IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

AUTOENCODERS

Auto-encoders

<u>Motivación</u>: Trabajar sobre una representación de baja dimensionalidad y con alta capacidad informativa para aprendizaje automático. Útil para *data augmentation*.

Los primeros autoencoders se construyeron a partir de redes neuronales entrenadas para reconstruir la entrada.

Formalmente, el problema corresponde a aprender dos funciones:

Encoder: $A : \mathbb{R}^n \longrightarrow \mathbb{R}^p$

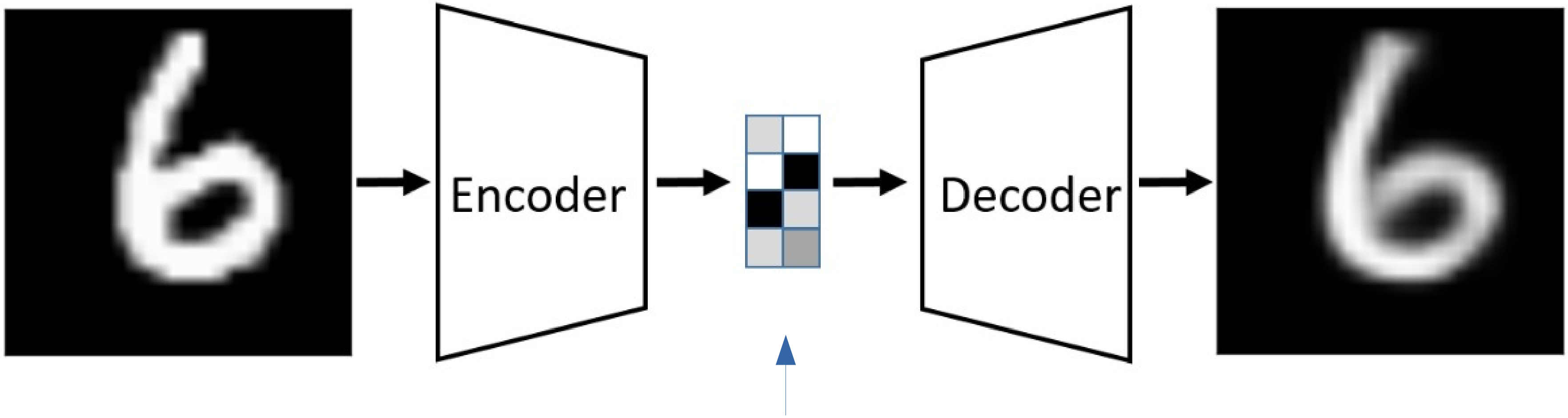
Decoder: $B : \mathbb{R}^p \longrightarrow \mathbb{R}^n$

de tal manera que: $\arg\min_{A,B} E[\Delta(\mathbf{x}, B \circ A(\mathbf{x})]$

Composición de funciones

Error de reconstrucción
(usualmente norma L_2)

Auto-encoders



Entrada

Representación compacta

Salida

- Si A y B implementan redes feed-forward lineales (activación lineal), el autoencoder se denomina autoencoder lineal.

- Un autoencoder es una generalización de PCA.

Auto-encoders

- Reducir la dimensionalidad es útil para evitar over-fitting ¿Por qué?

- Surge un tradeoff: Por un lado queremos que la arquitectura obtenga un error de reconstrucción muy bajo. Por otro, queremos que la representación compacta descarte información no esencial.

- Una forma de abordar el tradeoff consiste en introducir sparsity en las activaciones de las capas ocultas. Existen dos estrategias, regularización L_1 o uso de divergencia KL.

Requiere ajustar este parámetro

Sparse autoencoder con regularización L_1:

$$\arg\min_{A,B} E[\Delta(\mathbf{x}, B \circ A(\mathbf{x})] + \lambda \sum_i |a_i|$$

Activación de la *i*-th neurona oculta

Auto-encoders

Sparse autoencoder basada en divergencia KL: asumimos que la activación de cada neurona actúa como una variable Bernoulli con probabilidad *p*. Se controla el parámetro *p*.

Para cada batch, se estima la probabilidad y se calcula la diferencia, la cual es usada como regularizador.

Para cada neurona *j* la probabilidad empírica es:

$$\hat{p}_j = \frac{1}{m} \sum_i a_i(x)$$

Tamaño del batch

Activación del dato *i* en la neurona *j*

Luego, la función de pérdida es:

$$\arg\min_{A,B} E[\Delta(\mathbf{x}, B \circ A(\mathbf{x})] + \sum_j KL(p||\hat{p}_j)$$

Auto-encoders variacionales

El VAE es un modelo generativo (enfoque Bayesiano) que describe el proceso generativo de los datos.

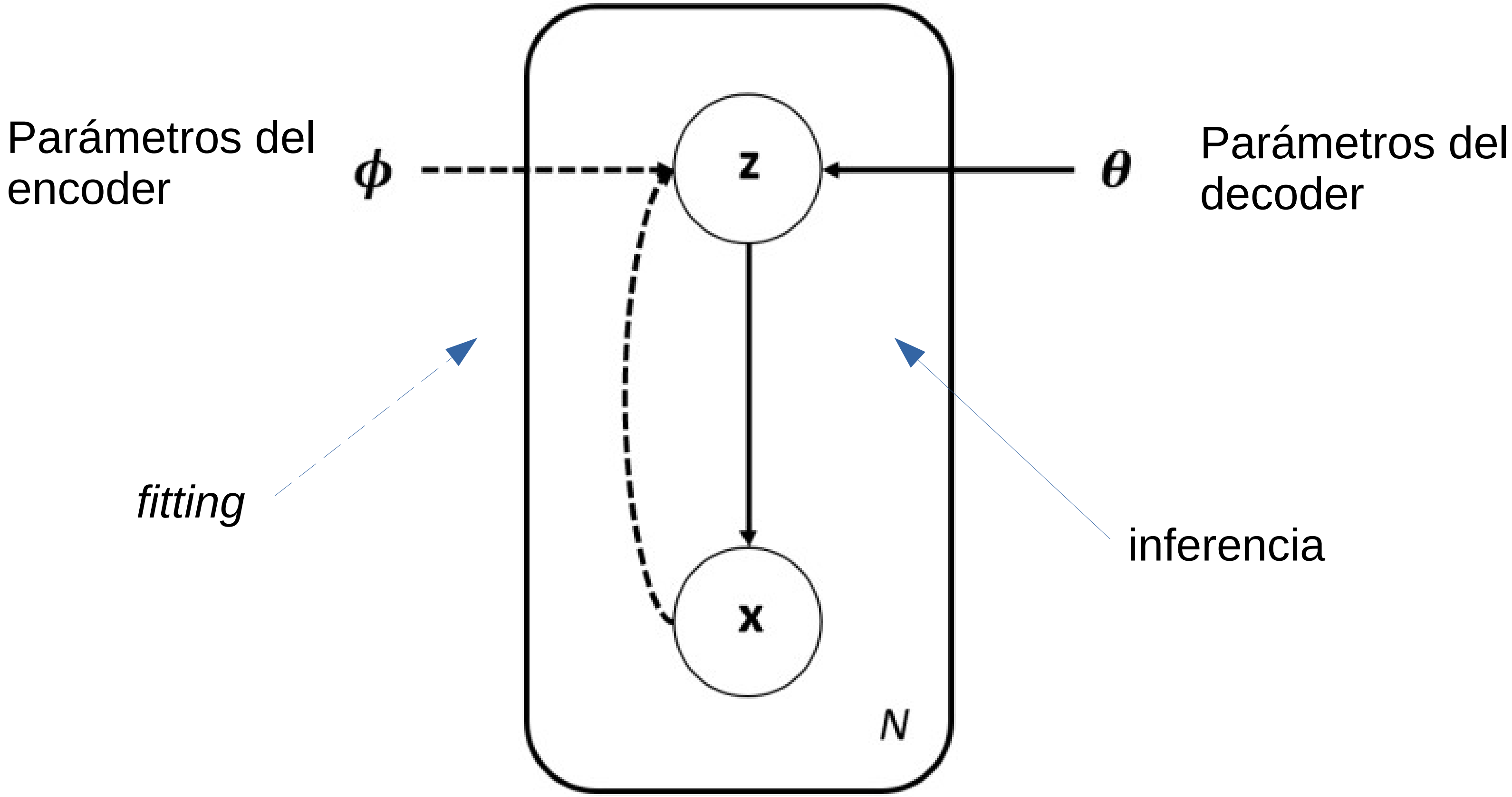
Dado un dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, el VAE asume la generación de cada dato condicionada a una variable latente aleatoria $\mathbf{z}_i$. Este modelo describe al decoder (probabilístico).

Análogamente, asumimos la existencia de una distribución a *posteriori* para generar las variables latentes a partir de los datos. Este modelo describe al encoder (probabilístico).

Las variables latentes tiene una distribución a priori denotada por $p_{\theta}(\mathbf{z}_i)$.

Diederik Kingma, Max Welling: Auto-Encoding Variational Bayes. ICLR 2014.

Auto-encoders variacionales



Auto-encoders variacionales

¿Qué hace el VAE?

Es un modelo de variable latente que usa redes neuronales (en específico perceptrón multicapa) para aproximar la *posterior* de

$q_\phi(z|x)$ y del modelo generativo $p_\theta(x, z)$.

Asumimos que la posterior aproximada es una Gaussiana multivariada. Los parámetros de esta distribución son calculados usando un MLP que toma los datos como entrada:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x)\mathbf{I})$$

Auto-encoders variacionales

¿Qué hace el VAE?

Es un modelo de variable latente que usa redes neuronales (en específico perceptrón multicapa) para aproximar la *posterior* de

$q_\phi(z|x)$ y del modelo generativo $p_\theta(x, z)$.

Asumimos que la posterior aproximada es una Gaussiana multivariada. Los parámetros de esta distribución son calculados usando un MLP que toma los datos como entrada:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x)\mathbf{I})$$

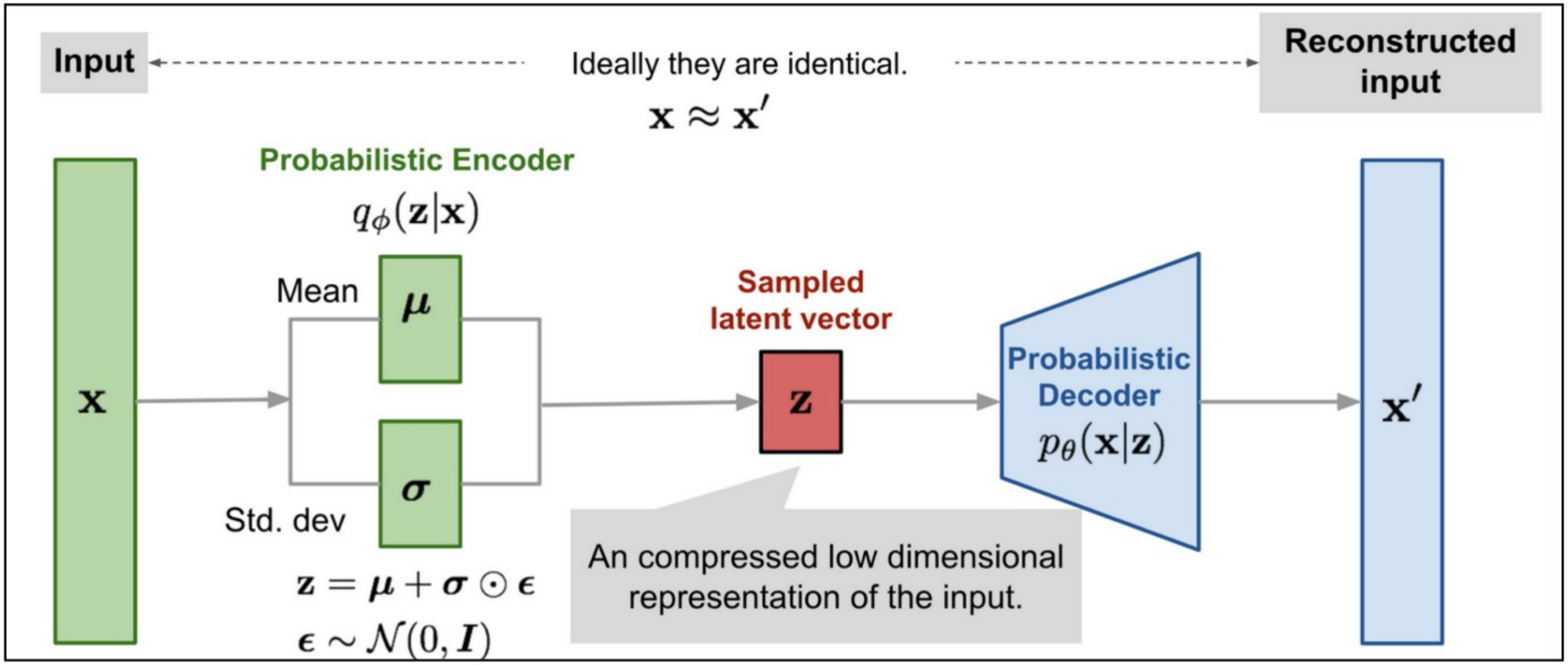
Para el *decoder*, se asume $p(z)$ fijo:

$$p(z) = \mathcal{N}(0, \mathbf{I})$$

El modelo generativo dependerá del tipo de datos con los que trabajamos. Por ejemplo:

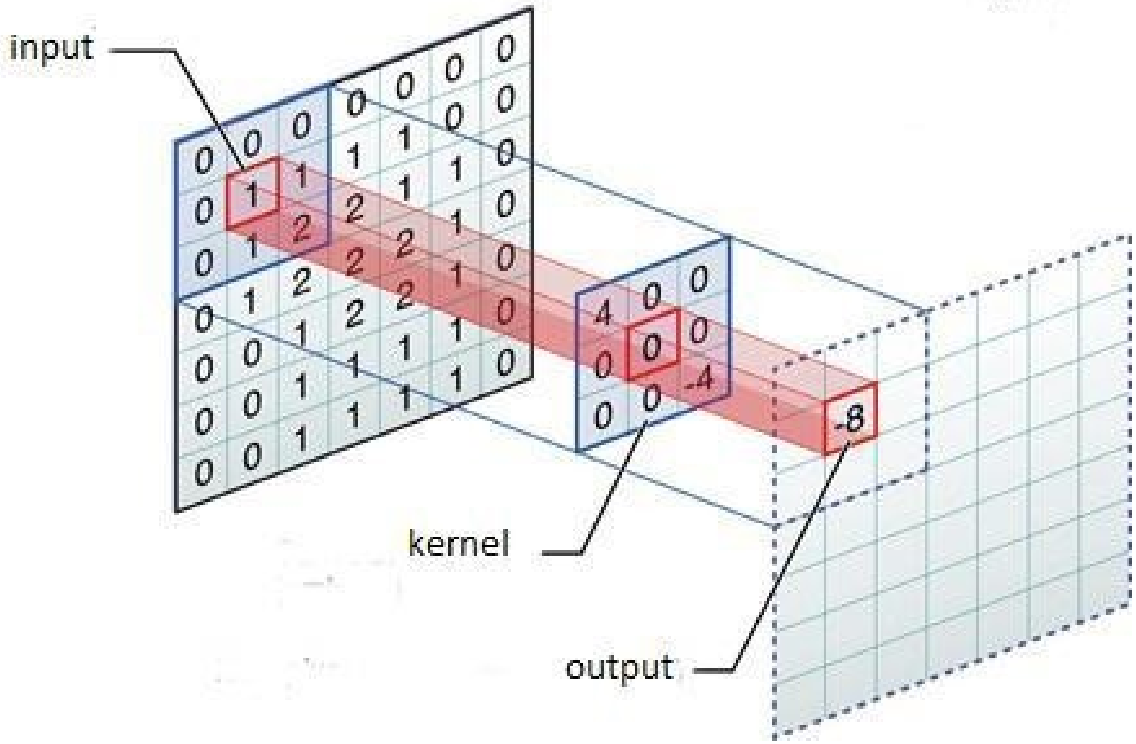
- Datos reales: Gaussiana multivariada
- Datos booleanos: Bernoulli

Auto-encoders variacionales



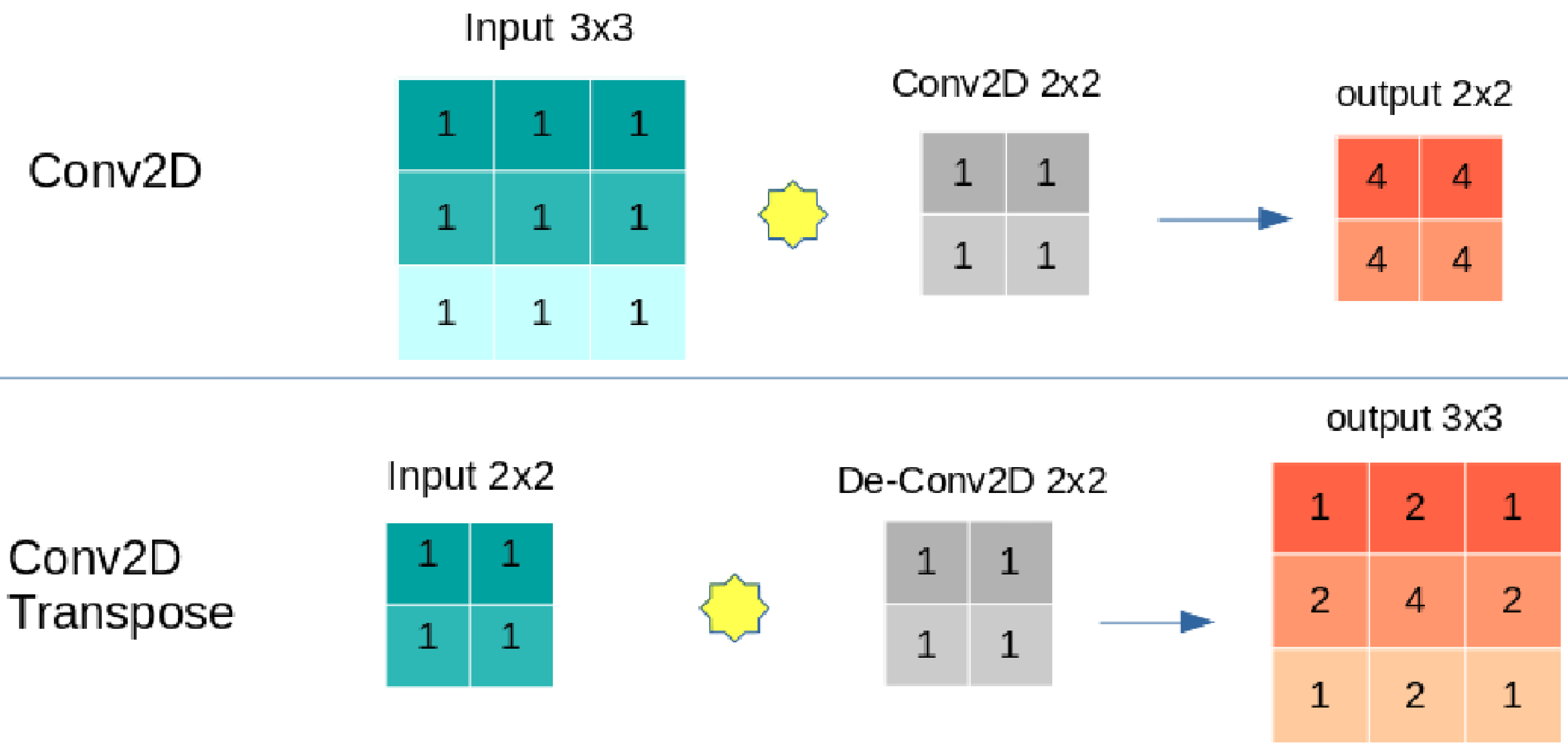
El VAE puede generar datos nuevos

Auto-encoders variacionales con imágenes



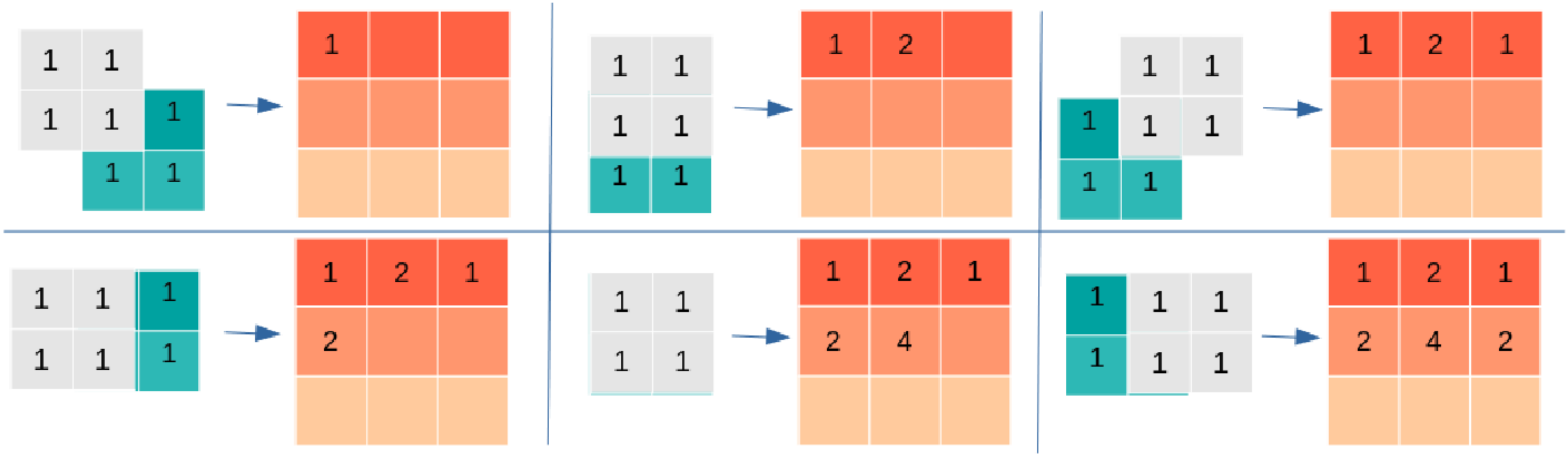
Auto-encoders variacionales con imágenes

Filtros convolucionales 2D

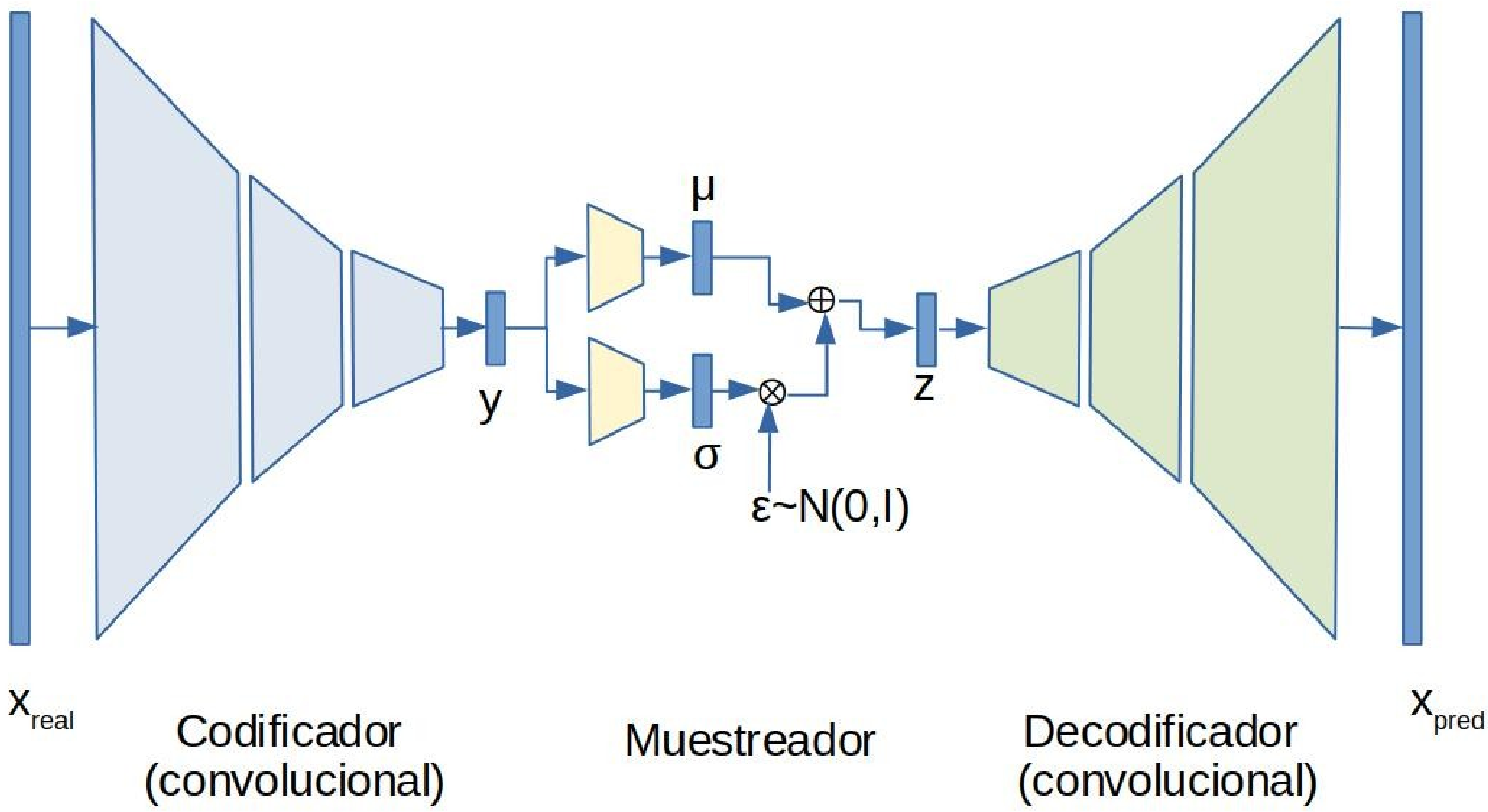


Auto-encoders variacionales con imágenes

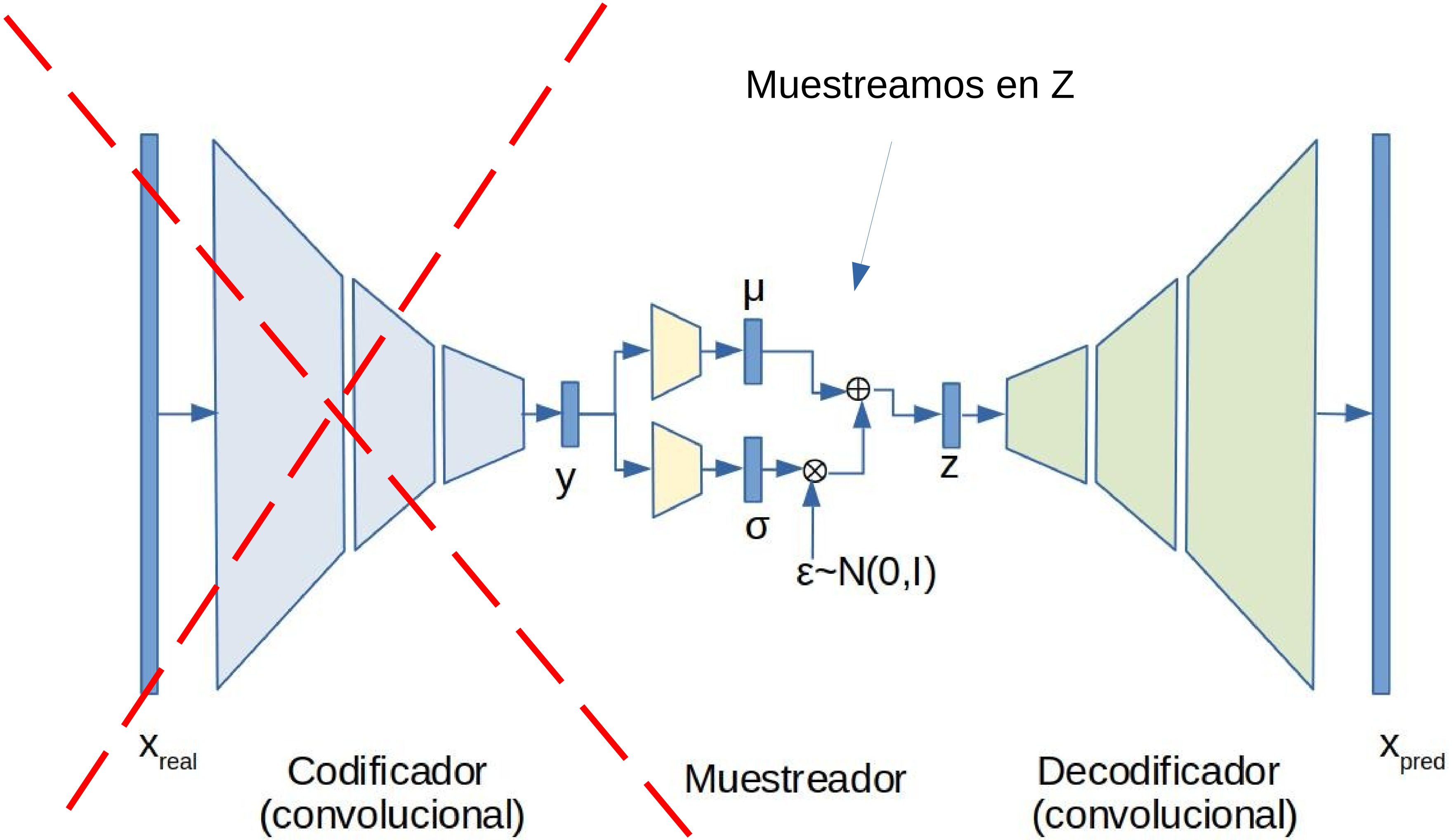
Filtros convolucionales 2D transpose



Auto-encoders variacionales con imágenes



Data augmentation en base al VAE preentrenado



- SUPPORT VECTOR MACHINES -

Hiperplano separador

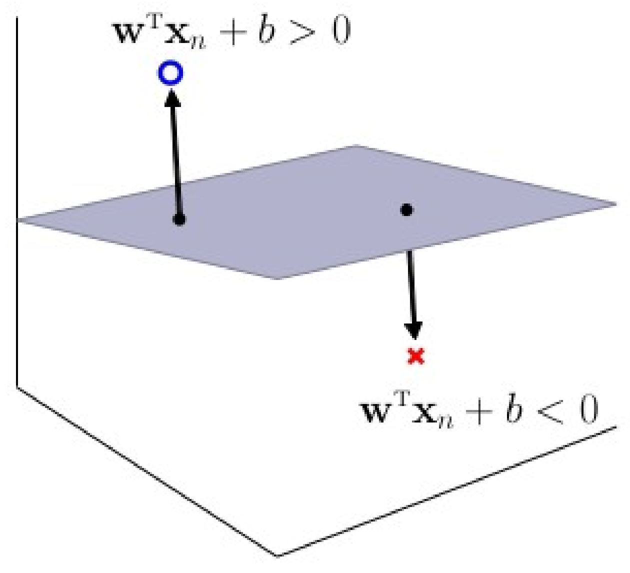
Queremos calcular el hiperplano más grueso que separa los datos. Es decir, queremos identificar el <u>hiperplano separador de máximo margen</u>.

El máximo margen es la distancia desde el hiperplano al dato más cercano.

Supongamos que tenemos un conjunto de datos que queremos separar del resto (son de la misma clase). El hiperplano separa estos ssi:

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) > 0$$

señal



Notar que la magnitud de la señal no es relevante para la decisión, dado que podemos **reescalar** los pesos y el bias.

Hiperplano separador

Encontramos **el dato más cercano**, y calculamos:

$$\rho = \min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$$

Y reescalamos con respecto a ρ:

$$\min_{n=1,\ldots,N} \; y_n\left(\frac{\mathbf{w}}{\rho}^{\mathrm{T}}\mathbf{x}_n + \frac{b}{\rho}\right) = \frac{1}{\rho}\min_{n=1,\ldots,N} \; y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = \frac{\rho}{\rho} = 1.$$

Es decir, es posible encontrar pesos tal que $y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$ es mayor o igual que 1 y al menos existe un dato que satisface la igualdad.

Hiperplano separador

Encontramos **el dato más cercano**, y calculamos:

$$\rho = \min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$$

Y reescalamos con respecto a ρ:

$$\min_{n=1,\ldots,N} y_n\left(\frac{\mathbf{w}}{\rho}^{\mathrm{T}}\mathbf{x}_n + \frac{b}{\rho}\right) = \frac{1}{\rho}\min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = \frac{\rho}{\rho} = 1.$$

Es decir, es posible encontrar pesos tal que $y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)$ es mayor o igual que 1 y al menos existe un dato que satisface la igualdad.

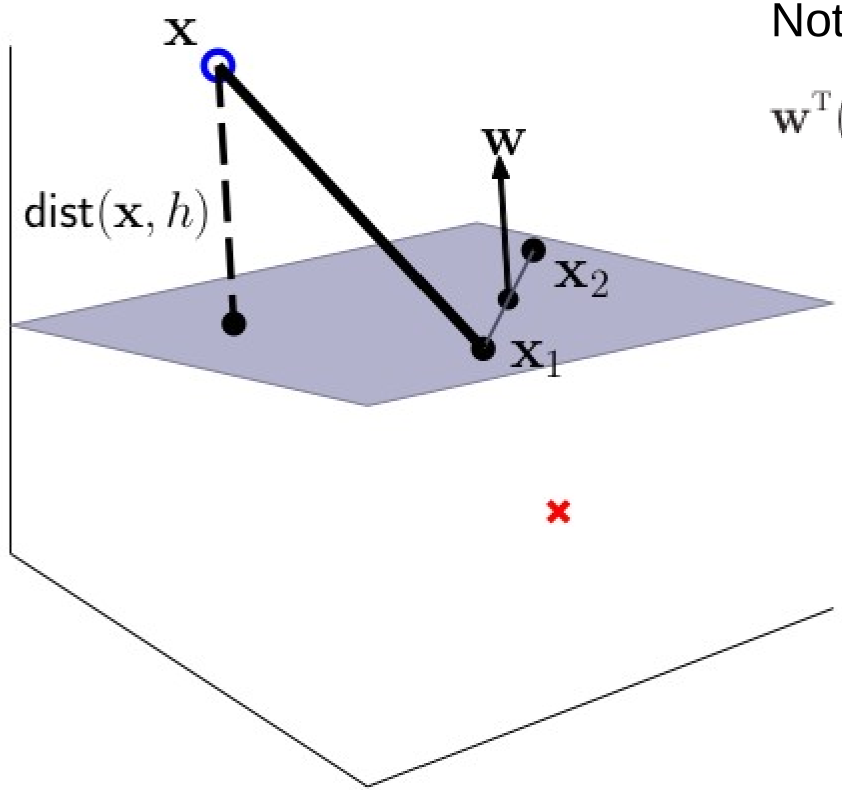
<u>Definición</u>. Un hiperplano separador (grueso) separa los datos si:

$$\min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1.$$

Básicamente, este es un esquema de normalización de pesos.

Margen del hiperplano separador

Margen: distancia desde el hiperplano al dato más cercano.



Notar que el vector de pesos cumple:

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_2 - \mathbf{x}_1) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_2 - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 = -b + b = 0.$$

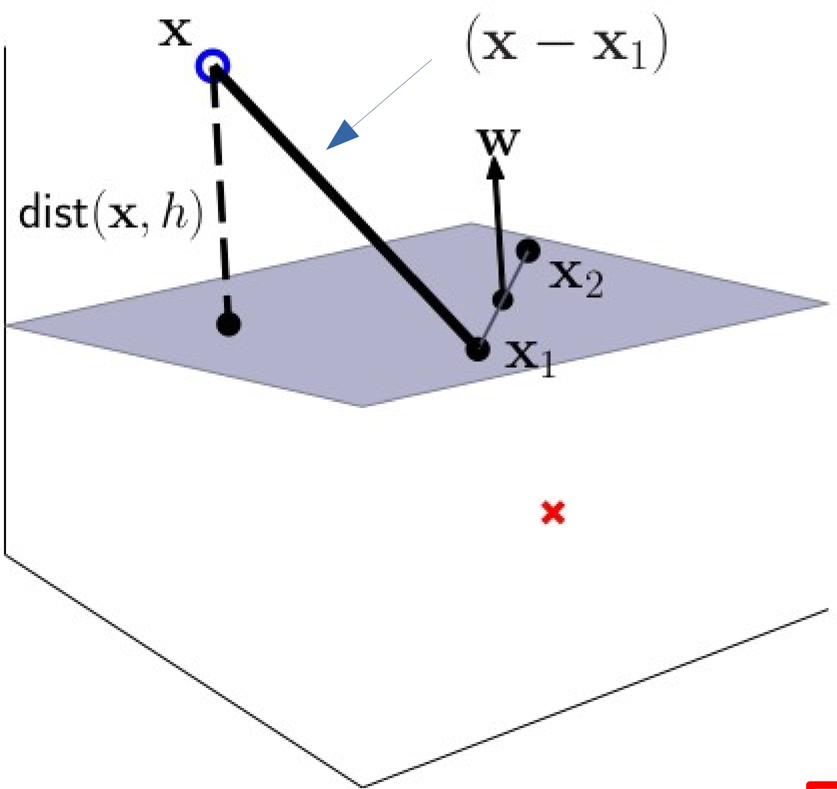
Esto porque para cualquier punto que este sobre el hiperplano se da que:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}' + b = 0.$$

Es decir, $\mathbf{w}$ es ortogonal al hiperplano.

Margen del hiperplano separador

Usando un vector <u>ortonormal</u> $\mathbf{u} = \mathbf{w}/\|\mathbf{w}\|$, obtenemos:



Proyección de la hipotenusa a la dirección del vector ortonormal

- *b*

$$\begin{aligned}\text{dist}(\mathbf{x}, h) &= |\mathbf{u}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}_1)| \\ &= \frac{1}{\|\mathbf{w}\|} \cdot |\mathbf{w}^{\mathrm{T}}\mathbf{x} - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1| \\ &= \frac{1}{\|\mathbf{w}\|} \cdot |\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|\end{aligned}$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}' + b = 0.$$

Entonces:

$$\boxed{\text{dist}(\mathbf{x}, h) = \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}}$$

Margen del hiperplano separador

$$\text{dist}(\mathbf{x}, h) = \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}$$

Como el hiperplano separa según el signo, es decir, $y_n = \pm 1$ se cumple que:

$\mathbf{x}_1, \ldots, \mathbf{x}_N$

$$|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

Datos separados

Margen del hiperplano separador

$$\text{dist}(\mathbf{x}, h) = \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}$$

Como el hiperplano separa según el signo, es decir, $y_n = \pm 1$ se cumple que:

$\mathbf{x}_1, \ldots, \mathbf{x}_N$

$$|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

Datos separados

La distancia la podemos reescribir según:

$$\text{dist}(\mathbf{x}_n, h) = \frac{y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}.$$

Margen del hiperplano separador

$$\text{dist}(\mathbf{x}, h) = \boxed{\frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}}$$

Como el hiperplano separa según el signo, es decir, $y_n = \pm 1$ se cumple que:

$\mathbf{x}_1, \ldots, \mathbf{x}_N$

$$|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

Datos separados

La distancia la podemos reescribir según:

$$\text{dist}(\mathbf{x}_n, h) = \frac{y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}.$$

El dato más cercano tiene distancia (recordar que $\min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1.$):

$$\min_n \text{dist}(\mathbf{x}_n, h) = \frac{1}{\|\mathbf{w}\|} \cdot \overset{1}{\cancel{\min_n y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}}$$

Margen del hiperplano separador

$$\text{dist}(\mathbf{x}, h) = \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x} + b|}{\|\mathbf{w}\|}$$

Como el hiperplano separa según el signo, es decir, $y_n = \pm 1$ se cumple que:

$$\mathbf{x}_1, \ldots, \mathbf{x}_N$$

$$|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b| = |y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)| = y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b),$$

Datos separados

La distancia la podemos reescribir según:

$$\text{dist}(\mathbf{x}_n, h) = \frac{y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}.$$

El dato más cercano tiene distancia (recordar que $\min\limits_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1.$):

$$\min_n \text{dist}(\mathbf{x}_n, h) = \frac{1}{\|\mathbf{w}\|} \cdot \underbrace{\min_n y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}_{1}$$

$$\min_n \text{dist}(\mathbf{x}_n, h) = \frac{1}{\|\mathbf{w}\|}$$

Máximo margen

Margen del hiperplano separador

Objetivo: $$\min_n \text{dist}(\mathbf{x}_n, h) = \frac{1}{\| \mathbf{w} \|}$$

Encontrar los pesos que maximizan la distancia al dato más cercano

¡Minimizar denominador!

Margen del hiperplano separador

Objetivo: $\min_n \text{dist}(\mathbf{x}_n, h) = \dfrac{1}{\|\mathbf{w}\|}$

Encontrar los pesos que maximizan la distancia al dato más cercano

¡Minimizar denominador!

$$\boxed{\begin{aligned} &\underset{b,\mathbf{w}}{\text{minimize}} && \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to:} && \min_{n=1,\ldots,N} y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) = 1. \end{aligned}}$$

Hiperplano separador de máximo margen

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \ \text{ for } n = 1, \ldots, N.$$

Hiperplano separador de máximo margen

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \ \text{ for } n = 1, \ldots, N.$$

Ejemplo:

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{aligned} -b &\geq 1 & (i) \\ -(2w_1 + 2w_2 + b) &\geq 1 & (ii) \\ 2w_1 + b &\geq 1 & (iii) \\ 3w_1 + b &\geq 1 & (iv) \end{aligned}$$

Hiperplano separador de máximo margen

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \text{ for } n = 1, \ldots, N.$$

Ejemplo:

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{aligned} -b &\geq 1 \quad (i) \\ -(2w_1 + 2w_2 + b) &\geq 1 \quad (ii) \\ 2w_1 + b &\geq 1 \quad (iii) \\ 3w_1 + b &\geq 1 \quad (iv) \end{aligned}$$

Se resuelve usando (i), (ii) y (iii):

$$\left.\begin{array}{l} \text{- (i) y (iii) dan: } w_1 \geq 1 \\ \\ \text{- (ii) y (iii) dan: } w_2 \leq -1 \end{array}\right\} \text{ Tomamos el borde: } (b = -1, w_1 = 1, w_2 = -1)$$

Hiperplano separador de máximo margen

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \text{ for } n = 1, \ldots, N.$$

Ejemplo:

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{aligned} -b &\geq 1 \quad (i) \\ -(2w_1 + 2w_2 + b) &\geq 1 \quad (ii) \\ 2w_1 + b &\geq 1 \quad (iii) \\ 3w_1 + b &\geq 1 \quad (iv) \end{aligned}$$

Se resuelve usando (i), (ii) y (iii):

- (i) y (ii) dan: $w_1 \geq 1$
- (ii) y (iii) dan: $w_2 \leq -1$

Tomamos el borde: $(b = -1, w_1 = 1, w_2 = -1)$

$$g(\mathbf{x}) = \text{sign}(x_1 - x_2 - 1)$$

y

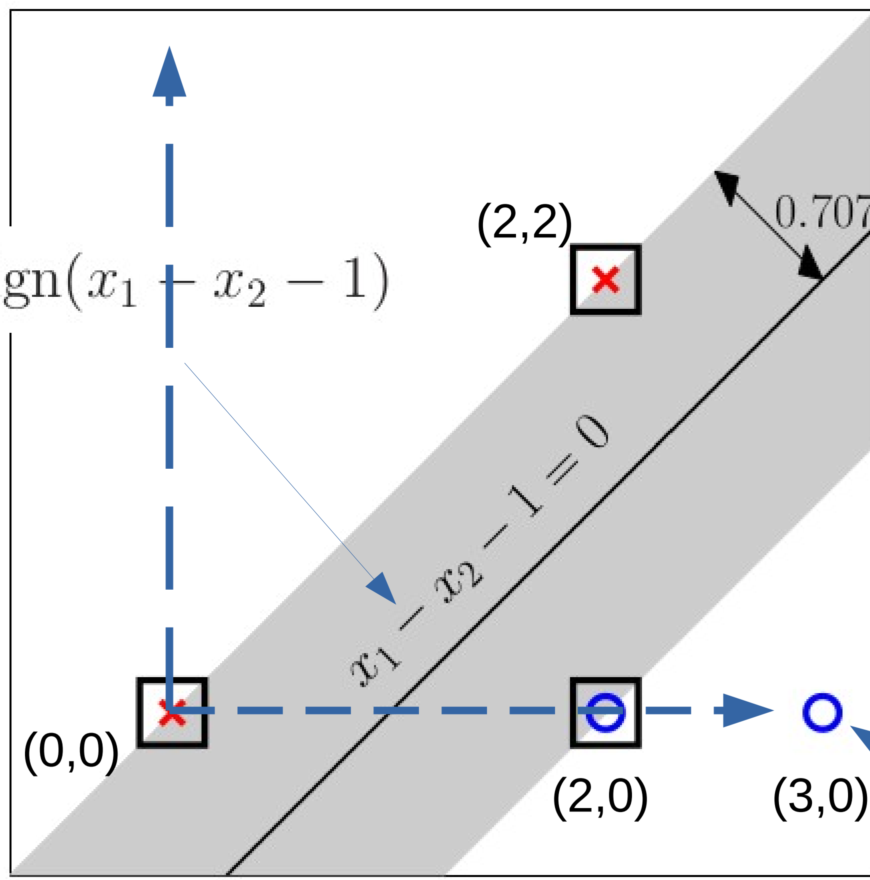
$$\frac{1}{\|\mathbf{w}^*\|} = \frac{1}{\sqrt{2}} \approx 0.707.$$

Hiperplano separador de máximo margen $y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$\begin{aligned} -b &\geq 1 & (i) \\ -(2w_1 + 2w_2 + b) &\geq 1 & (ii) \\ 2w_1 + b &\geq 1 & (iii) \\ 3w_1 + b &\geq 1 & (iv) \end{aligned}$$

$$g(\mathbf{x}) = \mathrm{sign}(x_1 - x_2 - 1)$$



$$\frac{1}{\|\mathbf{w}^*\|} = \frac{1}{\sqrt{2}} \approx 0.707.$$

$$y_n(\mathbf{w}^{*\mathrm{T}}\mathbf{x}_n + b^*) = 1$$

$\uparrow$

Support Vectors

Este dato no lo usamos!

Support Vector Machines (forma matricial)

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \ \text{ for } n = 1, \ldots, N.$$

Expresión matricial

$$\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} = \begin{bmatrix} b & \mathbf{w}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \begin{bmatrix} b \\ \mathbf{w}^{\mathrm{T}} \end{bmatrix} = \mathbf{u}^{\mathrm{T}} \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix} \mathbf{u} \implies \mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \ \mathbf{p} = \mathbf{0}_{d+1}$$

$$y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \equiv \begin{bmatrix} y_n & y_n\mathbf{x}_n^{\mathrm{T}} \end{bmatrix} \mathbf{u} \geq 1 \implies \begin{bmatrix} y_1 & y_1\mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^{\mathrm{T}} \end{bmatrix} \mathbf{u} \geq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \implies \mathrm{A} = \begin{bmatrix} y_1 & y_1\mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^{\mathrm{T}} \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Se reduce a un problema de optimización cuadrática para encontrar **Q** y **P**.

Support Vector Machines

Forma estándar de problema QP:

$$\begin{array}{ll} \underset{\mathbf{u}\in\mathbb{R}^L}{\text{minimize:}} & \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} \\ \text{subject to:} & \mathrm{A}\mathbf{u} \geq \mathbf{c}. \end{array}$$

donde:

$$\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^{\mathrm{T}} \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}, \ \mathbf{p} = \mathbf{0}_{d+1} \qquad \mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$\mathrm{A} = \begin{bmatrix} y_1 & y_1\mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^{\mathrm{T}} \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

ANEXO

- SUPPORT VECTOR MACHINES -
(KERNEL TRICK)

SVM en datos no separables (linealmente)

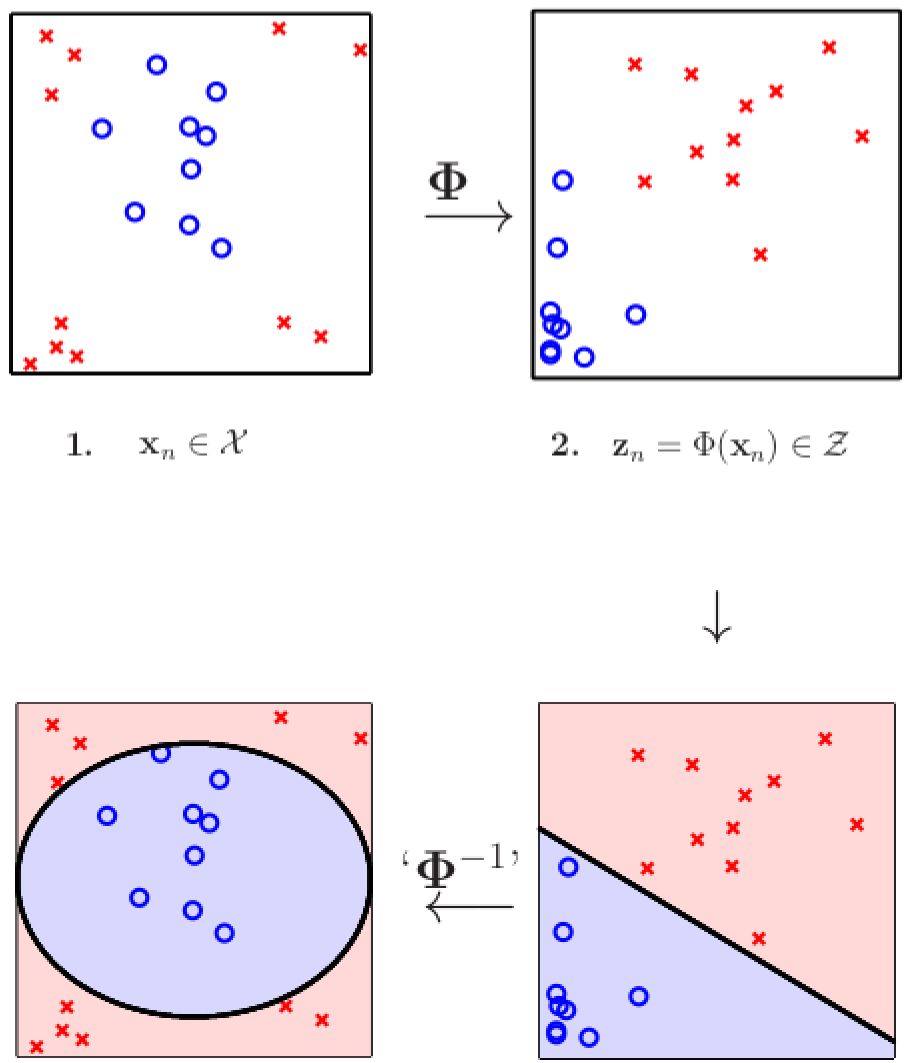
Consideremos una transformación $\Phi : \mathbb{R}^d \to \mathbb{R}^{\tilde{d}}$ tal que $\mathbf{z}_n = \boldsymbol{\Phi}(\mathbf{x}_n)$.
Después de transformar los datos, el problema SVM (hard margin) es:

$$\begin{array}{ll} \underset{\tilde{b},\tilde{\mathbf{w}}}{\text{minimize:}} & \frac{1}{2}\tilde{\mathbf{w}}^{\mathrm{T}}\tilde{\mathbf{w}} \\ \text{subject to:} & y_n\left(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z}_n + \tilde{b}\right) \geq 1 \qquad (n = 1, \cdots, N), \end{array}$$

donde $\tilde{\mathbf{w}}$ reside en el espacio de representación *Z*. Notemos que la dimensionalidad de Z puede ser distinta a la de la entrada.

Si la transformación es no lineal, puede ayudar a la SVM a separar datos no separables en el espacio original.

SVM en datos no separables (linealmente)



1. $\mathbf{x}_n \in \mathcal{X}$

2. $\mathbf{z}_n = \Phi(\mathbf{x}_n) \in \mathcal{Z}$

3. $\tilde{g}(\mathbf{z}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{z})$

4. $g(\mathbf{x}) = \tilde{g}(\Phi(\mathbf{x})) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\Phi(\mathbf{x}))$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \qquad \mathbf{z} = \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \vdots \\ \Phi_{\tilde{d}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_{\tilde{d}} \end{bmatrix}$$

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ $\qquad$ $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$

$y_1, y_2, \ldots, y_N$ $\qquad$ $y_1, y_2, \ldots, y_N$

$$\tilde{\mathbf{w}} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{d}} \end{bmatrix}$$

$$g(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^{\mathrm{T}}\mathbf{\Phi}(\mathbf{x}))$$

Este problema es difícil de resolver cuando $\tilde{d}$ es muy grande.

SVM en datos no separables (linealmente)

En la práctica, en lugar de trabajar sobre el problema original en el espacio *Z*, se aborda la formulación dual ya que es más fácil desde el punto de vista de optimización.

Teorema (KKT). El problema QP convexo en su forma primal:

$$\begin{aligned} \underset{\mathbf{u}\in\mathbb{R}^L}{\text{minimize:}} \quad & \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} \\ \text{subject to:} \quad & \mathbf{a}_m^{\mathrm{T}}\mathbf{u} \geq c_m \end{aligned}$$

define la función dual (Lagrange):

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{u} + \sum_{m=1}^{M} \alpha_m \left(c_m - \mathbf{a}_m^{\mathrm{T}}\mathbf{u}\right).$$

La solución del primal es óptima ssi es óptima en el dual. Luego:

$$\max_{\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}).$$

$$\max_{\boldsymbol{\alpha}\geq \mathbf{0}} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u},\boldsymbol{\alpha}).$$

$$\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$$

SVM en datos no separables (linealmente)

Apliquemos la formulación dual a la SVM:

$$\begin{aligned}\mathcal{L}(b,\mathbf{w},\boldsymbol{\alpha}) &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N}\alpha_n\left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N}\alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n - b\sum_{n=1}^{N}\alpha_n y_n + \sum_{n=1}^{N}\alpha_n\end{aligned}$$

$$\max_{\boldsymbol{\alpha}\geq\mathbf{0}} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}).$$

$$\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$$

SVM en datos no separables (linealmente)

Apliquemos la formulación dual a la SVM:

$$\begin{aligned}\mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)\right) \\ &= \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n - b\sum_{n=1}^{N} \alpha_n y_n + \sum_{n=1}^{N} \alpha_n\end{aligned}$$

Minimizamos w.r.t. $(b, \mathbf{w})$

$\frac{\partial \mathcal{L}}{\partial b} = 0$:

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{n=1}^{N} \alpha_n y_n \qquad \Longrightarrow \qquad \sum_{n=1}^{N} \alpha_n y_n = 0$$

$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

SVM en datos no separables (linealmente)

Usaremos $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ en $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n - b\cancelto{0}{\sum_{n=1}^{N} \alpha_n y_n} + \sum_{n=1}^{N} \alpha_n \\
&= -\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \\
&= -\frac{1}{2}\sum_{m,n=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m + \sum_{n=1}^{N} \alpha_n
\end{aligned}
$$

SVM en datos no separables (linealmente)

Usaremos $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ en $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{w}^{\mathrm{T}}\mathbf{x}_n + \sum_{n=1}^{N} \alpha_n.$

0

$$
\begin{aligned}
\mathcal{L} &= \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n - b\cancel{\sum_{n=1}^{N} \alpha_n y_n} + \sum_{n=1}^{N} \alpha_n \\
&= -\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \alpha_n \\
&= -\frac{1}{2}\sum_{m,n=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m + \sum_{n=1}^{N} \alpha_n
\end{aligned}
$$

Ojo que:

$$\max_{\boldsymbol{\alpha}\geq \mathbf{0}} \min_{\mathbf{u}} \mathcal{L}(\mathbf{u}, \boldsymbol{\alpha}).$$

es lo mismo que

Es decir, debemos hacer lo siguiente:

$$
\boxed{
\begin{array}{ll}
\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimize}}: & \displaystyle \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m - \sum_{n=1}^{N} \alpha_n \\
\text{subject to:} & \displaystyle \sum_{n=1}^{N} y_n \alpha_n = 0 \\
& \alpha_n \geq 0 \qquad\qquad (n = 1, \cdots, N).
\end{array}
}
$$

SVM en datos no separables (linealmente)

El dual se puede reescribir en una forma más amable:

$$\begin{aligned}
\underset{\boldsymbol{\alpha}\in\mathbb{R}^N}{\text{minimize}}: \quad & \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m - \sum_{n=1}^{N} \alpha_n \\
\text{subject to:} \quad & \sum_{n=1}^{N} y_n \alpha_n = 0 \\
& \alpha_n \geq 0 \qquad\qquad (n = 1, \cdots, N).
\end{aligned}$$

$$\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathrm{G}\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \\
\text{subject to:} \quad & \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0 \\
& \boldsymbol{\alpha} \geq \mathbf{0}
\end{aligned}$$

donde $(\mathrm{G}_{nm} = y_n y_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m)$

SVM en datos no separables (linealmente)

Podemos recuperar la SVM del primal desde el dual:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n^* y_n \mathbf{x}_n$$

$$\begin{aligned} \alpha_s > 0 \quad & \Longrightarrow \quad y_s(\mathbf{w}^{\mathrm{T}}\mathbf{x}_s + b) - 1 = 0 \\ & \Longrightarrow \quad b = y_s - \mathbf{w}^{\mathrm{T}}\mathbf{x}_s \end{aligned}$$

SVM en datos no separables (linealmente)

Podemos recuperar la SVM del primal desde el dual:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n^* y_n \mathbf{x}_n$$

$$\begin{aligned} \alpha_s > 0 \quad & \Longrightarrow \quad y_s(\mathbf{w}^{\mathrm{T}}\mathbf{x}_s + b) - 1 = 0 \\ & \Longrightarrow \quad b = y_s - \mathbf{w}^{\mathrm{T}}\mathbf{x}_s \end{aligned}$$

$$(\mathrm{G}_{nm} = y_n y_m \mathbf{x}_n^{\mathrm{T}} \mathbf{x}_m)$$

Una forma simple de calcular G:

$$\mathrm{X} = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \quad \longrightarrow \quad \mathrm{X}_s = \begin{bmatrix} 0 & 0 \\ -2 & -2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad \longrightarrow \quad \overset{\substack{\text{signed data matrix} \\ \downarrow}}{\mathrm{G} = \mathrm{X}_s \mathrm{X}_s^{\mathrm{T}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 8 & -4 & -6 \\ 0 & -4 & 4 & 6 \\ 0 & -6 & 6 & 9 \end{bmatrix}$$

SVM en datos no separables (linealmente)

$\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}$

<u>QP problem</u>

$$\begin{array}{ll} \underset{\mathbf{u}}{\text{minimize}} & \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathrm{Q}\mathbf{u} + \mathbf{p}^{\mathrm{T}}\mathbf{z} \\ \text{subject to:} & \mathrm{A}\mathbf{u} \geq \mathbf{c} \end{array}$$

primal desde el dual

<u>Dual SVM</u>

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\text{minimize}} & \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathrm{G}\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{subject to:} & \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0 \\ & \boldsymbol{\alpha} \geq \mathbf{0} \end{array}$$

Resuelvo para u

$$\boxed{\mathbf{u} = \boldsymbol{\alpha}}$$

$$\mathrm{Q} = \mathrm{G}$$

$$\mathbf{p} = -\mathbf{1}_N$$

$$\mathrm{A} = \begin{bmatrix} \mathbf{y}^{\mathrm{T}} \\ -\mathbf{y}^{\mathrm{T}} \\ \mathrm{I}_N \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{0}_N \end{bmatrix}$$

$$\xrightarrow{\mathrm{QP}(\mathrm{Q},\mathbf{p},\mathrm{A},\mathbf{c})}$$

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{w} = \sum_{n=1}^{4} \alpha_n^* y_n \mathbf{x}_n = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b = y_1 - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 = -1$$

$$\gamma = \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$$

SVM en datos no separables (linealmente)

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n^* y_n \mathbf{x}_n$$

$$\begin{aligned} \alpha_s > 0 \quad &\Longrightarrow \quad y_s(\mathbf{w}^{\mathrm{T}}\mathbf{x}_s + b) - 1 = 0 \\ &\Longrightarrow \quad b = y_s - \mathbf{w}^{\mathrm{T}}\mathbf{x}_s \end{aligned}$$

QP

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{w} = \sum_{n=1}^{4} \alpha_n^* y_n \mathbf{x}_n = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b = y_1 - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 = -1$$

$$\gamma = \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$$

SVM en datos no separables (linealmente)

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n^* y_n \mathbf{x}_n$$

$$\alpha_s > 0 \implies y_s(\mathbf{w}^{\mathrm{T}}\mathbf{x}_s + b) - 1 = 0$$
$$\implies b = y_s - \mathbf{w}^{\mathrm{T}}\mathbf{x}_s$$

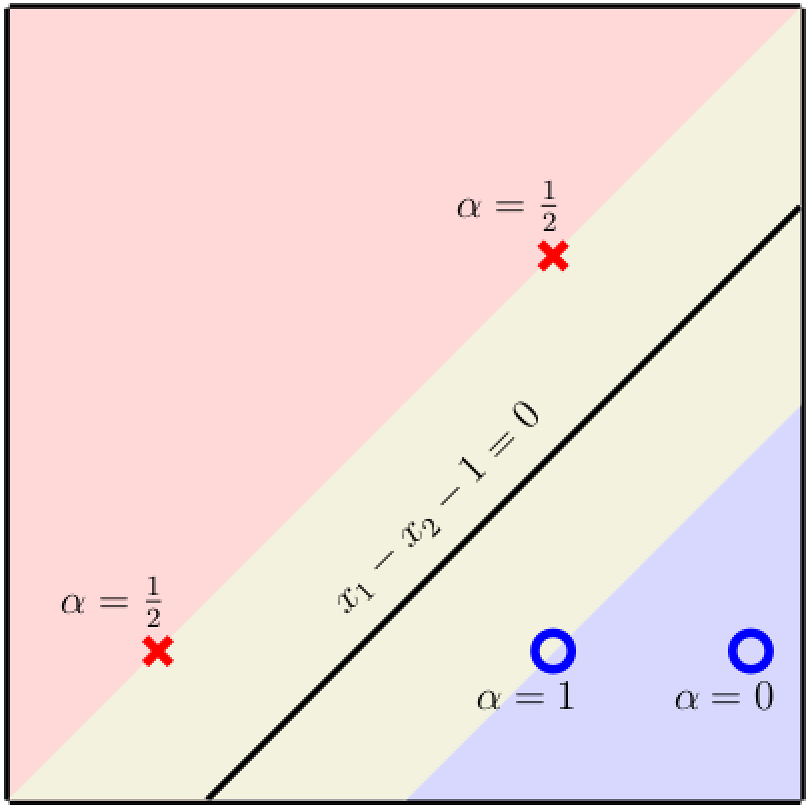
QP

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{w} = \sum_{n=1}^{4} \alpha_n^* y_n \mathbf{x}_n = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b = y_1 - \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 = -1$$

$$\gamma = \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$$



Los que no son vectores de soporte tienen: $\alpha_n = 0$

SVM en datos no separables (linealmente)

Resolver el problema en el dual nos permite trabajar en el espacio *Z*.

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad \tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathrm{G}\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha}$$

$$\text{subject to: } \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0$$

$$\mathbf{C} \geq \boldsymbol{\alpha} \geq \mathbf{0}$$

$$G_{nm} = y_n y_m(\mathbf{z}_n^{\mathrm{T}}\mathbf{z}_m)$$

$$g(\mathbf{x}) = \operatorname{sign}\left(\sum_{\alpha_n^*>0} \alpha_n^* y_n(\mathbf{z}_n^{\mathrm{T}}\mathbf{z}) + b^*\right)$$

$$C > \alpha_s^* > 0$$

$$b^* = y_s - \sum_{\alpha_n^*>0} \alpha_n^* y_n(\mathbf{z}_n^{\mathrm{T}}\mathbf{z}_s)$$

producto interno

SVM en datos no separables (linealmente)

Un kernel es una función que combina tanto la transformación como el producto interno:

$$K_\Phi(\mathbf{x}, \mathbf{x}') \equiv \Phi(\mathbf{x})^{\mathrm{T}} \Phi(\mathbf{x}').$$

El kernel toma dos vectores y entrega el producto interno en *Z*.

SVM en datos no separables (linealmente)

Un kernel es una función que combina tanto la transformación como el producto interno:

$$K_{\Phi}(\mathbf{x}, \mathbf{x}') \equiv \Phi(\mathbf{x})^{\mathrm{T}}\Phi(\mathbf{x}').$$

El kernel toma dos vectores y entrega el producto interno en *Z*.

Ejemplo: Kernel polinomial de segundo orden.

$$\Phi_2(\mathbf{x})^{\mathrm{T}}\Phi_2(\mathbf{x}') \quad = \quad 1 + (\mathbf{x}^{\mathrm{T}}\mathbf{x}') + (\mathbf{x}^{\mathrm{T}}\mathbf{x}')^2.$$

SVM en datos no separables (linealmente)

1: **Input:** $\mathrm{X}, \mathbf{y}$.

2: Compute G: $\mathrm{G}_{nm} = y_n y_m K(\mathbf{x}_n, \mathbf{x}_m)$.

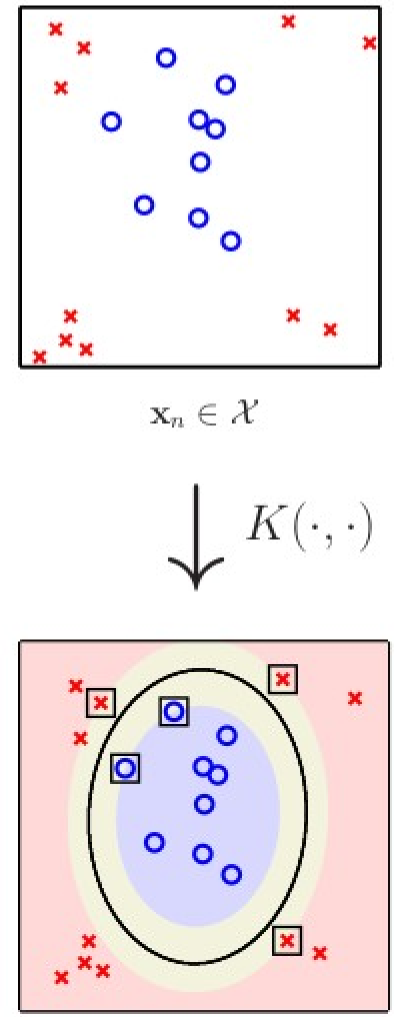
3: Solve (QP):

$$\left.\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\text{minimize:}} & \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathrm{G}\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{subject to:} & \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0 \\ & \boldsymbol{\alpha} \geq \mathbf{0} \end{array}\right\} \longrightarrow \begin{array}{l} \boldsymbol{\alpha}^* \\ \text{index } s: \quad \alpha_s^* > 0 \end{array}$$

4: $$b^* = y_s - \sum_{\alpha_n^* > 0} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x}_s)$$

5: The final hypothesis is

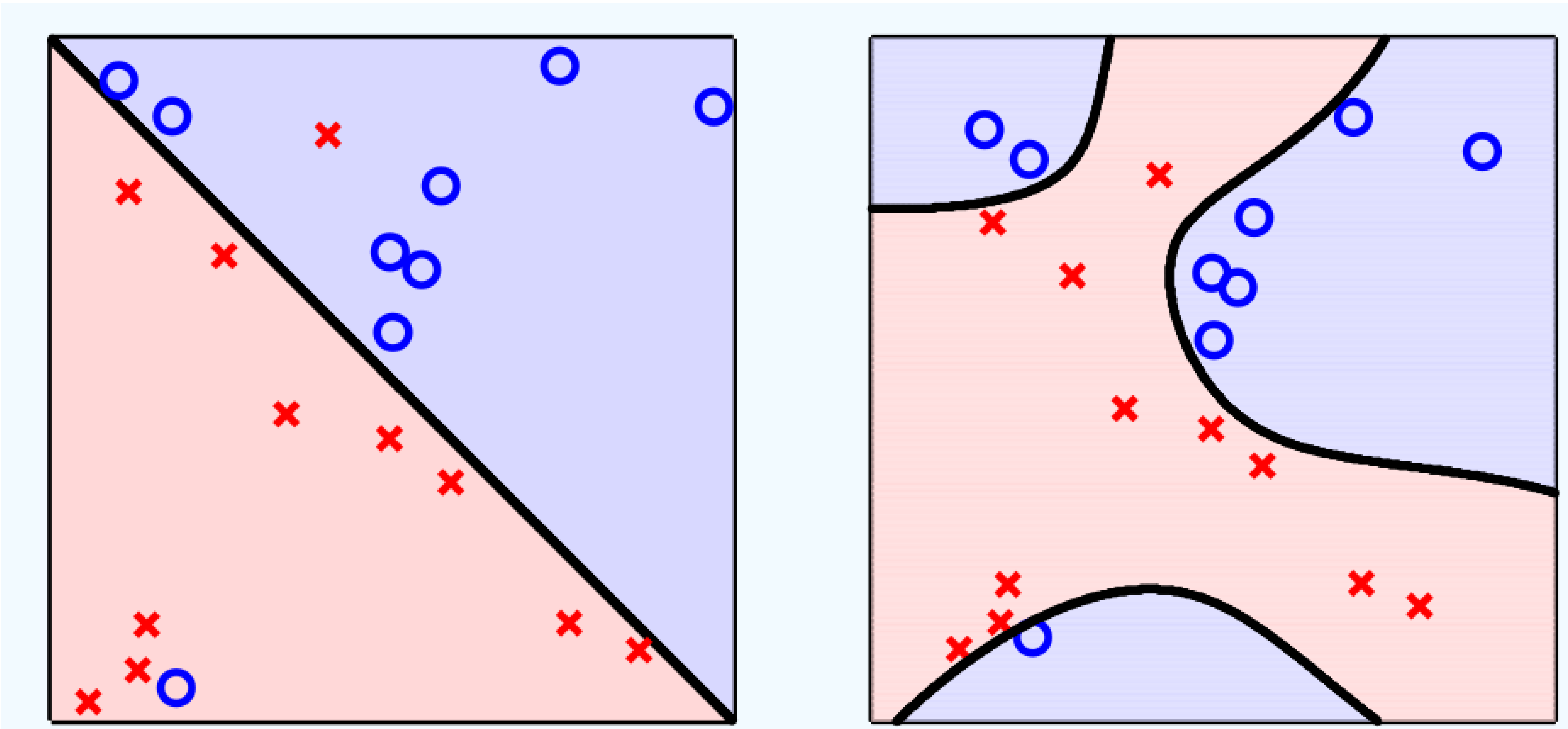
$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{\alpha_n^* > 0} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x}) + b^*\right)$$



$$g(\mathbf{x}) = \mathrm{sign}\left(\sum_{\alpha_n^* > 0} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x}) + b^*\right)$$

$$b^* = y_s - \sum_{\alpha_n^* > 0} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x}_s)$$

SVM con kernel Gaussiano



a) SVM lineal b) SVM con kernel Gaussiano