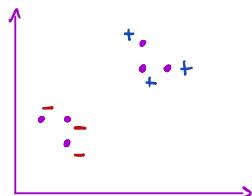


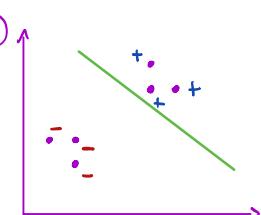
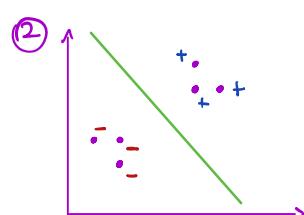
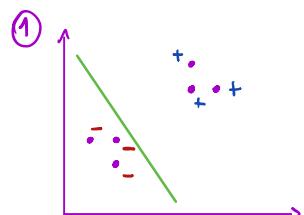
SVM

Ya hemos disutido sobre clasificadores lineales y clasificadores basados en distancia. Ahora vamos a ver un modelo que mezcla estos dos ideas y que además se puede extender para hacer clasificación no lineal.

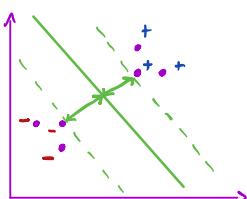
Supongamos el siguiente Dataset:



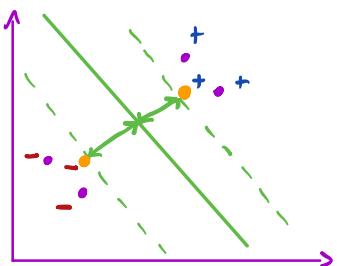
¿Cuál de los siguientes opciones clasificaría mejor?



Pareciera que la opción ② es un clasificador más general esto es porque la frontera de decisión maximiza la distancia a ambos conjuntos.



La idea del modelo **Support Vector Machine (SVM)** es clasificar con este idea: encontrar la "calle más ancha" que separe ambas clases. El margen del clasificador a ambas clases se representa por la linea punteada. Es importante ver que agregar instancias fuera de la calle no cambie el clasificador.



Así, el clasificador depende de los elementos a la orilla de la calle (puntos naranjos), que son llamados **Vectores de Soporte**.

Calculando el clasificador

Supongamos un vector de pesos \vec{w} y un coeficiente de posición (o **bias** en inglés) b .

Para clasificar una instancia desconocida x tendremos:

$$\hat{y} = \begin{cases} 0 & \text{si } \vec{w}^T x + b < 0 \\ 1 & \text{si } \vec{w}^T x + b \geq 0 \end{cases}$$

con $\vec{w}^T x = w_1 x_1 + \dots + w_k x_k$
(es decir, tenemos k features)

Nos gustaría tener un \vec{w} y un b tal que:

$$\begin{cases} \vec{w}^\top x_+ + b \geq 1 & \text{con } x_+ \text{ un ejemplo } \oplus \\ \vec{w}^\top x_- + b \leq -1 & \text{y } x_- \text{ un ejemplo } \ominus \end{cases}$$

→ Esto es, si tomamos un ej. \oplus o \ominus queremos tener una "holgura"

¿Por qué? Supongamos: (Parentesis)

$$\vec{w}^\top x_+ + b \geq \delta \quad \text{con } \delta > 0$$

$$\vec{w}^\top x_- + b \leq -\delta$$

↓

$$\left(\frac{1}{\delta}\right) \vec{w}^\top x_+ + \left(\frac{b}{\delta}\right) \geq 1 \Rightarrow \vec{w}'^\top x_+ + b' \geq 1$$

$$\left(\frac{1}{\delta}\right) \vec{w}^\top x_- + \left(\frac{b}{\delta}\right) \leq -1 \Rightarrow \vec{w}'^\top x_- + b' \leq -1$$

Con $\vec{w}' = \frac{1}{\delta} \vec{w}$, y como veremos, querremos minimizar $\|\vec{w}'\|$, por lo que no importa que este escala do por $\frac{1}{\delta}$.

Entonces, al entrenar, tenemos nuestros ejemplos de tal forma que la respuesta y_i es -1 para ejemplos negativos y $+1$ para positivos.

Así, diremos que para los elementos en los bordes de "la calle" cumplen con:

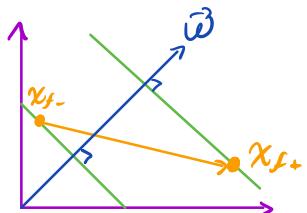
$$y_i (\vec{w}^\top \vec{x}_i + b) - 1 = 0 \quad (1)$$

Ahora, el ancho de "la calle" es:

$$\text{width} = (\vec{x}_{f+} - \vec{x}_{f-}) \cdot \frac{\vec{w}}{\|\vec{w}\|}$$

$\vec{x}_{f+}, \vec{x}_{f-}$ son
 x en el borde
⊕ y ⊖ resp

¿Por qué? La intuición (en 2D)



$(\vec{x}_{f+} - \vec{x}_{f-})$ es el vector naranja
 \vec{w} es perpendicular a la frontera de decisión y a las calles del borde

Así, es claro que el ancho es la proyección de $(\vec{x}_{f+} - \vec{x}_{f-})$ sobre la dirección del vector \vec{w} .

Recordemos que de (1) tenemos:

$$\underbrace{(\vec{x}_{f+} - \vec{x}_{f-})}_{((1-b) + (1+b))} \cdot \frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$

Nosotros queremos maximizar $\frac{2}{\|\vec{w}\|}$

$$\text{Max } \frac{2}{\|\vec{\omega}\|} \rightsquigarrow \text{Max } \frac{1}{\|\vec{\omega}\|} \rightsquigarrow \text{Min } \|\vec{\omega}\| \rightsquigarrow \text{Min } \frac{1}{2} \|\vec{\omega}\|^2$$

Entonces tenemos:

$$\begin{aligned} & \text{Min } \frac{1}{2} \|\vec{\omega}\|^2 \\ \text{s.t. } & y_i (\vec{\omega}^\top \vec{x}_i + b) \geq 1 \quad \text{for } i = 1, \dots, n \end{aligned}$$

$\underbrace{\text{ejemplos}}_{\text{de entrenamiento}}$

Para resolver este problema de optimización, usamos Lagrange:

$$\mathcal{L} = \frac{1}{2} \|\vec{\omega}\|^2 - \sum \alpha_i [y_i (\vec{\omega}^\top \vec{x}_i + b) - 1]$$

$$\frac{\delta \mathcal{L}}{\delta \vec{\omega}} = \vec{\omega} - \sum \alpha_i y_i \vec{x}_i = 0$$

\Downarrow

$$\vec{\omega} = \sum \alpha_i y_i \vec{x}_i$$

Ojo: \mathcal{L} es una función de $\mathbb{R}^n \rightarrow \mathbb{R}$, así que lo que estamos calculando es $\nabla \mathcal{L}$

$$\frac{\delta \mathcal{L}}{\delta b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0 \quad (2)$$

Al reemplazar los valores en \mathcal{L} obtenemos el problema dual:

$$\mathcal{L} = \frac{1}{2} \left(\sum \alpha_i y_i \vec{x}_i \right) \cdot \left(\sum \alpha_j y_j \vec{x}_j \right) - \left(\sum \alpha_i y_i \vec{x}_i \right) \cdot \left(\sum \alpha_j y_j \vec{x}_j \right) - \underbrace{\sum \alpha_i y_i b}_{0 \text{ debido a (2)}} + \sum \alpha_i$$

A sr, tenemos:

$$\max \sum_{i=1}^n d_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_i d_j y_i y_j x_i^T x_j$$

O bien:

$$\min \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_i d_j y_i y_j x_i^T x_j - \sum d_i$$

$$\text{s.a. } d_i \geq 0 \text{ con } i=1, \dots, n$$

Ojo, maximizamos en función de d_i porque este es el problema dual. Esto lo podemos hacer porque el problema de optimización original respeta ciertas condiciones.

(Parentesis) Al usar Lagrange uno tiene:

Problema primal: $\min_{\vec{w}, b} \max_{\vec{x} \geq 0} \frac{1}{2} \|\vec{w}\|^2 - \sum d_i [y_i (\vec{w}^T \vec{x}_i + b) - 1]$

Problema dual: $\max_{\vec{d} \geq 0} \min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2 - \sum d_i [y_i (\vec{w}^T \vec{x}_i + b) - 1]$

El problema dual es un problema de **Programación Cuadrática**:

$$\min \frac{1}{2} \sum_i \sum_i d_i d_j y_i y_j x_i^T x_j - \sum d_i$$

Tenemos variables multiplicadas entre si en la función objetivo

Estos son nuestros datos, finalmente dependemos de un producto punto entre los datos (esto será importante)

Podemos pasarlo este problema a un solver con nuestros datos y listo! Finalmente cuando tenemos los valores de $\vec{\alpha}$, podemos calcular \hat{w} y \hat{b} .

$$\hat{w} = \sum_{i=1}^n \hat{\alpha}_i y_i x_i$$

$$\hat{b} = \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i - \hat{w}^T x_i)$$

$\hat{\alpha}_i > 0$

Con n_s número de vectores de soporte.

Aunque aquí hay un detalle: usamos el supuesto que nuestros datos podían ser separados perfectamente por una línea. ¿Qué pasa en caso contrario?

Para resolver este problema estudiaremos el modelo

Soft Margin SVM