# Clase 11: Clasificador Lineal

Laboratorio de Datos, FCEyN, 30/04/2021

# Introducción

La clase de hoy probablemente sea más cortas que las anteriores y eso está bueno porque las últimas clases venimos a full y terminando sobre la hora o nos falta tiempo

No hay mucho más para decir de un Clasificador lineal que no hayamos dicho hasta ahora

Los conceptos nuevos de la clase de hoy son el discriminador lineal de Fisher el cual es un ejemplo particular de LDA (análisis discriminante lineal) y el perceptrón

La novedad de la notebook que vamos a ver hoy es el dataset y un pispeo a redes neuronales. Vamos a trabajar con un dataset de imágenes de distintas prendas de vestir (pantalones, remeras, zapatos, etc.). Al final de la clase voy a comentar cómo está compuesto el dataset y cómo vamos a trabajar con las imágenes.

# Resumen de lo visto

Vimos modelos de regresión: lineal, polinomial y logística [clases del 13-16-20/04]
donde optimizamos los pesos del modelo para obtener **y** números reales que mejor se ajusten a los **y** datos

Vimos clasificación simple (una sola clase binaria/dos clases) usando regresión logística [clase del 20/04]
donde ahora los **y** que queremos obtener tienen un rango acotado [0,1] porque queremos identificarlos con una determinada clase

Vamos a generalizar (y repetir) el concepto de clasificación y ver distintos tipos de clasificadores

# Definición

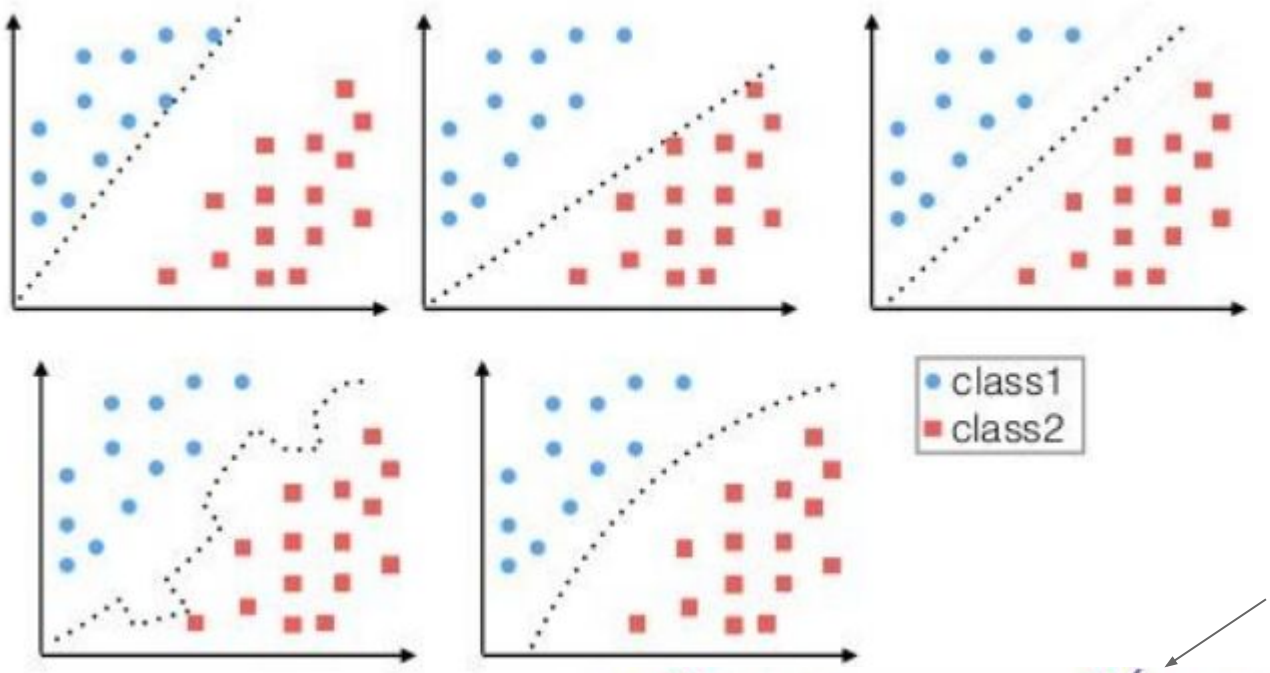
- Una definición de **clasificación**:

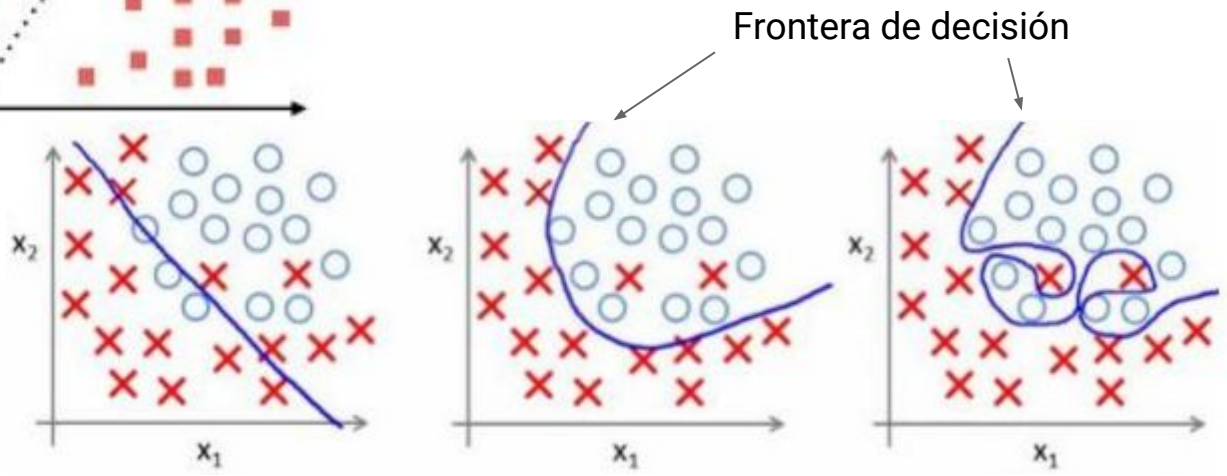
  "Usar las CARACTERÍSTICAS de un objeto para identificar a qué CLASE/CATEGORÍA pertenece"

  - Ejemplos:
    - Si un mail es "spam" o no
    - Qué tipo de objeto está presente en una imágen
    - Diagnosticar una enfermedad según los síntomas de la persona
    - El tipo celular según el transcriptoma de la célula en un experimento sc-RNAseq

- La clasificación es un ejemplo de **reconocimiento de patrones**

# Definición

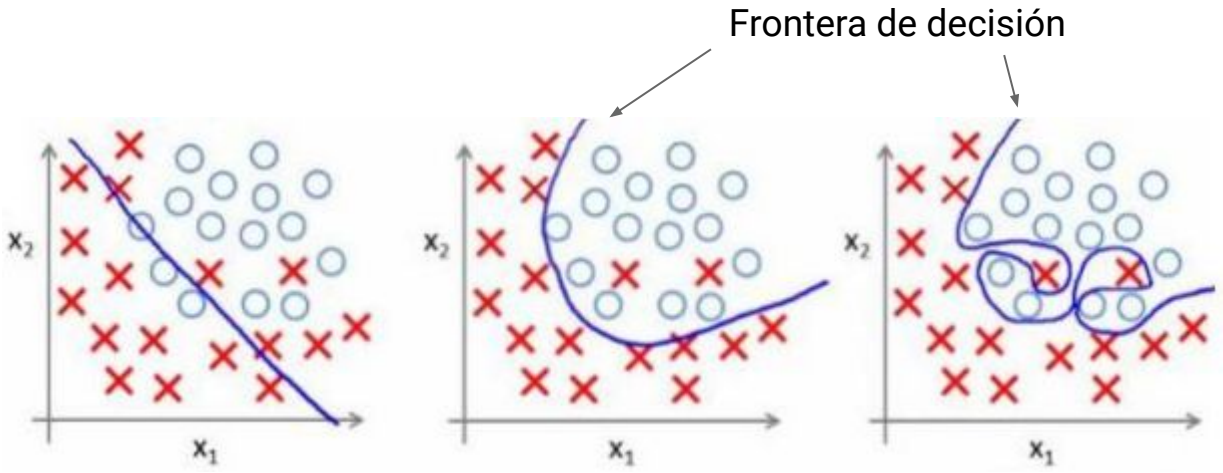


# Definición

En modelos lineales de clasificación las fronteras de decisión son funciones lineales del input **X** (vector de D-características), es decir están definidas por hiperplanos D-1 dimensionales

Datasets son *linealmente separables* si sus clases están bien separadas por fronteras de decisión lineales



# Generalización de la regresión logística

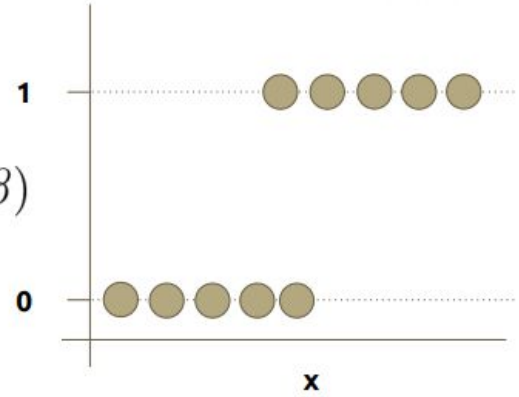
## La hipótesis de representación

Pensemos a f como una probabilidad y usemos una función con imagen acotada entre cero y uno

**función sigmoidea**

$$P(y|x,\beta) = \frac{1}{1+e^{-(\beta_0+\beta_1 \cdot x)}}$$

$$y = f(x,\beta) \equiv P(y|x,\beta)$$




$$y = f\left(\vec{w} \cdot \vec{x}\right)$$

$$f\left(\vec{w} \cdot \vec{x}\right) = f\left(\sum_j w_j \cdot x_j\right) = f(\beta_0 + \beta_1 \cdot x)$$

f es la **función de activación** y puede ser una función no lineal

# Identidad del clasificador lineal

- Clasificación lineal:
  - Un algoritmo de clasificación (*clasificador*) que hace sus clasificaciones (*predicciones*) basadas en una función de predicción lineal combinando un set de pesos, **W**, con el input **X**

$$y = f\left(\vec{w} \cdot \vec{x}\right) = f\left(\sum_j w_j \cdot x_j\right)$$

  - Fronteras de decisión planas:
    - líneas, planos, ...

# Diferentes “approaches”

- Creando explícitamente la **función discriminante:** (función que le asigna a cada vector **X** una clase y, y se optimizan los coeficientes w para disminuir el error)
  - Perceptrón
  - Support Vector Machines (SVM)
    [Clase que viene]

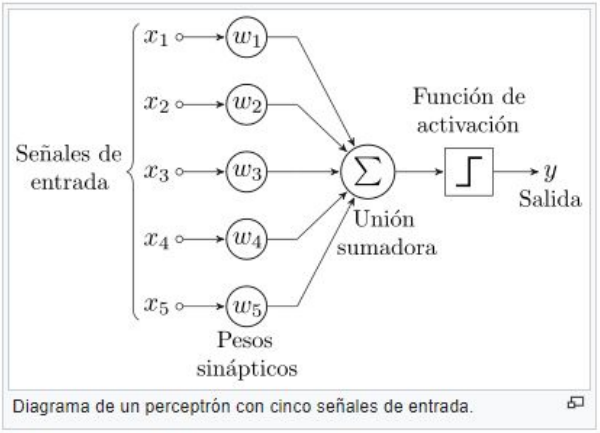


Diagrama de un perceptrón con cinco señales de entrada.

- Approach probabilístico:
  - Modelar la probabilidad condicional (Bayesiana)
  - Algoritmos
    - Regresión logística

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

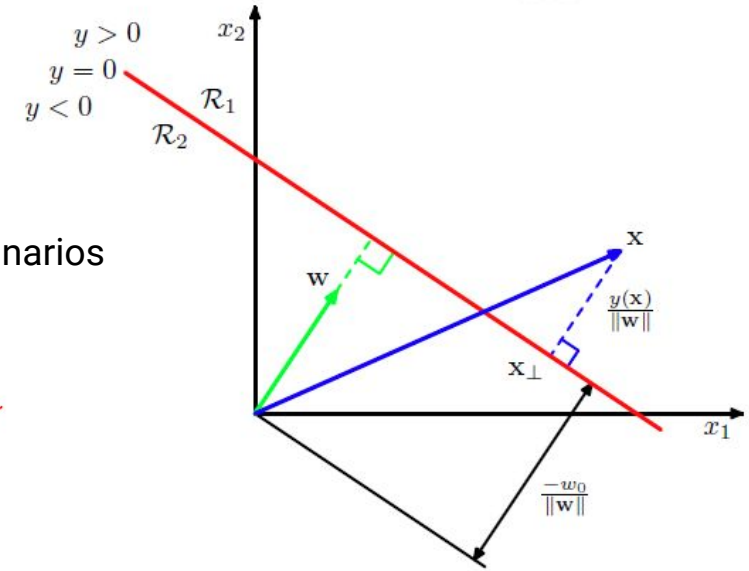
$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1+\exp(-a)} = \sigma(a) \end{aligned}$$

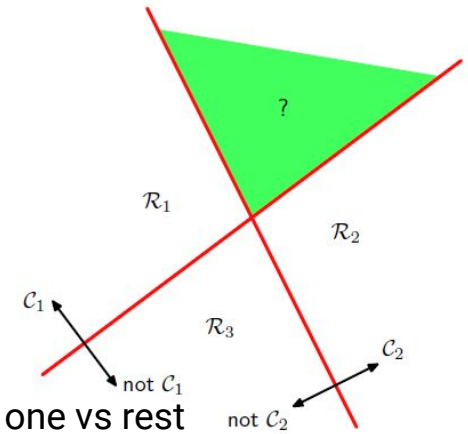
# Función Discriminante

- Dos clases (K=2):
  - Más simple -> función lineal del input $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

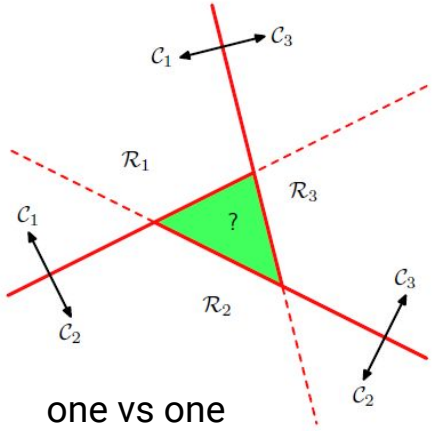
$$\widetilde{\mathbf{w}} = (w_0, \mathbf{w})$$
$$\widetilde{\mathbf{x}} = (x_0, \mathbf{x})$$
$$x_0 = 1$$
$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$




- Más de dos clases (K>2):
  - Tentados a usar K-1 clasificadores lineales binarios




one vs rest

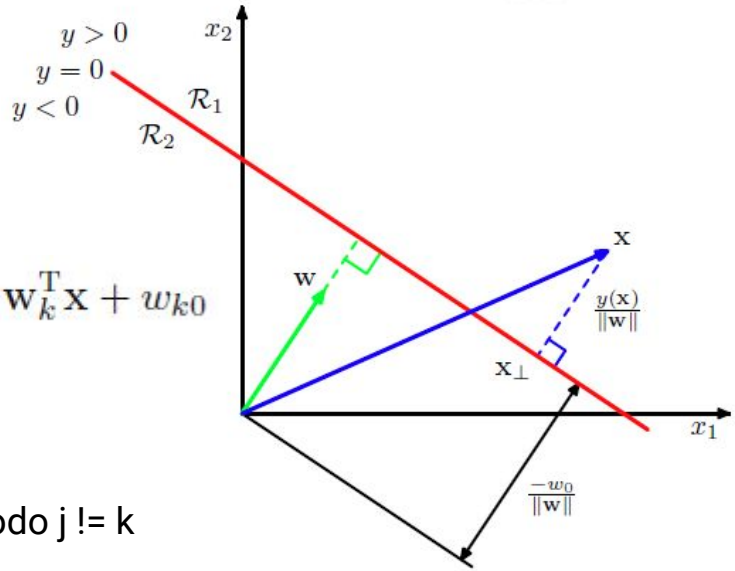



one vs one

# Función Discriminante

$$\widetilde{\mathbf{w}} = (w_0, \mathbf{w})$$
$$\widetilde{\mathbf{x}} = (x_0, \mathbf{x})$$
$$x_0 = 1$$
$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{x}}$$

- Dos clases (K=2):
  - Más simple -> función lineal del input: $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$




- Más de dos clases (K>2):
  - Función discriminante de K-clases: $y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$ (K funciones lineales)
  - Fronteras de decisión: $y_k(\mathbf{x}) = y_j(\mathbf{x})$
  - asigno X a la clase K si $y_k(\mathbf{x}) > y_j(\mathbf{x})$ para todo j != k
  - One-hot encoding: $y = (y_1, y_2, \ldots, y_k)$
    - w ahora es una matriz de pesos cuya k-ésima columna es $\widetilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^{\mathrm{T}})^{\mathrm{T}}$

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} \qquad \widetilde{\mathbf{x}} = (1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$$

# Aprendizaje de parámetros de función discriminante

1. Método basado en cuadrados mínimos

2. Discriminante lineal de Fisher
    - Método de reducción de dimensionalidad

3. Perceptrón

# Aprendizaje de parámetros de función discriminante

$$\mathrm{y}(\mathrm{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathrm{x}}$$

1. Método basado en cuadrados mínimos

- Ajusta simultáneamente un modelo de regresión lineal a cada una de las columnas de $y = (y_1, y_2, \ldots, y_k)$
  - Los pesos van a tener esta pinta: $\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T}$

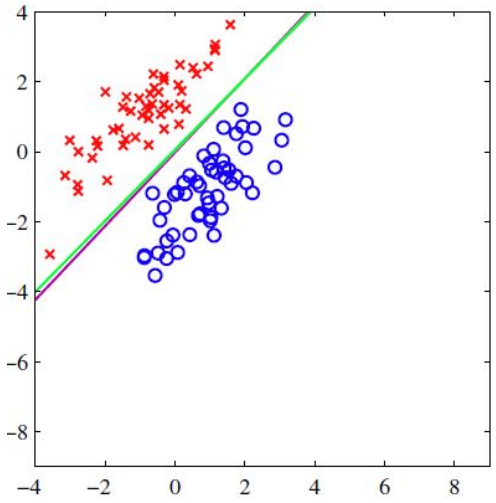
    donde $\mathbf{T}$ son las etiquetas reales y $\widetilde{\mathbf{X}}$ una matriz con los datos de entrada

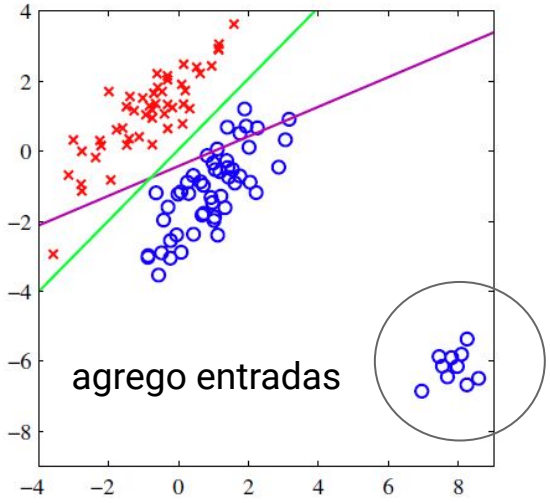
- Clasificar una nueva observación x:
  - Calculo la función discriminante para cada clase $y_i(x) = W \cdot X$
  - Me quedo con la clase que tenga mayor valor de $y_i(x)$

# Aprendizaje de parámetros de función discriminante

1. Método basado en cuadrados mínimos

- Funciona bien:
  - linealmente separables
  - pocos outliers
  - K = 2



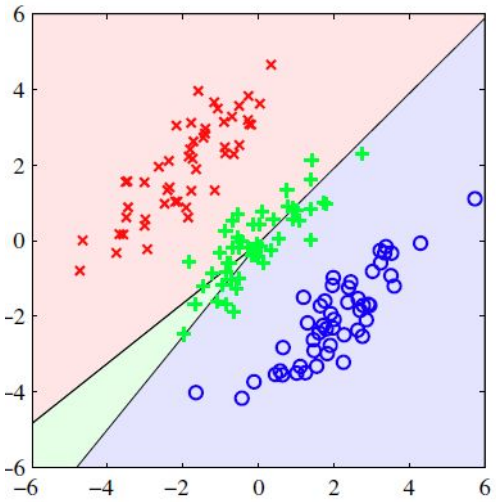

Frontera de decisión de la función logística

Frontera de decisión por cuadrados mínimos

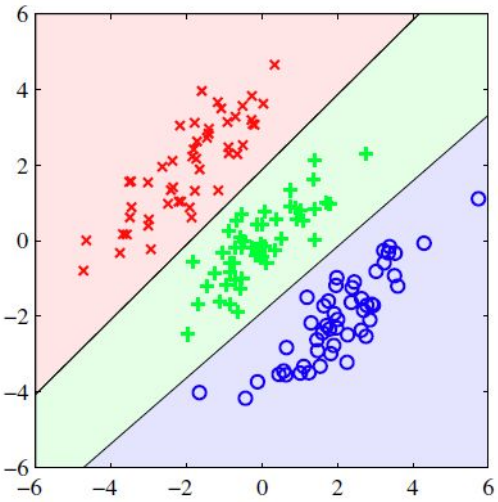
# Aprendizaje de parámetros de función discriminante

1. Método basado en cuadrados mínimos

- No funciona bien:
  - K > 2




cuadrados mínimos



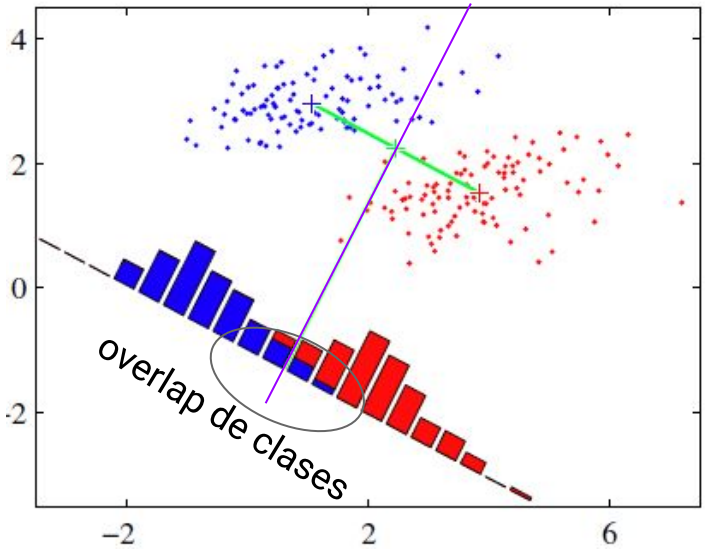

función logística

# Aprendizaje de parámetros de función discriminante

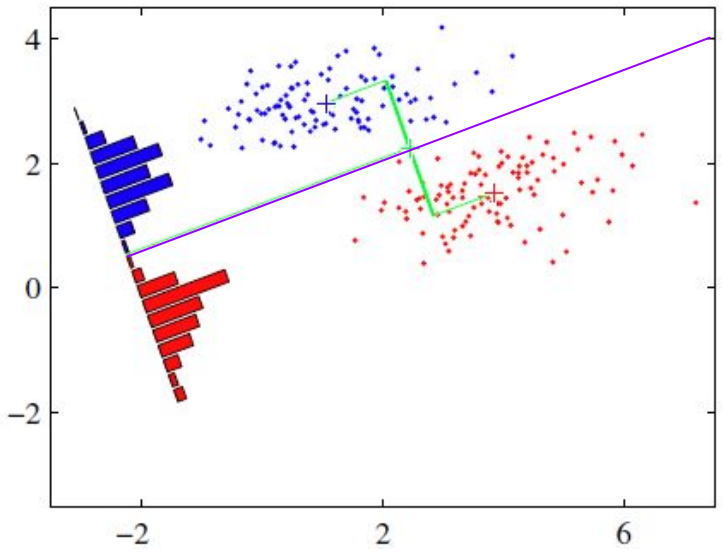
2. Discriminante lineal de Fisher

Criterio de Fisher: $J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$ $J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$

- Reducción de dimensionalidad




Histogramas de la proyección de las clases sobre la línea que pasa por sus medias



Histogramas de la proyección de las clases sobre la línea determinada por el discriminante lineal de Fisher

- Al proyectar estamos usando W -> para que no se disparen los pesos pedimos $\sum_i w_i^2 = 1$

# Aprendizaje de parámetros de función discriminante

3. Perceptrón

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right), f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0. \end{cases}$$

¿Cómo encontramos los pesos?

- Elegir unos pesos arbitrarios
- Para cada ejemplo $x_j$ del set de entrenamiento calculo $y_j$

$$\begin{aligned} y_j(t) &= f[\mathbf{w}(t) \cdot \mathbf{x}_j] \\ &= f[w_0(t)x_{j,0} + w_1(t)x_{j,1} + w_2(t)x_{j,2} + \cdots + w_n(t)x_{j,n}] \end{aligned}$$

- Actualizamos los pesos

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_{j,i} \quad 0 \leq i \leq n$$

$\alpha$ es el “learning rate”

$d_j$ es el vector de etiquetas reales

# Aprendizaje de parámetros de función discriminante

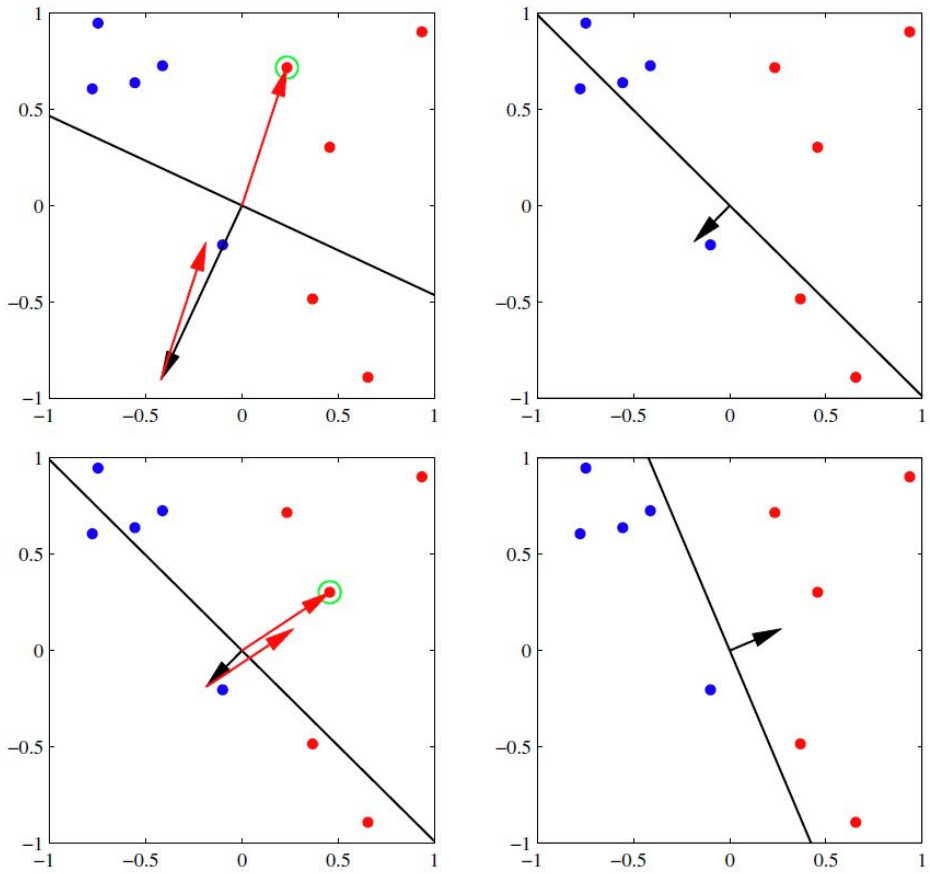
3. Perceptrón

la línea negra es el vector W

su punta indica dónde deberían estar los puntos rojos

Me paro en un dato (el verde) y me fijo si lo clasificó bien.
Si está bien clasificado no hago nada.
Si lo clasificó mal entonces le sumo su vector de features al vector W y así sucesivamente

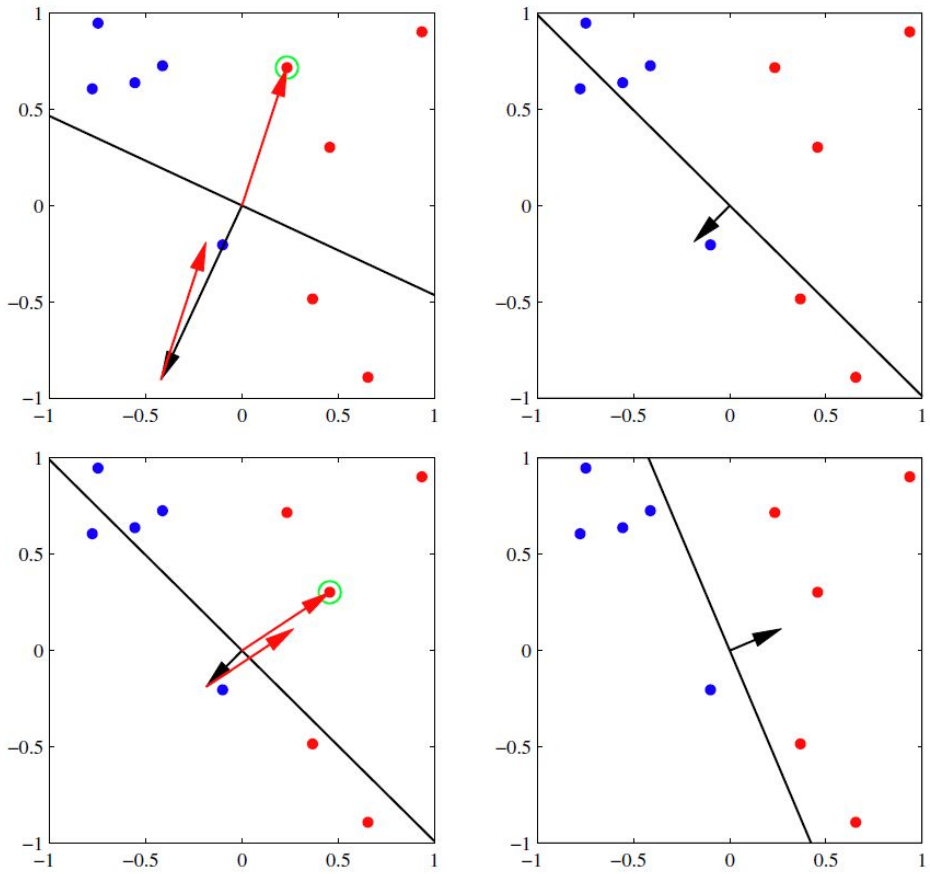
Termina cuando no queda ejemplos mal clasificados



# Aprendizaje de parámetros de función discriminante

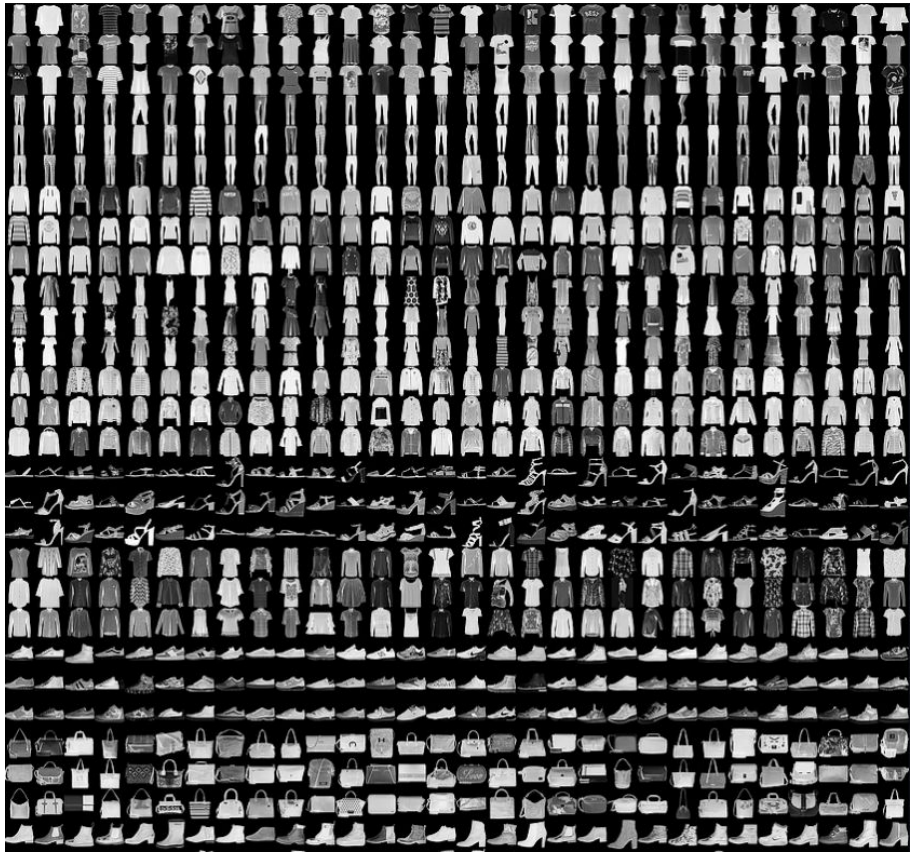
3. Perceptrón

- Funciona bien:
  - dataset linealmente separables
  - es rápido
- No tan bueno:
  - no elige las mejores fronteras de decisión



# Notebook

Dataset fashion.mnist



| Label | Description |
|---|---|
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |