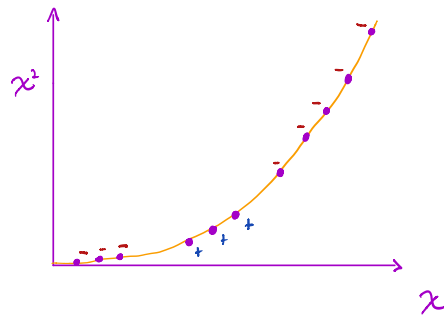


Kernels y SVM

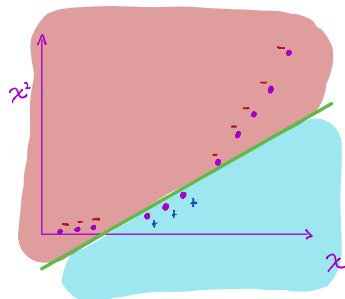
Recordemos el ejemplo del siguiente dataset unidimensional:



Una forma de resolver este problema es llevar este dataset a una dimensión más alta usando una función ϕ que lleve el punto de coordenada x al punto en el plano (x, x^2) .



Aquí sí podemos encontrar una forma de resolver nuestro problema:



¿Qué hace a esta técnica particularmente útil junto a los SVM?

Lo que la hace útil es que **no tenemos que transformar nuestras instancias gracias al "kernel trick".**

Transformación y Kernel

La transformación ϕ del punto anterior puede ser descrita como $(x, x^2, \frac{1}{2})$ en tres dimensiones (como el eje z esté fijo no es relevante).

Si tomamos dos puntos a y b en una dimensión, el producto punto $\phi(a) \cdot \phi(b)$ es:

$$\begin{pmatrix} a \\ a^2 \\ \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} b \\ b^2 \\ \frac{1}{2} \end{pmatrix} = ab + a^2b^2 + \frac{1}{4} = \left(ab + \frac{1}{2}\right)^2$$

Entonces, al resolver el problema de optimización dual:

$$\min \quad \frac{1}{2} \sum_i \sum_j d_i d_j \gamma_i \gamma_j x_i^T x_j - \sum d_i \quad \text{con } d_i \geq 0$$

Debemos reemplazar el producto punto:

$$x_i \cdot x_j = x_i^T x_j \quad \text{por} \quad \phi(a) \cdot \phi(b) = (ab + \frac{1}{2})^2$$

Lo que es bueno porque no tenemos que transformar los datos a la hora de entrenar nuestro modelo, solo tenemos que calcular el producto punto con $(ab + \frac{1}{2})^2$.

Este tipo de funciones que calculan el producto punto sin hacer transformaciones es lo que llamaremos **Kernel**. Esta técnica hace el proces mucho más eficiente.

Formalmente, un **Kernel** es una función K capaz de computar el producto punto

$$\phi(\vec{a})^T \phi(\vec{b})$$

Basado solamente en los valores originales de \vec{a} y \vec{b} . En el ejemplo, \vec{a} y \vec{b} eran unidimensionales, pero en general trabajaremos con vectores.

Ojo, nunca vamos a tener que computar ϕ , de hecho hay veces que no lo conoceremos.

Algunos Kernel:

- Linear: $K(\vec{a}, \vec{b}) = \vec{a}^T \vec{b}$
- Polinomial: $K(\vec{a}, \vec{b}) = (\gamma \vec{a}^T \vec{b} + r)^d$
 - ↳ En el ejemplo usamos este con $\gamma=1$, $r=\frac{1}{2}$ y $d=2$
- Gaussian RBF: $K(\vec{a}, \vec{b}) = e^{(-\gamma \|\vec{a} - \vec{b}\|^2)}$
- Sigmoid: $K(\vec{a}, \vec{b}) = \tanh(\gamma \vec{a}^T \vec{b} + r)$

Teorema de Mercer

El teorema de Mercer nos dice que si una función $K(\vec{a}, \vec{b})$ respeta ciertas condiciones (llamadas "Mercer's Condition"), entonces existe una función ϕ tal que mapee \vec{a} y \vec{b} a un espacio (de posiblemente muchas dimensiones) donde:

$$K(\vec{a}, \vec{b}) = \phi(\vec{a})^T \phi(\vec{b})$$

Probablemente, nunca conozcamos ϕ . Por ejemplo con el Kernel Gaussian RBF podemos mostrar que ϕ mapea cada instancia a un espacio

de dimensiones infinitas.

Ahora para predecir, uno en vez de calcular \vec{w} y \vec{b} , uno puede expresar la función de decisión en terminos del Kernel (i.e. productos punto).