \perp H$).
- El **objetivo** es encontrar un set de pesos w_i y los puntos b (bias) para cada uno de las variables predictoras, cuya combinación lineal predice un valor para y .

Support Vector Machines

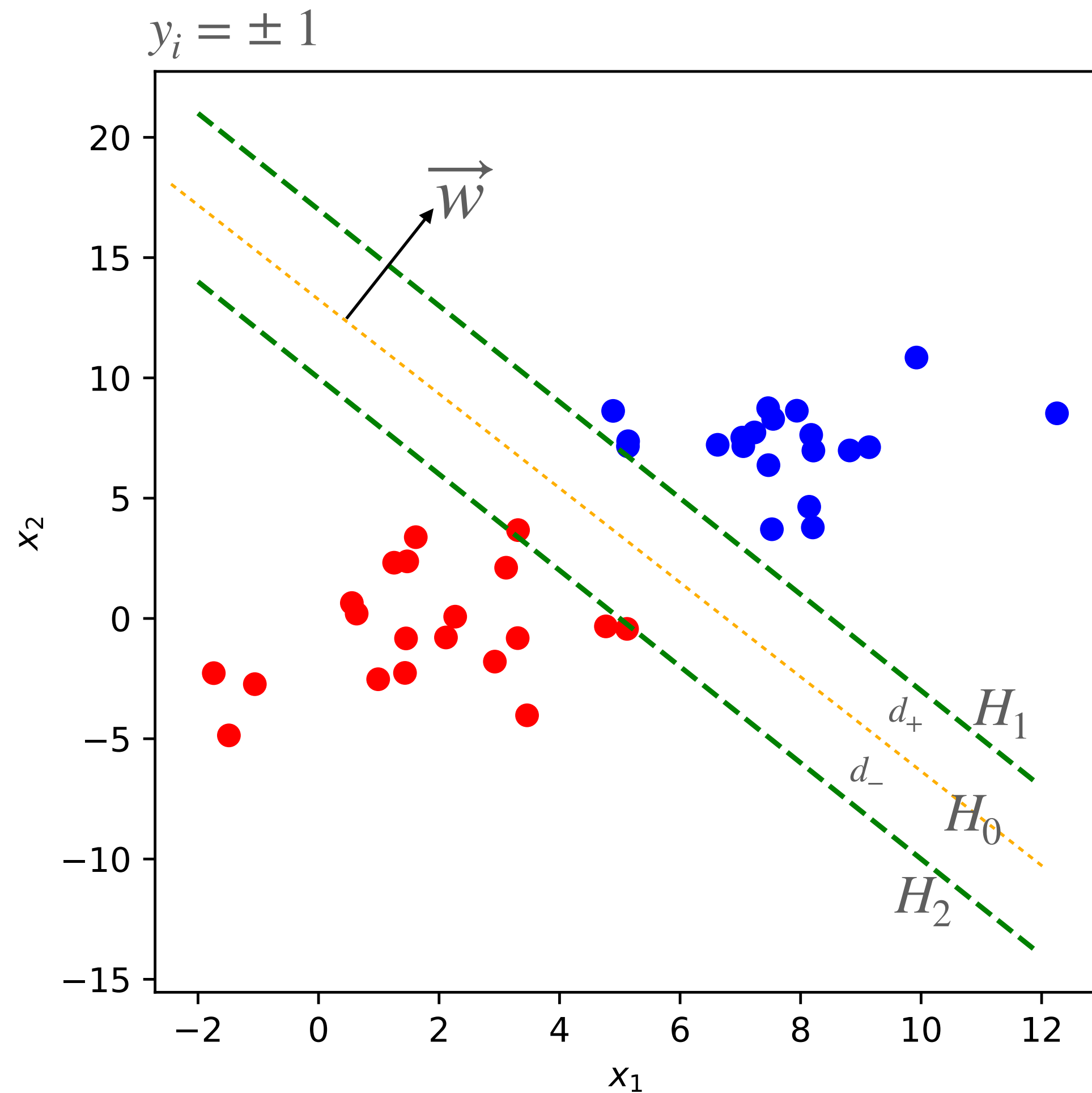
Separabilidad lineal



- La anchura de este margen es $d_+ + d_-$. Los puntos más cercanos a esta frontera crean otros dos hiperplanos paralelos al original.
- Esta separación clasifica los puntos, de tal forma que,
$$\begin{cases} x_i^T \cdot w + b \geq +a, & \text{si } y_i = +1 \\ x_i^T \cdot w + b \leq -a, & \text{si } y_i = -1 \end{cases}$$
- Entonces, estos dos hiperplanos adicionales serían,
- $H_1 : x^T \cdot w + b = +1$
- $H_2 : x^T \cdot w + b = -1$
- Y el margen se podría calcular con $M = d_+ + d_- = \frac{2a}{\|w\|}$
- Normalmente, la escala se escoge de tal forma que $a = 1$.

Support Vector Machines

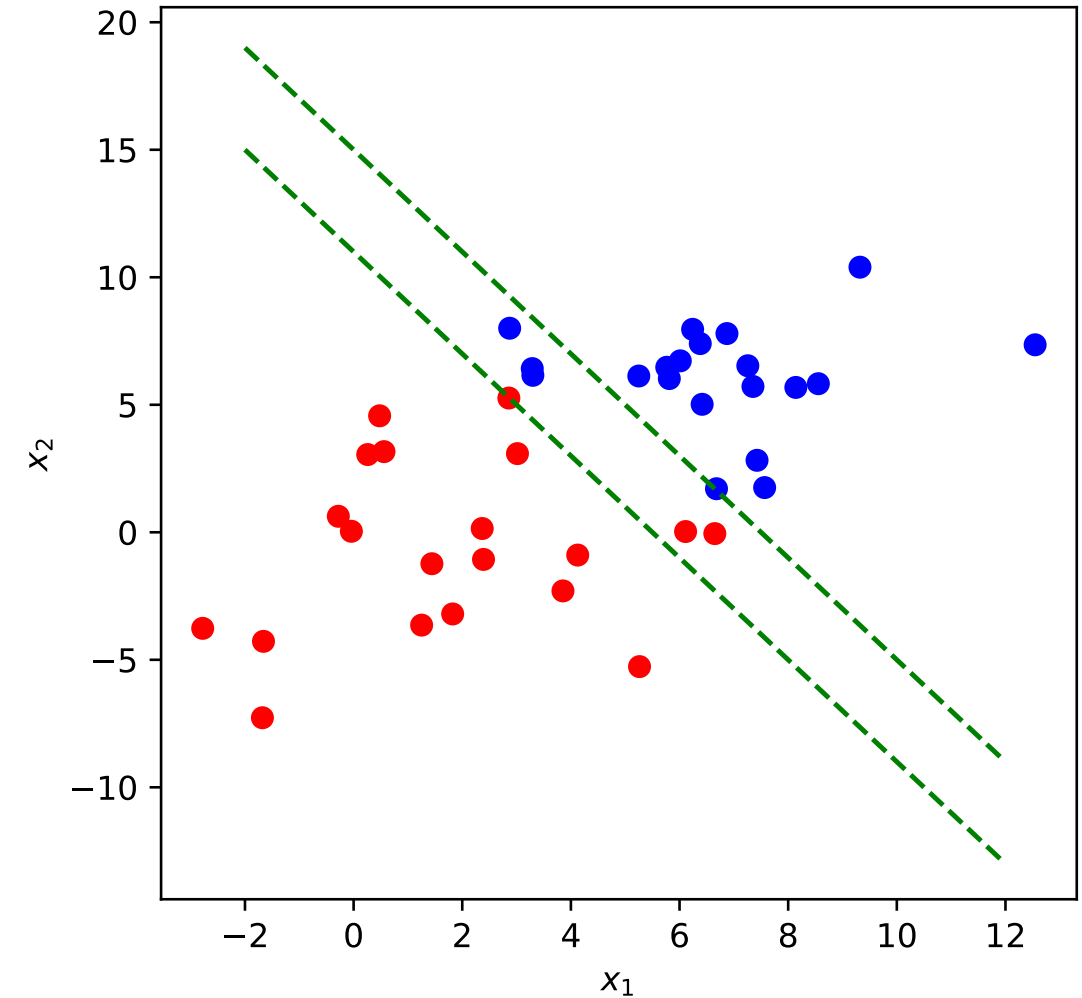
Optimización



- Entonces, se busca maximizar este margen, $M = d_+ + d_- = \frac{2}{\|w\|}$, con la condición de que no existan datos dentro de este margen.
- Hay dos opciones: se maximiza la M o se minimiza la w (en este caso se busca minimizar el cuadrado de esto).
- Queremos minimizar una función
- $f(w) = \frac{1}{2} \|w\|^2$
- sujeto a una restricción de forma
- $g(w, b) = -y_i(x_i^T w + b) + 1 \leq 0 \quad \forall i = 1, 2, \dots, n.$
- La función a maximizar es cuadrática (a diferencia del margen M), un paraboloides con un único punto mínimo global.
- **Sin embargo**, en muchos casos esto no va a tener solución, el hiperplano no existe. Tiene que considerarse o permitirse cierto overlapping de puntos.

Support Vector Machines

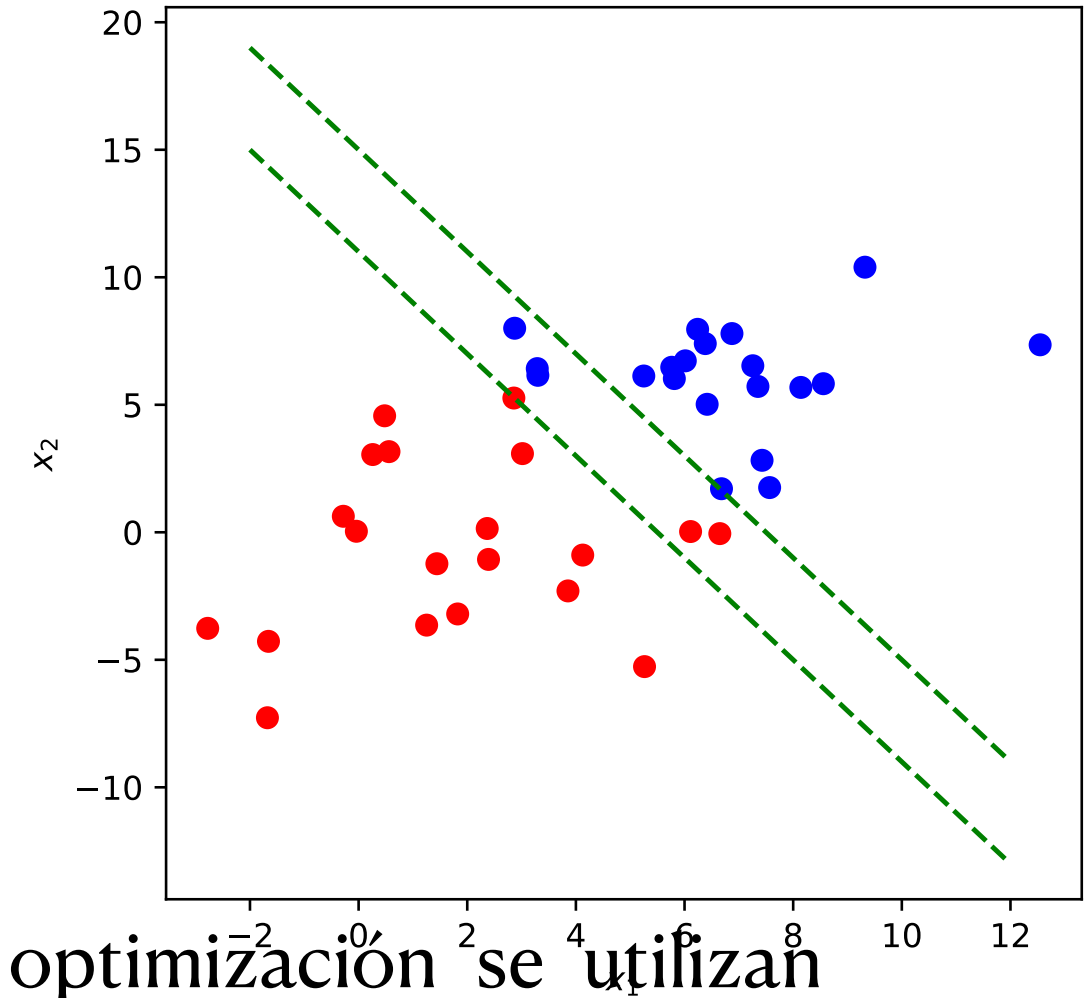
Optimización



- Consideramos un margen "casi separable", que permite que ciertos puntos se encuentren dentro del margen.
- Estas variables adicionales serán lo que se conocen como soportes vectoriales. Se agregan una serie de vectores adicionales $(\psi_1, \psi_2, \dots, \psi_n)$, con las siguientes restricciones:
- $\psi \geq 0$, positivo.
- De esta forma, la nueva función a **minimizar** es,
- $$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \psi_i$$
- Con constricciones
- $$y_i(x_i^T w + b) \geq 1 - \psi_i; \quad \sum_{i=1}^n \psi_i \leq C \text{ constante.}$$
- El valor ψ_i actúa como una cantidad proporcional de que el valor i-ésimo caiga del lado equivocado del margen. El hiperparámetro C del modelo SVM permite acotar el número total de errores en la frontera.

Support Vector Machines

Multiplicadores de Lagrange (Forma Primal)



- Este es un problema de optimización cuadrático, con restricciones lineales. Para resolver el problema de optimización se utilizan multiplicadores de Lagrange. Con las funciones anteriores se construye el Lagrangiano,

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \psi_i - \sum_{i=1}^n \alpha_i [y_i(x_i^T w + b) - (1 - \psi_i)] - \sum_{i=1}^n \mu_i \psi_i$$

- Queremos optimizar esta función y encontrar valores óptimos para w (vector normal al hiperplano H), b (ordenada al origen), ψ_i (cantidad de error). El método de multiplicadores de Lagrange involucra realizar derivada parcial de estas variables.

$$\frac{\partial L_p}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$\frac{\partial L_p}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^n \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial \psi_i} = C \sum_{i=1}^n 1 - \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \mu_i = 0 \quad \Rightarrow \quad \alpha_i = C - \mu_i$$

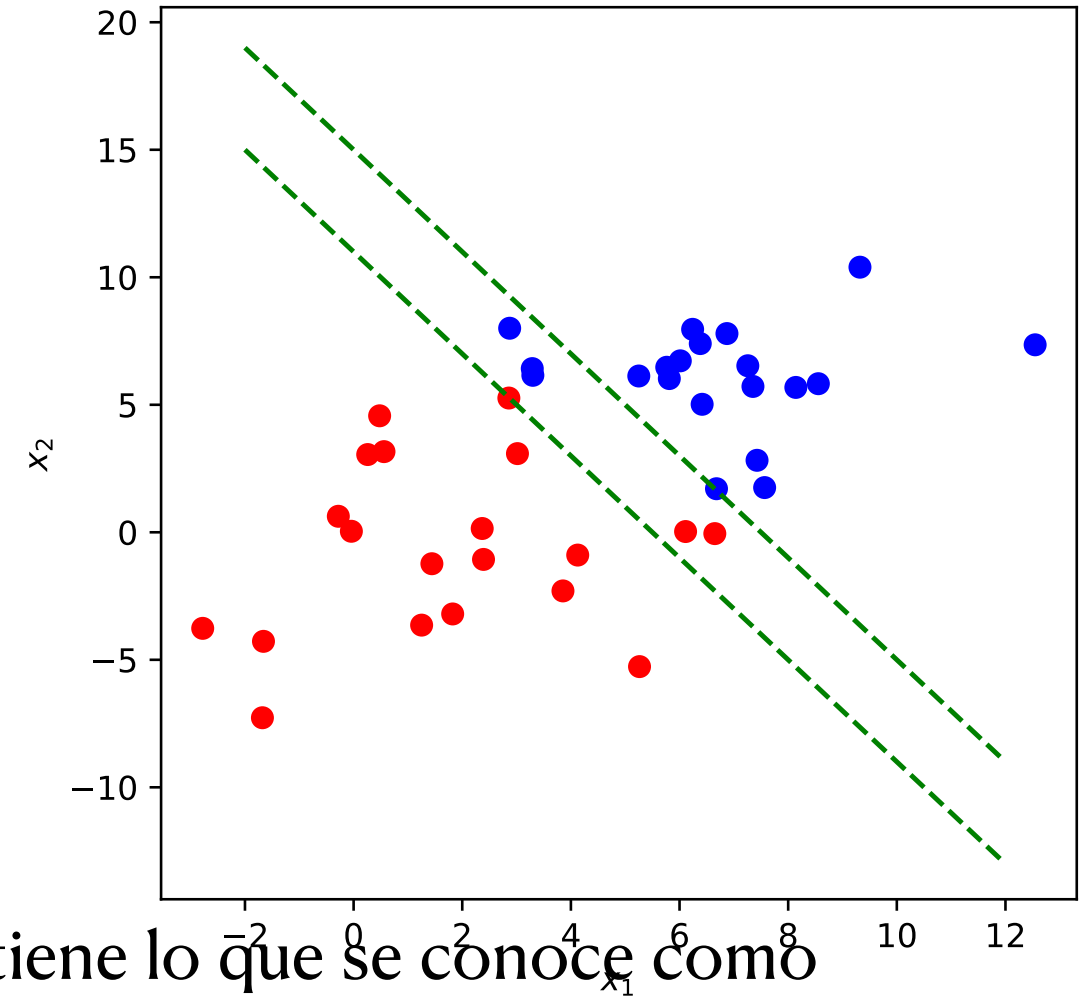
Condiciones de Karush-Kuhn-Tucker (KKT).

En optimización matemática, son condiciones necesarias de primer orden para que una solución en programación no lineal (NLP). sea óptima.

[Link](#) de consulta.

Support Vector Machines

Problema dual de Wolfe (forma dual)



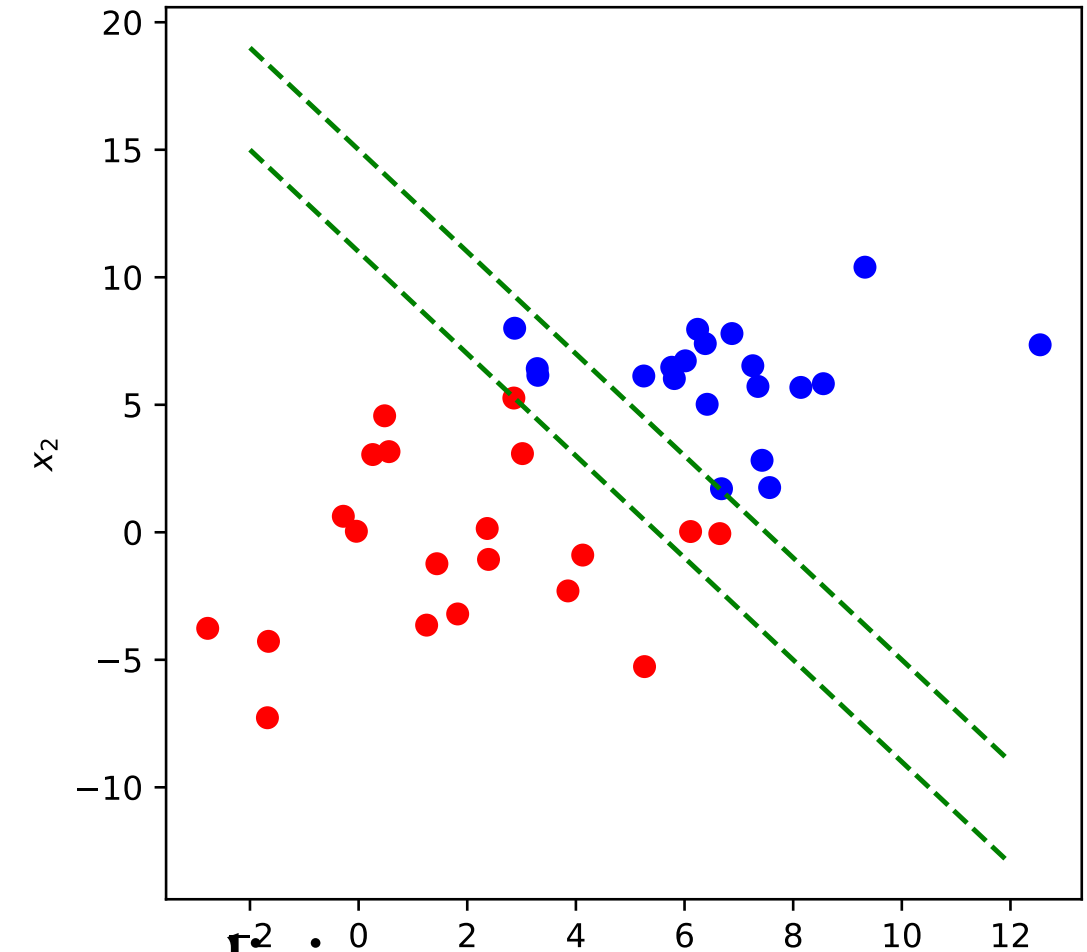
- El introducir estas condiciones en el Lagrangiano primal, ya no se tienen todas las variables que se tenían originalmente. Se obtiene lo que se conoce como problema dual de Wolfe (link), con el siguiente Lagrangiano,

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$

- Donde $0 \leq \alpha_i \leq C$ y con restricción $\sum_{i=1}^n \alpha_i y_i = 0$.
- En lugar de minimizar para w, b, ψ_i de la forma primal, se maximiza esta nueva función sobre α_i sujeto a las relaciones obtenidas previamente para w y b .
- Las condiciones de KKT para este caso son las siguientes,
- $\alpha_i [y_i(x_i^T w + b) - (1 - \psi_i)] = 0$
- $\mu_i \psi_i = 0$
- $y_i(x_i^T w + b) - (1 - \psi_i) \geq 0$

Support Vector Machines

Problema dual de Wolfe (forma dual)



- El teorema de Karush-Kuhn-Tucker menciona que en caso de existir un punto óptimo, se cumplen estas condiciones, que son las restricciones del problema primal original.
- Hay una solución tanto del problema primal como del dual con la forma obtenida,
- $w = \sum_{i=1}^n \alpha_i y_i x_i$ donde los valores α_i cumplen lo siguiente,
- $\alpha_i \neq 0 \iff y_i(x_i^T w + b) - (1 - \psi_i) = 0$. Es decir, los valores α_i solamente son no nulos cuando se trata de valores x_i dentro del margen/'street'/'corredor'. Estos valores x_i son llamados soportes vectoriales, donde particularmente,
 - $\begin{cases} 0 < \alpha_i < C & \text{si } \psi_i = 0 \text{ frontera} \\ \alpha_i = C & \text{si } \psi_i > 0 \text{ interior} \end{cases}$
- Cualquiera de los puntos que hay aparecido se pueden utilizar para resolver w para después encontrar el valor de b . Se utiliza normalmente un promedio de todas las soluciones que se han obtenido como puntos de soporte.

Support Vector Machines

Kernel no lineal

- En el problema dual sólo aparece dependencia con respecto al producto punto de los datos x_i .
- Un mapeo adecuado de los datos pueden resultar de pasar de algo no-lineal a algo lineal.
- Se realiza el modelo sobre el espacio transformado por un kernel.
- El requisito es que este kernel se encuentra definido por una matriz simétrica semi definida positiva (no negativos en la diagonal).

- Separador lineal

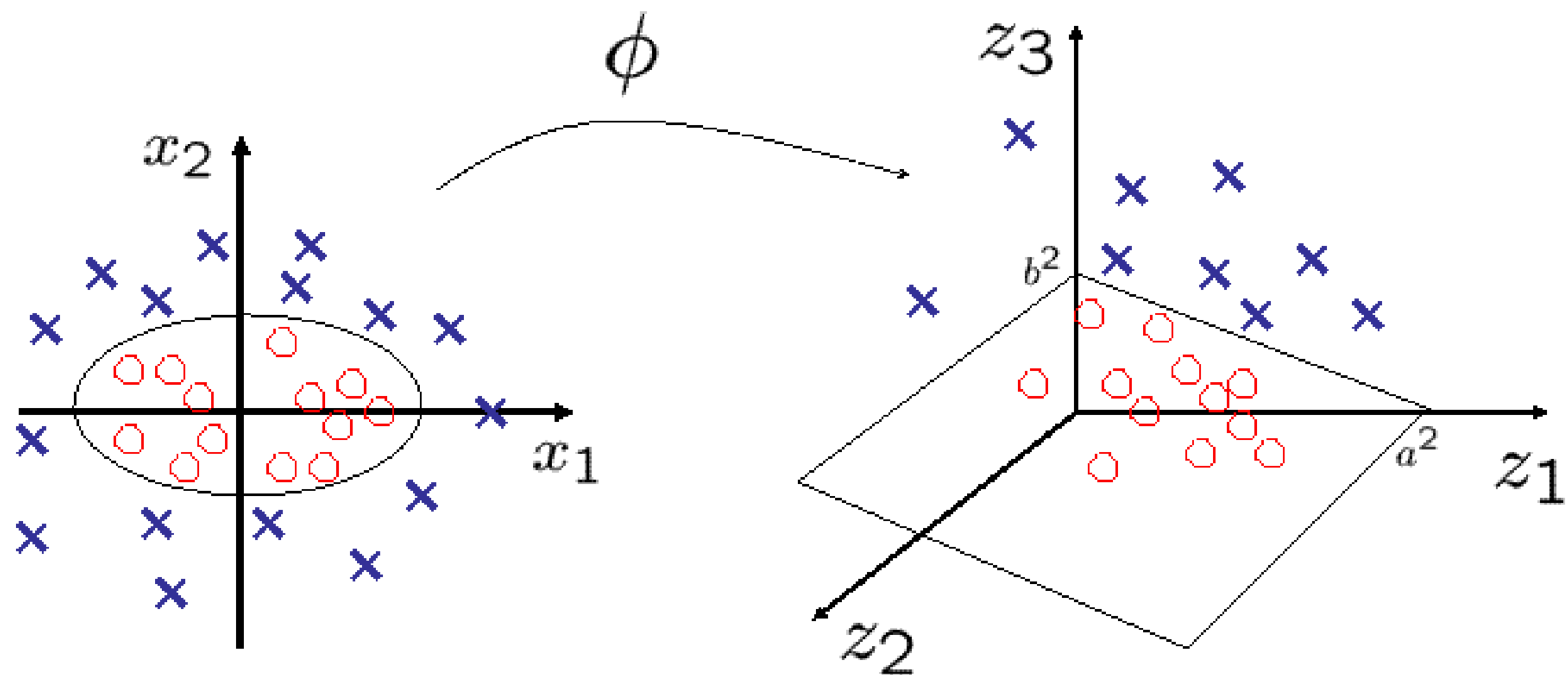
$$f(x) = x^T w + b = x^T \sum_{i=1}^n \alpha_i y_i x_i + b$$

$$= \sum_{i=1}^n \alpha_i y_i x^T x_i + b = \sum_{i=1}^n \alpha_i y_i \langle x, x_i \rangle + b$$

- Con cierta función $h_i(x)$

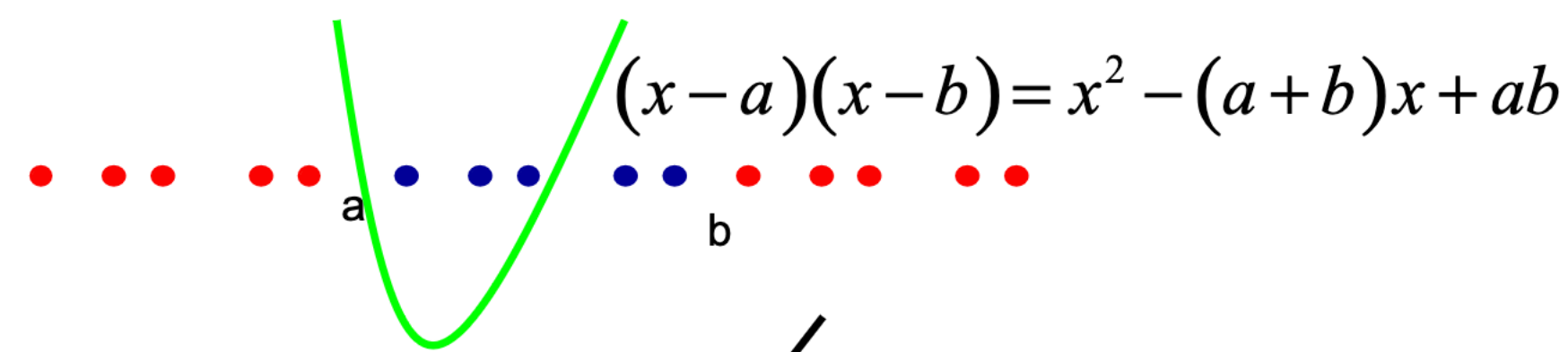
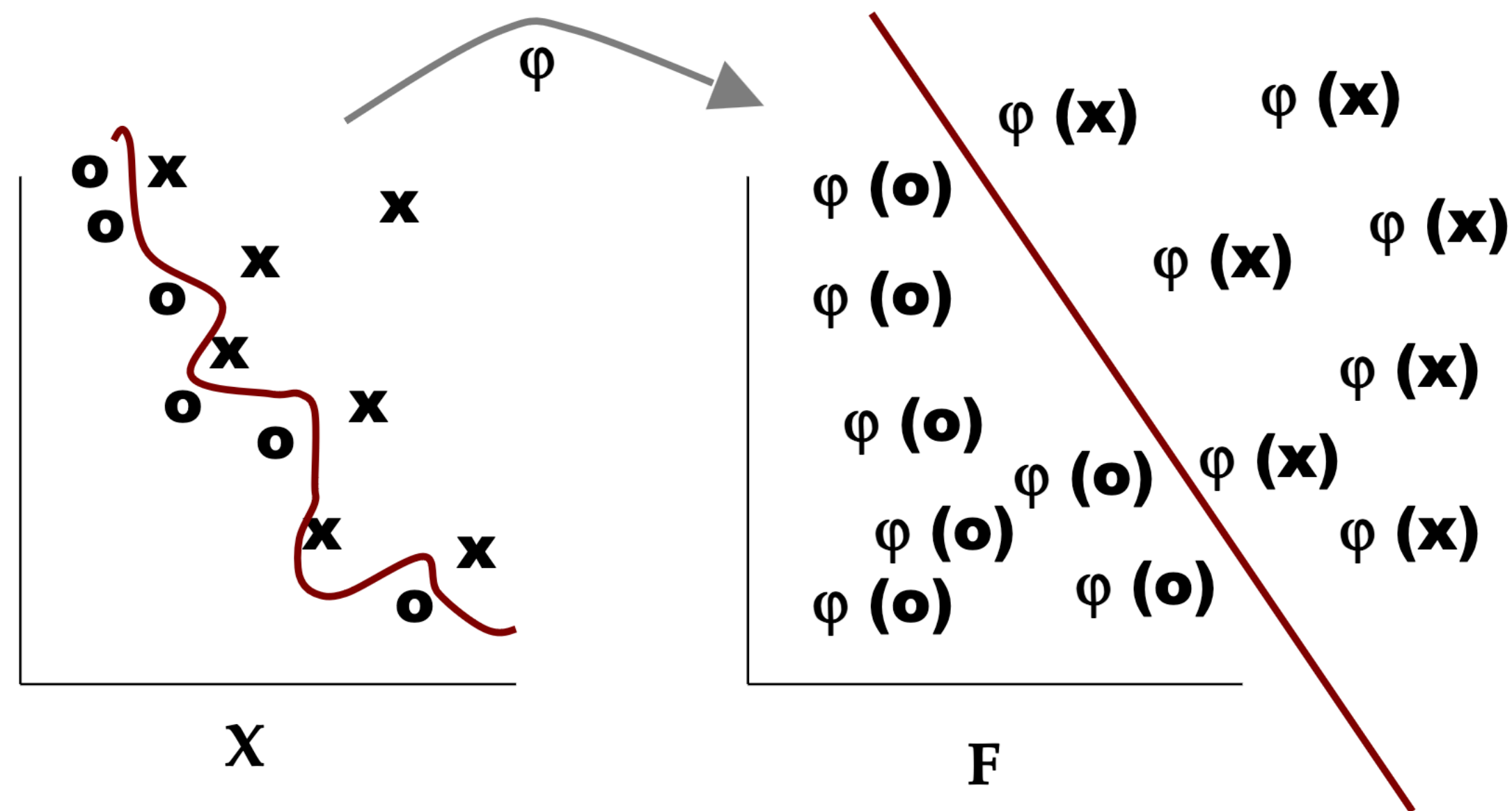
$$f(x) = h(x)^T w + b = h(x)^T \sum_{i=1}^n \alpha_i y_i x_i + b$$

$$= \sum_{i=1}^n \alpha_i y_i h(x)^T x_i + b = \sum_{i=1}^n \alpha_i y_i \langle h(x), h(x_i) \rangle + b$$

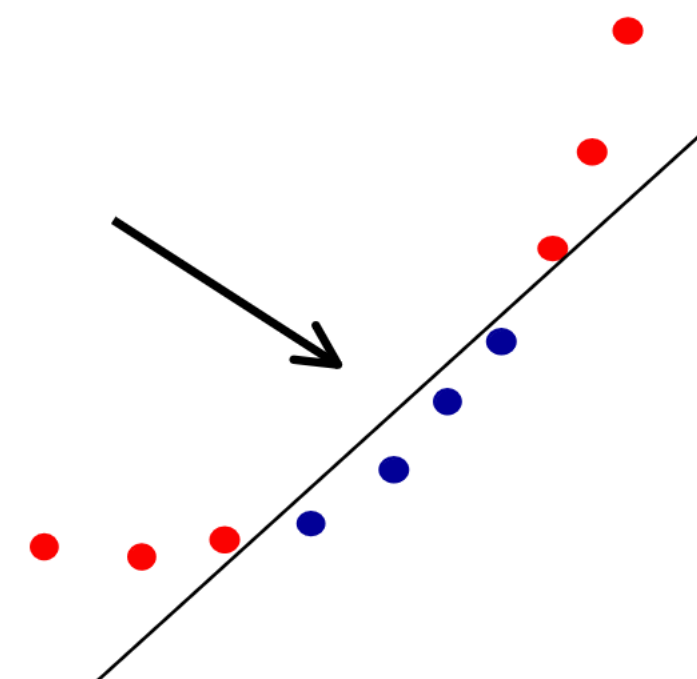


$$\phi : (x_1, x_2) \longrightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\left(\frac{x_1}{a}\right)^2 + \left(\frac{x_2}{b}\right)^2 = 1 \longrightarrow \frac{z_1}{a^2} + \frac{z_3}{b^2} = 1$$

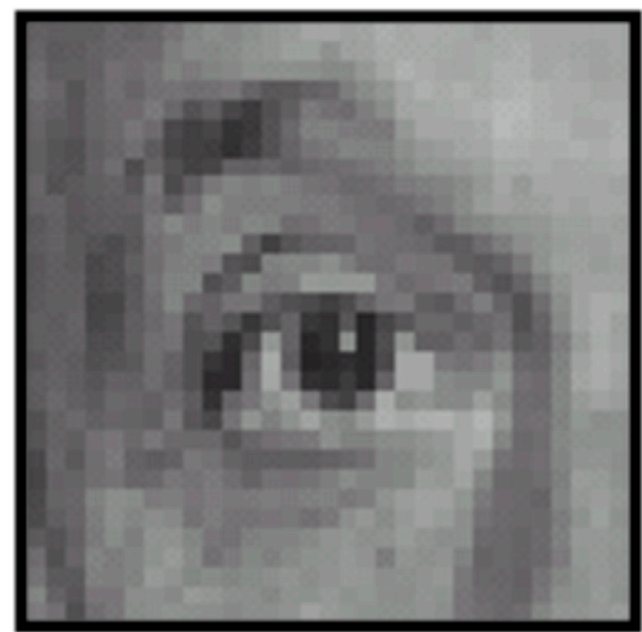


$$(x-a)(x-b) = x^2 - (a+b)x + ab$$



Support Vector Machines

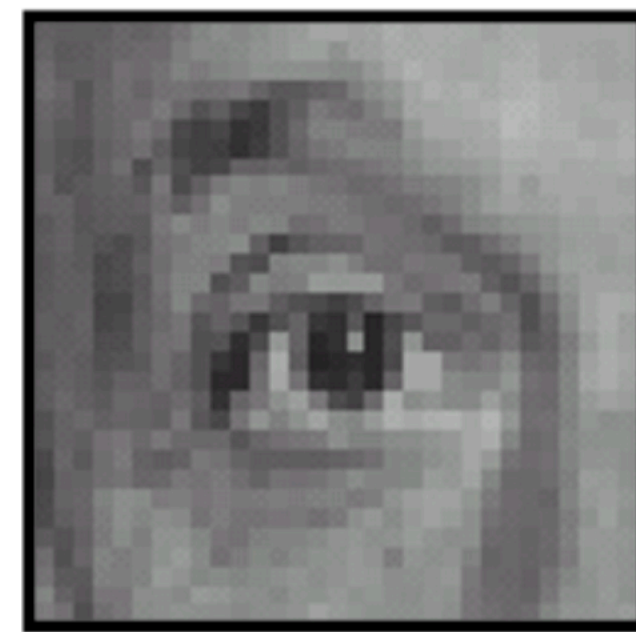
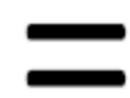
Kernel en imágenes



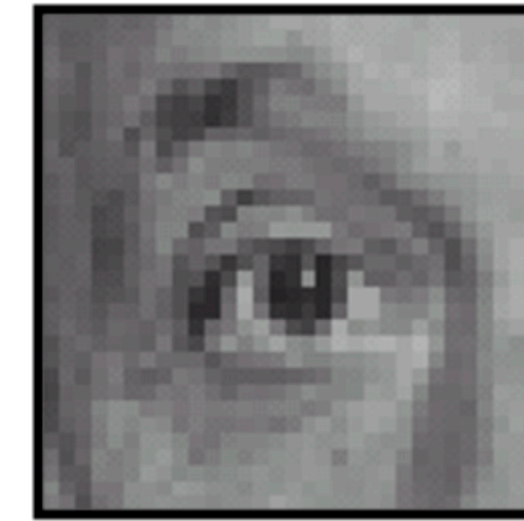
Original



0	0	0
0	1	0
0	0	0



Identical image

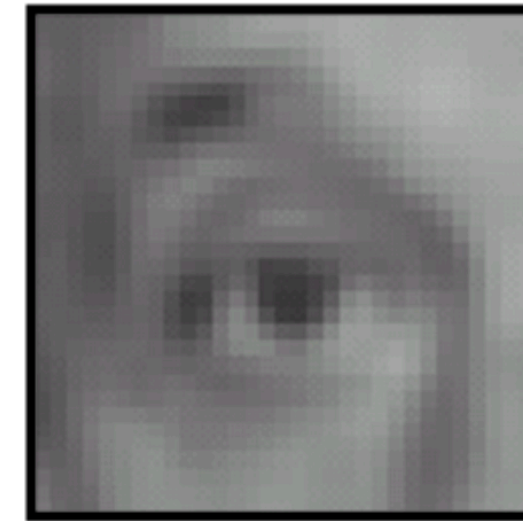


Original

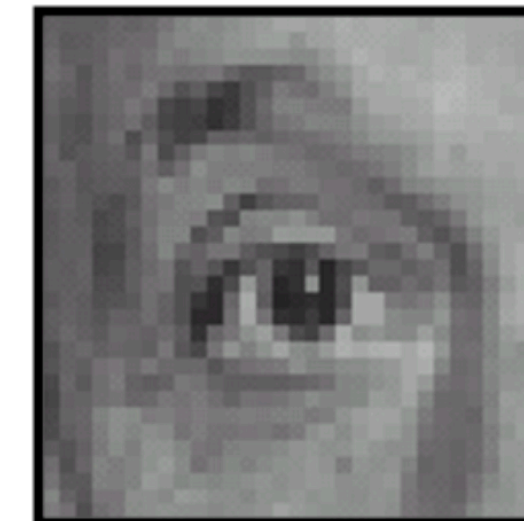


$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



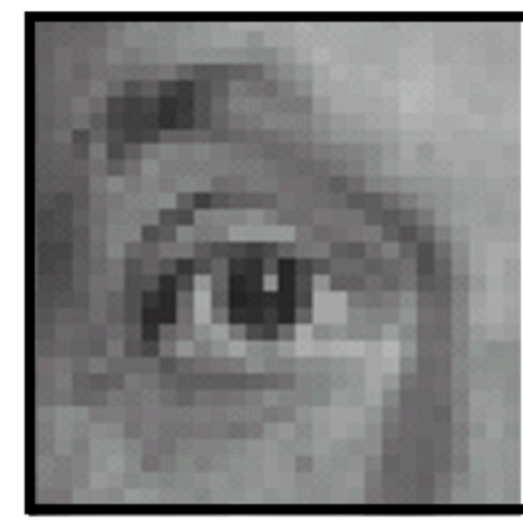
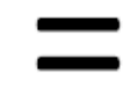
Blur (with a mean filter)



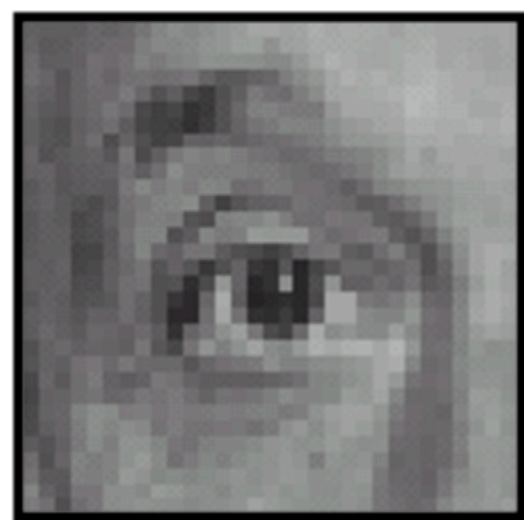
Original



0	0	0
1	0	0
0	0	0



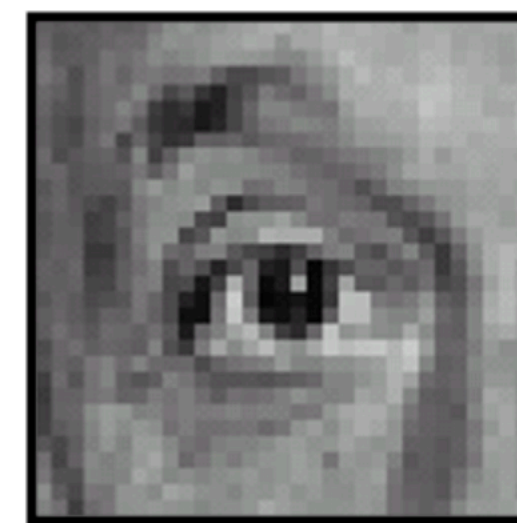
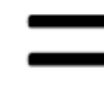
Shifted left
By 1 pixel



Original



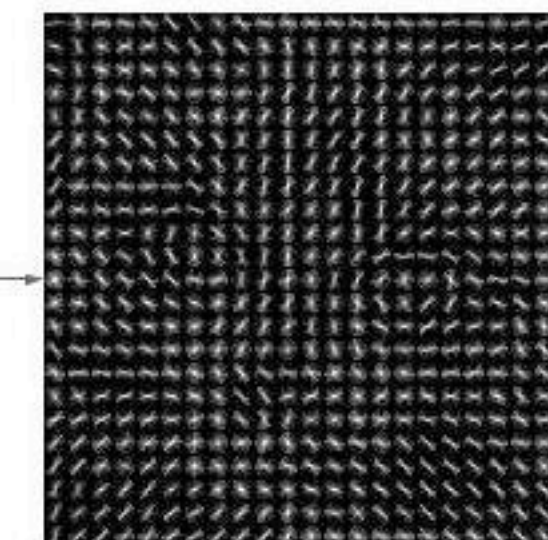
$$\left(\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \right)$$



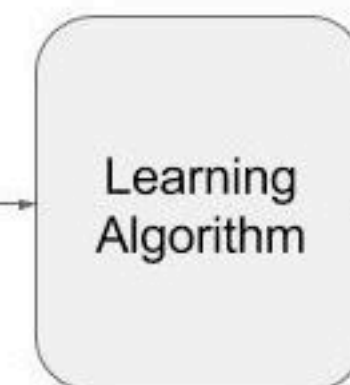
Sharpening filter
(accentuates edges)



Input image



Features
HAAR, HOG, SIFT,
SURF



Learning
Algorithm

SVM,
Adaboost,
ANN

CAT

Label
Assignment

Support Vector Machines

Kernel comunes

- **Kernel:** $K(x, y) = \langle h(x), h(y) \rangle$
- **Polinomial**

$$K(x, y) = (1 + \langle x, y \rangle)^d$$











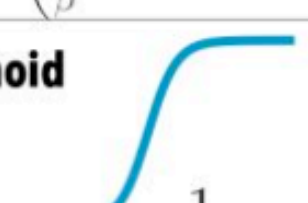

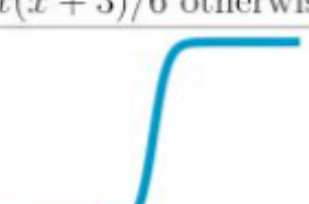
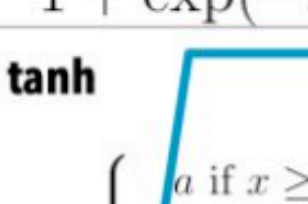
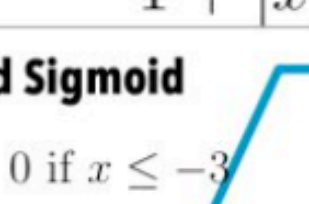

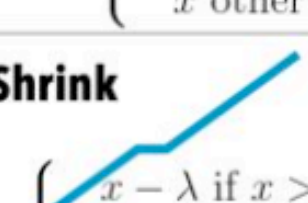
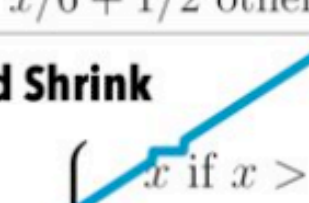
- **Radial**

$$K(x, y) = e^{-\gamma \|x-y\|^2}$$

- **Tanh**

$$K(x, y) = \tanh(\kappa_1 \langle x, y \rangle + \kappa_2)$$

Neural Network Activation Functions: a small subset!

ReLU  $\max(0, x)$	GELU  $\frac{x}{2} \left(1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + ax^3) \right) \right)$	PReLU  $\max(0, x)$
ELU  $\begin{cases} x & \text{if } x > 0 \\ \alpha(x \exp x - 1) & \text{if } x < 0 \end{cases}$	Swish  $\frac{x}{1 + \exp -x}$	SELU  $\alpha(\max(0, x) + \min(0, \beta(\exp x - 1)))$
SoftPlus  $\frac{1}{\beta} \log (1 + \exp(\beta x))$	Mish  $x \tanh \left(\frac{1}{\beta} \log (1 + \exp(\beta x)) \right)$	RReLU  $\begin{cases} x & \text{if } x \geq 0 \\ ax & \text{if } x < 0 \text{ with } a \sim \mathcal{R}(l, u) \end{cases}$
HardSwish  $\begin{cases} 0 & \text{if } x < -3 \\ x & \text{if } x \geq 3 \\ x(x+3)/6 & \text{otherwise} \end{cases}$	Sigmoid  $\frac{1}{1 + \exp(-x)}$	SoftSign  $\frac{x}{1 + x }$
Tanh  $\tanh(x)$	Hard tanh  $\begin{cases} a & \text{if } x \geq a \\ b & \text{if } x \leq b \\ x & \text{otherwise} \end{cases}$	Hard Sigmoid  $\begin{cases} 0 & \text{if } x \leq -3 \\ 1 & \text{if } x \geq 3 \\ x/6 + 1/2 & \text{otherwise} \end{cases}$
Tanh Shrink  $x - \tanh(x)$	Soft Shrink  $\begin{cases} x - \lambda & \text{if } x > \lambda \\ x + \lambda & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{cases}$	Hard Shrink  $\begin{cases} x & \text{if } x > \lambda \\ x & \text{if } x < -\lambda \\ 0 & \text{otherwise} \end{cases}$