

SVM

SVM

Las SVM son clasificadores derivados de la teoría de aprendizaje estadístico postulada por Vapnik y Chervonenkis.

Las SVM fueron presentadas en 1992 y adquirieron fama cuando dieron resultados muy superiores a las redes neuronales en el reconocimiento de letra manuscrita, usando como entrada píxeles.

Pretenden predecir a partir de lo ya conocido.

SVM

Dadas n observaciones, cada una consiste en un par de datos:

un vector $x_i \in R^n, i = 1, \dots, n$

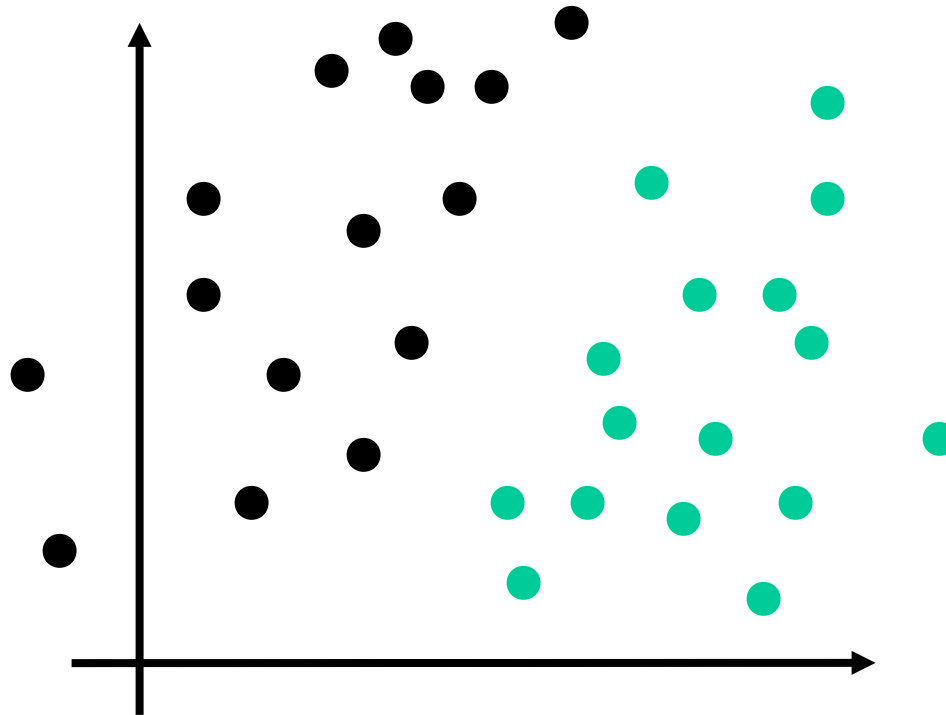
una etiqueta $y_i \in \{+1, -1\}$

Se pretende obtener un modelo simple para clasificarlas.

SVM Interpretación geométrica

● +1

● -1

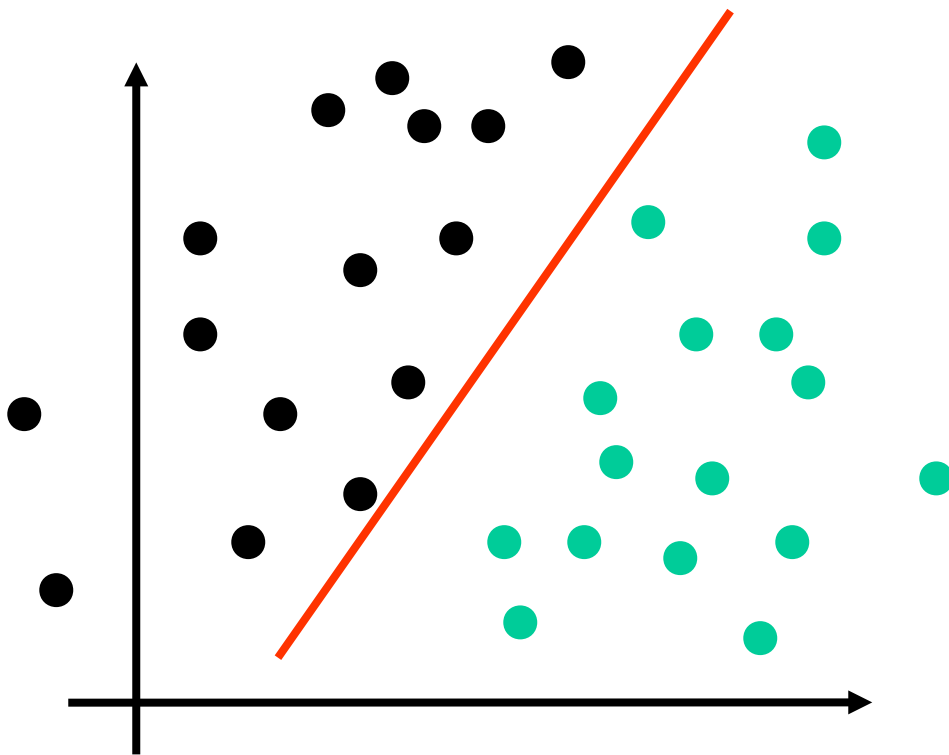


¿cómo clasificar estos datos?

SVM Interpretación geométrica

● +1

● -1

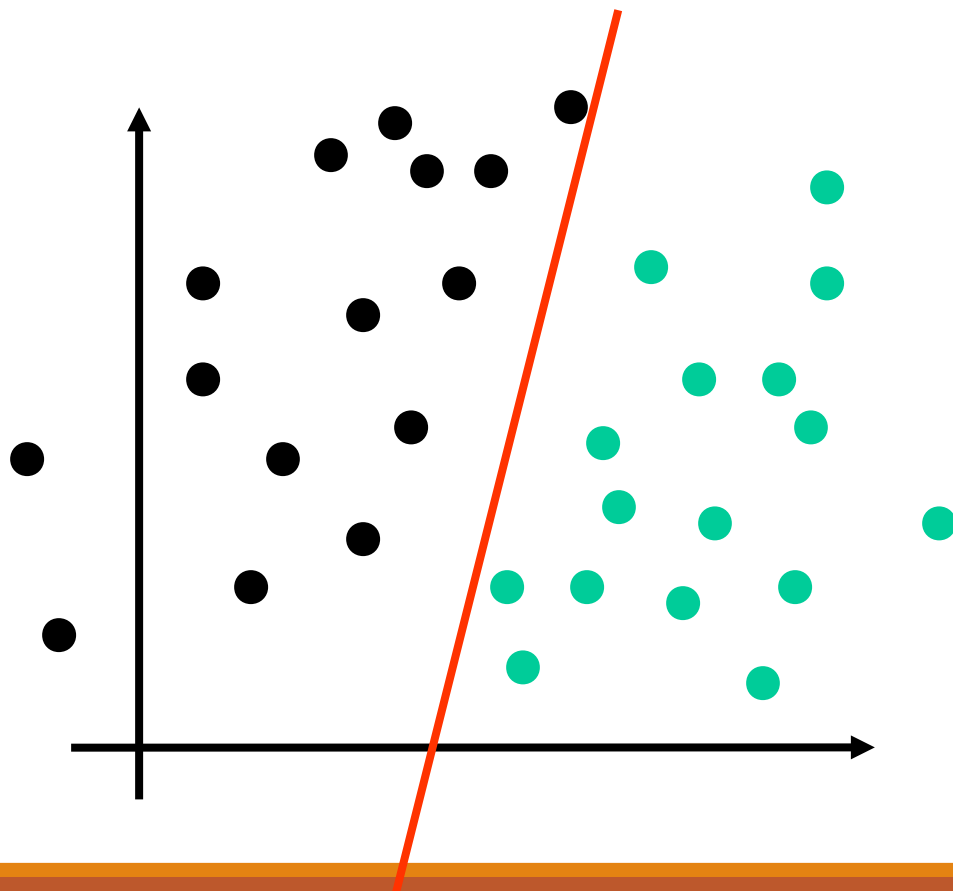


¿cómo clasificar estos datos?

SVM Interpretación geométrica

● +1

● -1



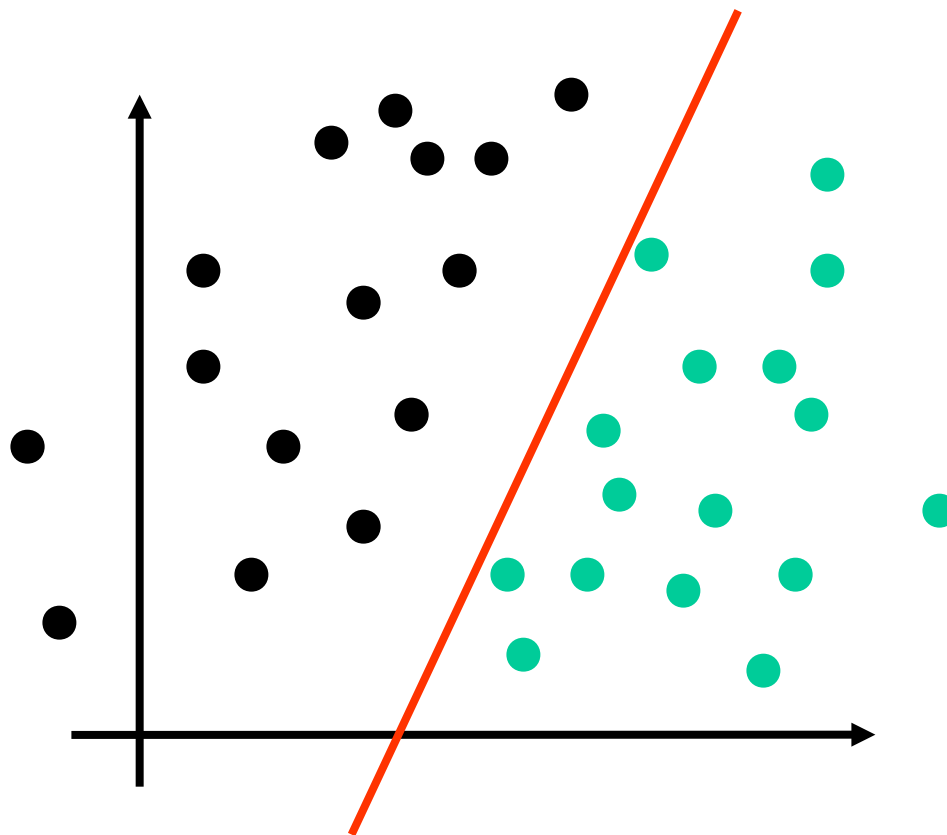
¿cómo
datos?

clasificar
estos

SVM Interpretación geométrica

● +1

● -1

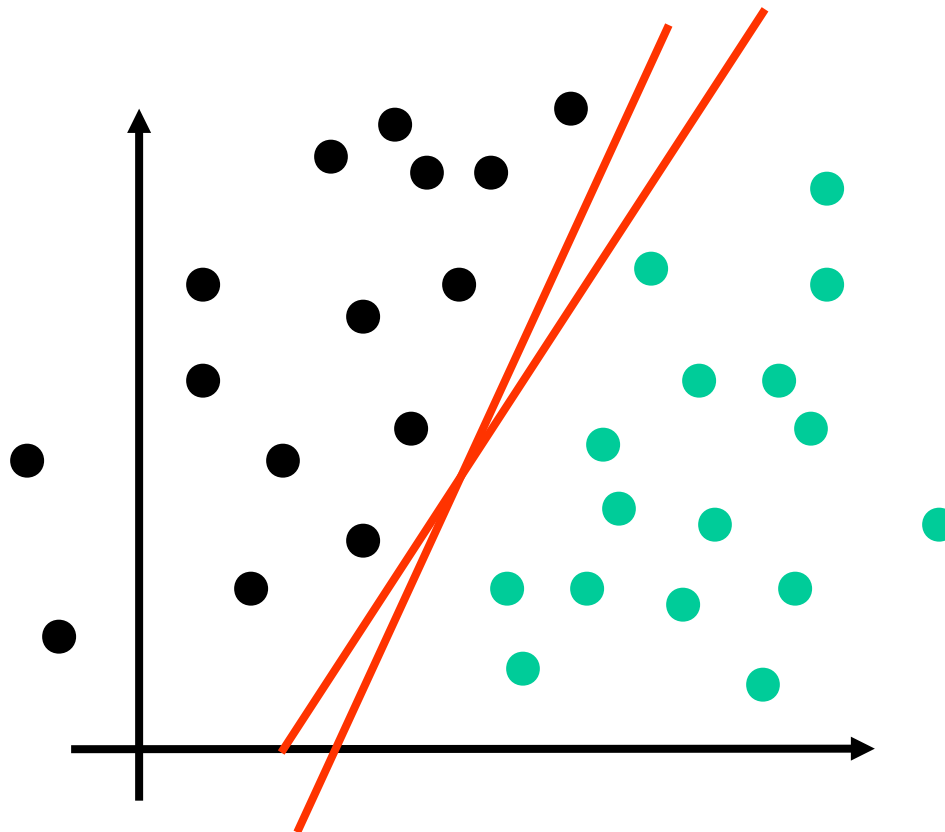


¿cómo
datos?
clasificar
estos

SVM Interpretación geométrica

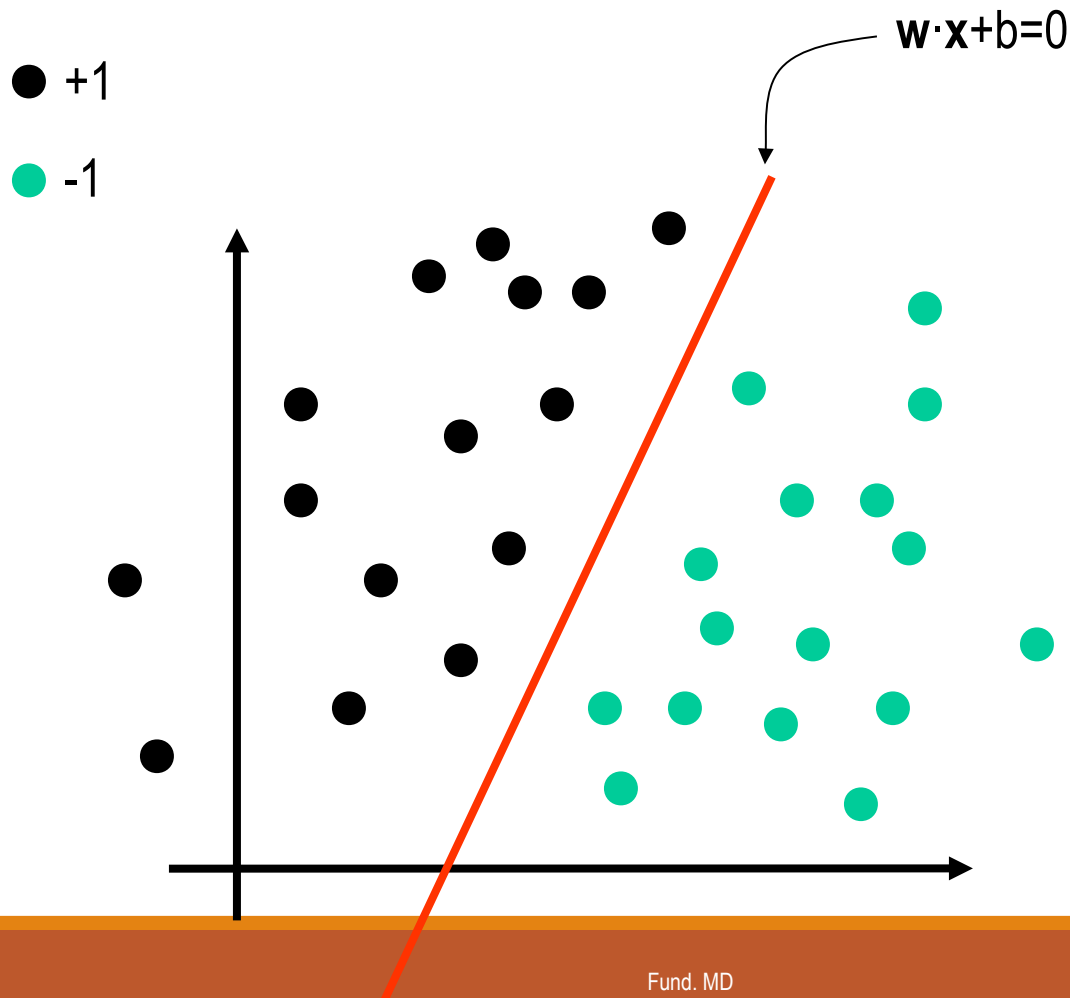
● +1

● -1



Cualquiera puede ser buena, ¿pero cuál es la mejor?

SVM Interpretación geométrica

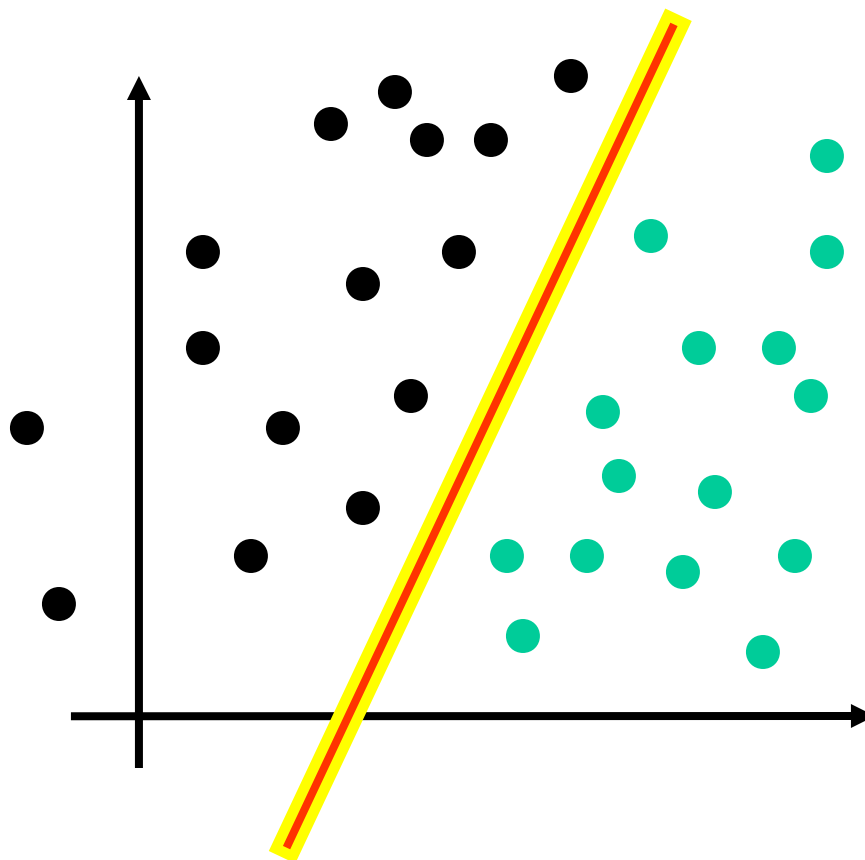


Definimos el hiperplano

SVM Interpretación geométrica

● +1

● -1

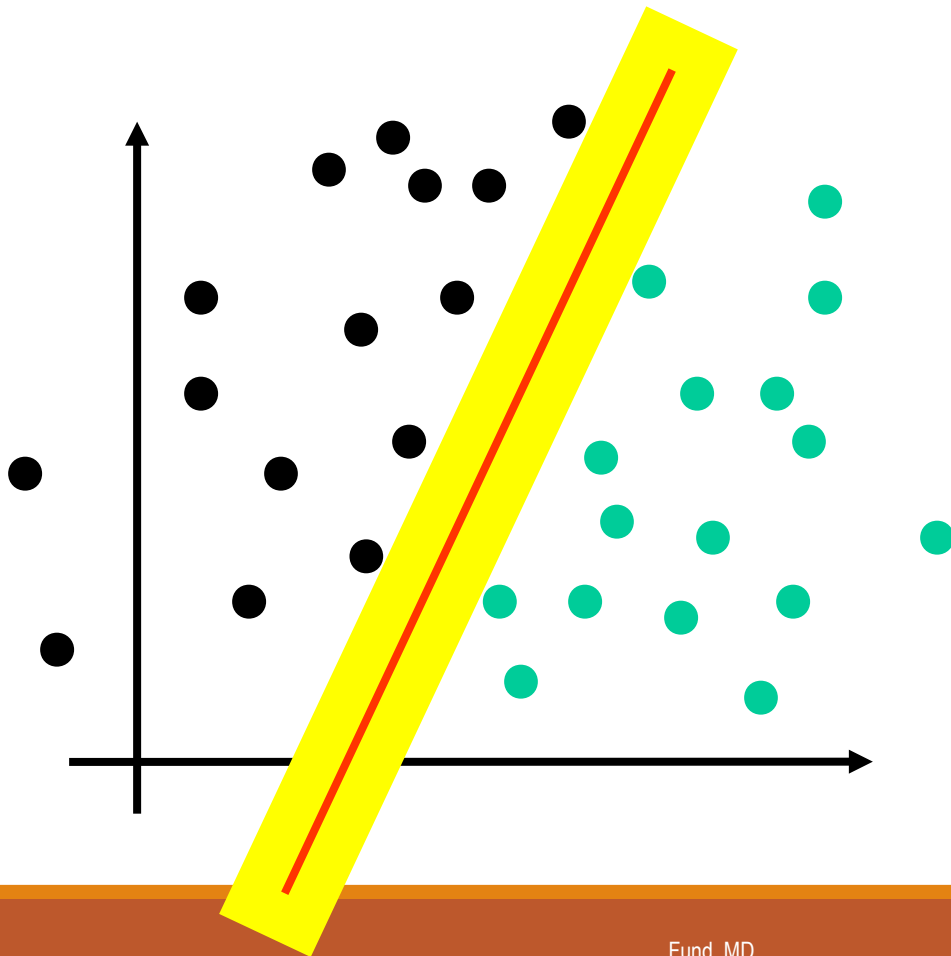


Definimos el margen

SVM Interpretación geométrica

● +1

● -1



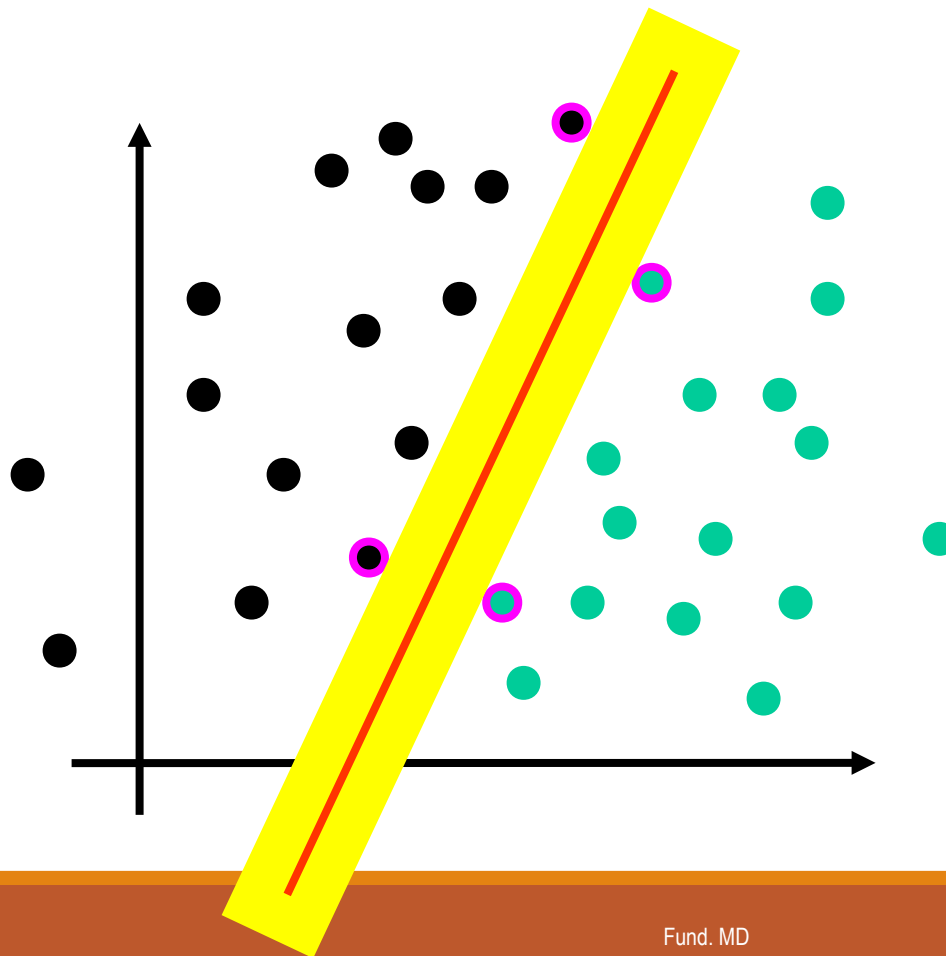
El hiperplano que tenga el mayor margen es el mejor clasificador de los datos.

Esta es la clase más simple de SVM, la LSVM.

SVM Interpretación geométrica

● +1

● -1

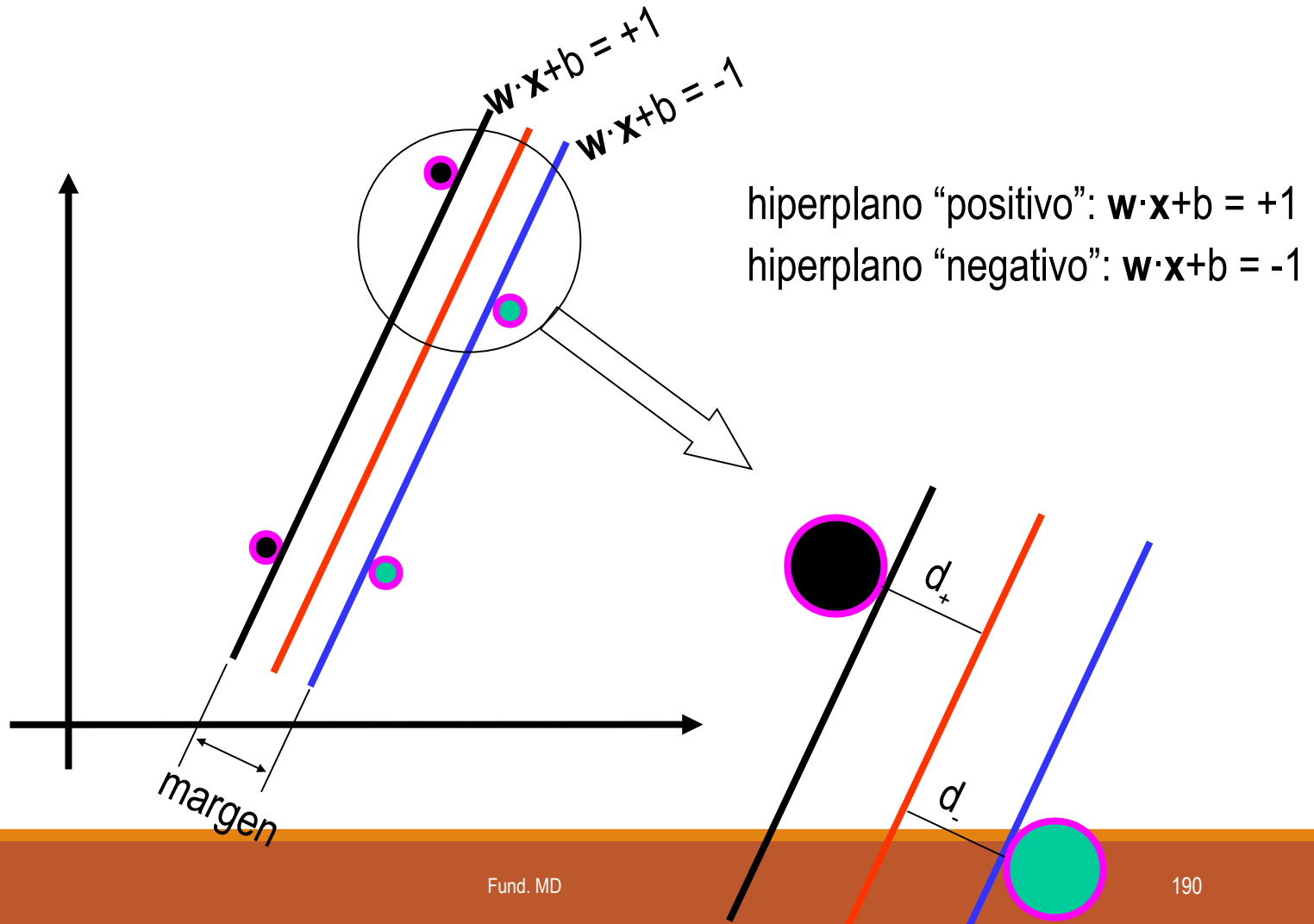


Los vectores de soporte son los puntos que tocan el límite del margen.

SVM Interpretación geométrica

● +1

● -1



SVM


Hay n observaciones y cada una consiste en un par de datos:

un vector $x_i \in R^n, i = 1, \dots, n$

una etiqueta $y_i \in \{+1, -1\}$

Supóngase que se tiene un hiperplano que separa las muestras positivas (+1) de las negativas (-1). Los puntos x_i que están en el hiperplano satisfacen $w \cdot x + b = 0$.

SVM

- w es normal al hiperplano
-  es la distancia perpendicular del hiperplano al origen
- $\|w\|$ es la norma euclídea de w

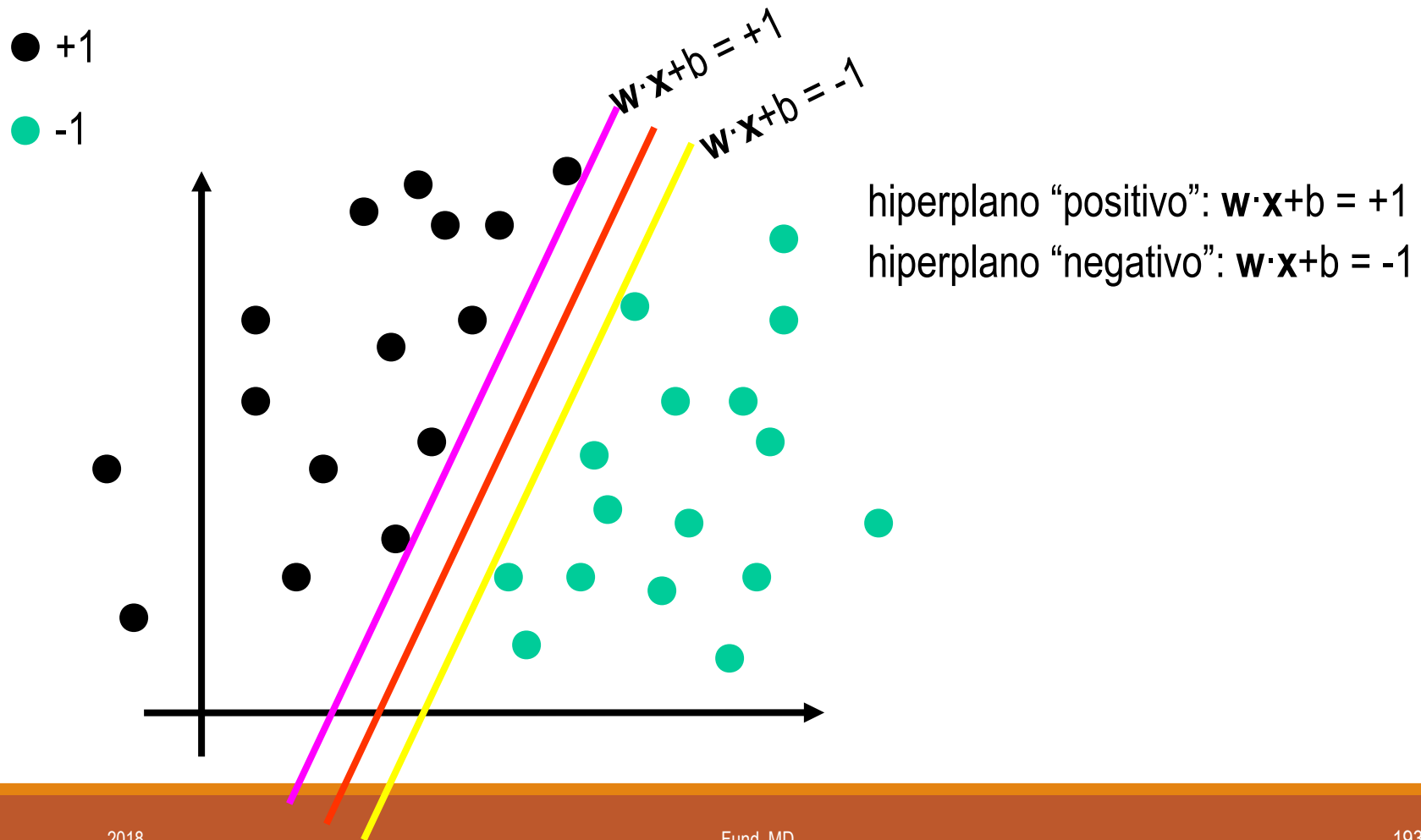
Lo que se quiere es separar los puntos de acuerdo al valor de su etiqueta y_i en dos hiperplanos diferentes:

$w \cdot x_i + b \geq +1$ para $y_i = +1$. (hiperplano “positivo”)

$w \cdot x_i + b \leq -1$ para $y_i = -1$ (hiperplano “negativo”)

Simplificado: $y_i(w \cdot x_i + b) \geq +1$

SVM



SVM

El problema se puede expresar:

minimizar $\|\mathbf{w}\|^2$

sujeto a $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq +1$

El problema se puede transformar para que quede más fácil de manejar! Se usan multiplicadores de Lagrange (~~4~~)

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^l \beta_i$$

SVM

Haciendo que los gradientes de L_p respecto a \mathbf{w} y b sean cero, se obtienen las siguientes condiciones:

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad \sum_{i=1}^l \alpha_i y_i = 0$$

Reemplazando en L_p se obtiene el problema dual:

$$L_D = \sum_{i=1}^l \alpha_i + \sum_{i=1, j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

SVM

La forma para optimizar es:

$$\text{maximizar } L_D = \sum_{i=1}^l \frac{1}{2} \|\mathbf{x}_i\|^2 + \sum_{i=1, j=1}^l \frac{1}{2} \langle \mathbf{x}_i, \mathbf{x}_j \rangle y_i y_j$$

$$\text{sujeto a } \mathbf{w} = \sum_{i=1}^l y_i \mathbf{x}_i \quad \sum_{i=1}^l y_i = 0$$

Problemas con SVM

Sobreentrenamiento: se han aprendido muy bien los datos de entrenamiento pero no se pueden clasificar bien ejemplos no vistos antes.

SVM son resistentes al sobreentrenamiento porque la clasificación depende solo de algunos datos de entrenamiento, los vectores de soporte.

La porción n de los datos no conocidos que será mal calificada, está limitada por:

$$n = \frac{\text{No. vectores de soporte}}{\text{No. de ejemplos de entrenamiento}}$$

SVM

Cuando los datos no se pueden separar linealmente se hace un cambio de espacio mediante una función que transforme los datos de manera que se puedan separar linealmente, se llama función ***Kernel***.

Por ejemplo: ***funciones polinómicas*** y ***funciones de Base Radial*** (RBF).

Operaciones de una SVM

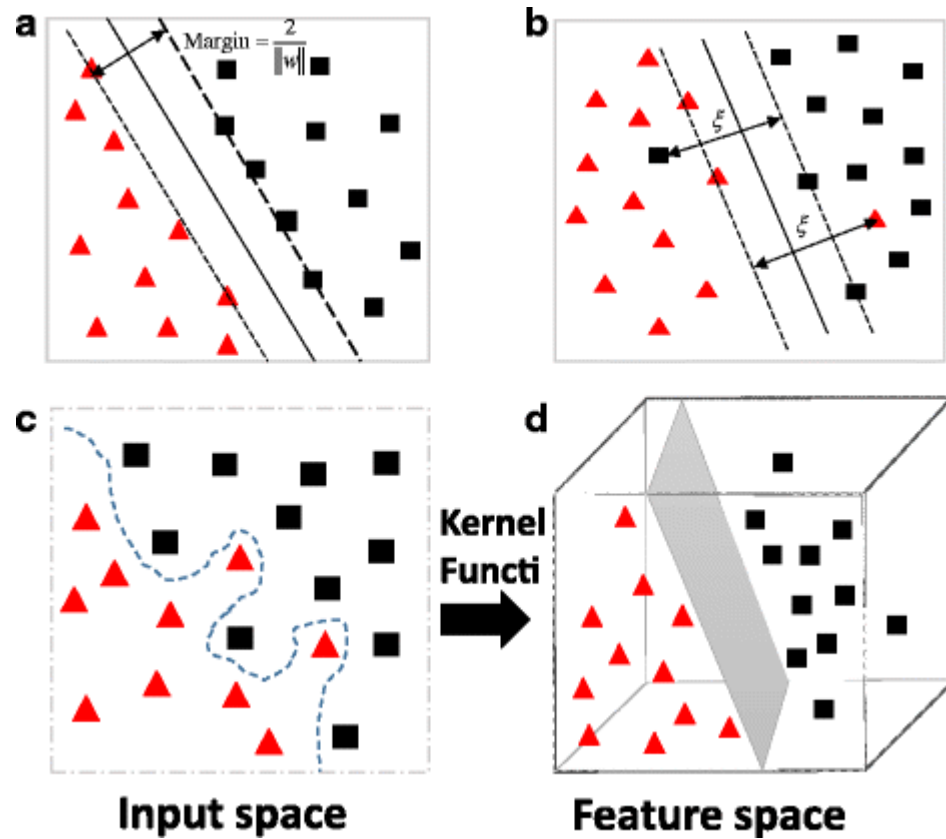
Transforma los datos a un espacio de dimensión muy alta a través de una función kernel -> se reformula el problema de tal forma que los datos se mapean implícitamente en este espacio.

Encuentra el hiperplano que maximiza el “margen” entre dos clases -> cálculo eficiente del hiperplano óptimo.

SVM

Si los datos no son linealmente separables encuentra el hiperplano que maximiza el margen y minimiza una función del número de clasificaciones incorrectas (término de penalización de la función) → Soft margin.

SVM



Funciones Kernel SVM

Lineal: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

Polinomial de grado p : $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$

Gausiana (radial-basis function RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Sigmoidea: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$

SVM

Flexibilidad en la elección de la función de similaridad
Eficiencia en el cálculo de la solución aún con grandes conjuntos de datos

- sólo se usan los vectores de soporte para calcular el hiperplano

Capacidad de modelar datos con gran dimensionalidad

- complejidad no depende de la dimensionalidad del espacio

Sobreentrenamiento se controla con soft margin

Propiedad matemática

- optimización convexa simple, converge a una solución global

SVM

SVM se usa en problemas como:

- categorización de texto
- clasificación de imágenes
- bioinformática (clasificación de proteínas, diagnósticos, etc)
- reconocimiento de caracteres escritos a mano

SVM

Sólo clasifica en dos clase

¿Cómo hacer clasificación multiclase?

Por ejemplo entrenar m SVM's, una para cada clase

SVM 1 aprende "Output==1" vs "Output != 1"

SVM 2 aprende "Output==2" vs "Output != 2"

...

SVM m aprende "Output== m " vs "Output != m "

Para un nuevo dato, obtener resultado con cada una de los m modelos de SVM y asignar al que da una predicción más adentrada en la región positiva.

SVM vs ANN

ANNs

- Capas ocultas transforman a espacios de cualquier dimensión
- Espacio de búsqueda con múltiples mínimos locales
- Entrenamiento costoso
- Clasificaciones muy eficiente
- Se diseña el número de capas ocultas y nodos
- Muy buen funcionamiento en problemas típicos

SVMs

- Kernels transforman a espacios de dimensión muy superior
- Espacio de búsqueda tiene sólo un mínimo global
- Entrenamiento muy eficiente
- Clasificación muy eficiente
- Se diseña la función kernel y otros parámetros
- Muy buen funcionamiento en problemas típicos
- Extremadamente robusto para generalización
- Menos necesidad de heurísticas para entrenamiento