Proyecto 2

Optimización en ML – SVM

MM3025 Métodos Numéricos 2

Objetivos

Definir el algoritmo SVM.

Deducir el problema de optimización.

Comprender el funcionamiento del algoritmo SMO para resolver el problema del entrenamiento de SVM.

¿Qué es SVM?

SVM

Support Vector Machines – SVM es un algoritmo de aprendizaje supervisado que se utiliza en problemas de clasificación y regresión.

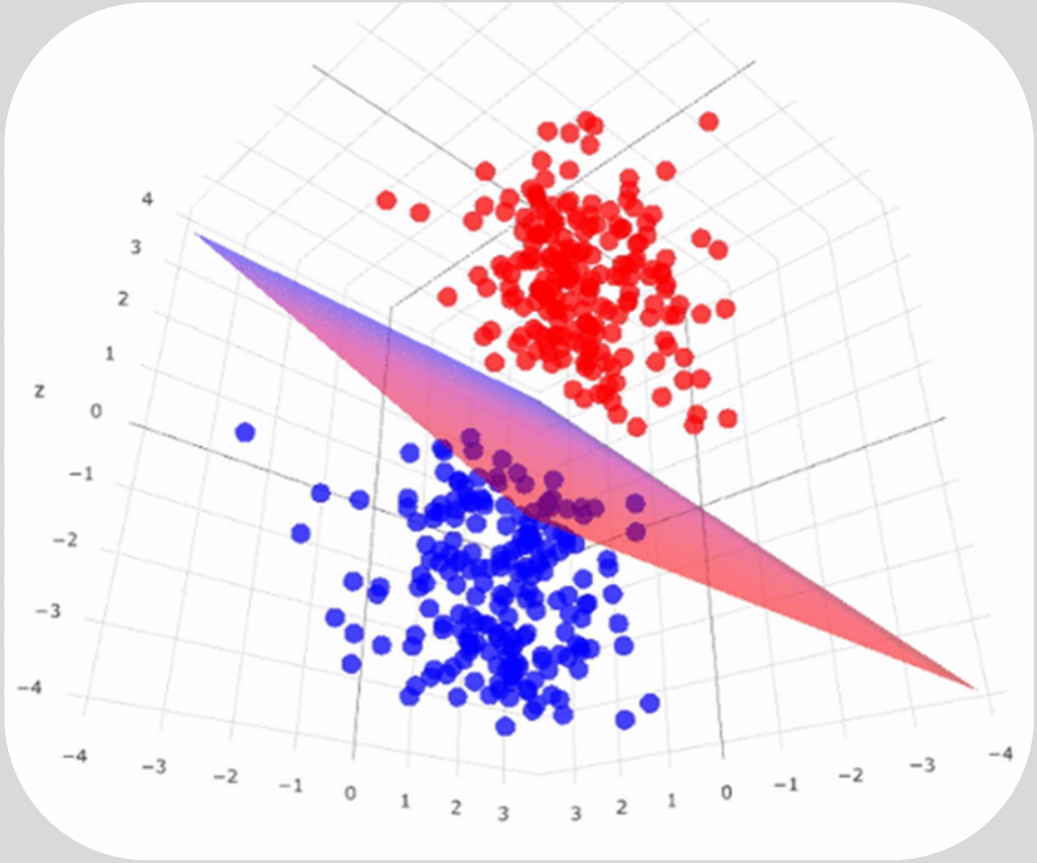
SVM

Cada objeto que se desea clasificar se representa como un punto en un espacio n-dimensional.

El objetivo del algoritmo SVM es encontrar un hiperplano que separe de la mejor forma posible dos clases diferentes de puntos.

El hiperplano es una superficie de dimensión $n - 1$.

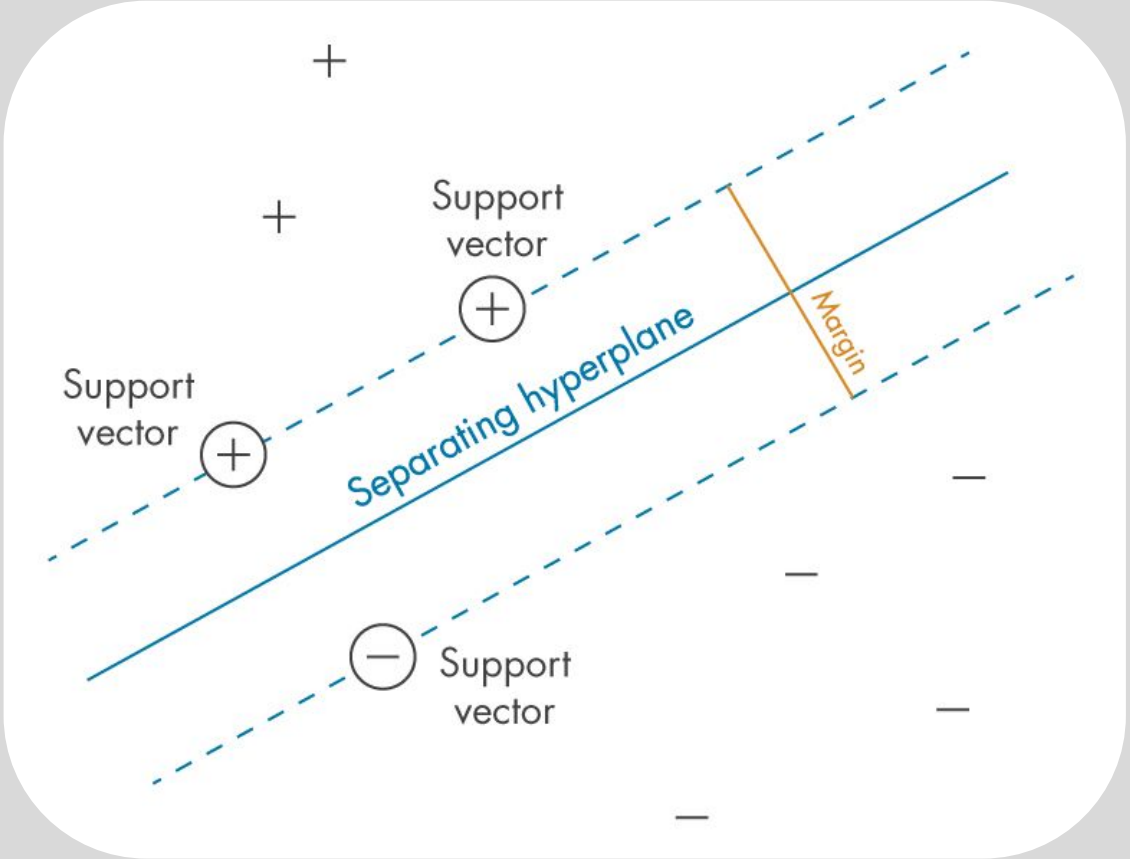
Nota: En 2 dimensiones es una recta y en 3 dimensiones un plano.



SVM

El margen es la distancia entre el hiperplano y el dato más cercano de cada clase, llamados vectores de soporte.

El algoritmo busca maximizar el margen, por lo que la optimización en SVM es esencial.



El problema de optimización

El problema de optimización

Para encontrar el hiperplano, SVM requiere un conjunto de entrenamiento que consiste de n puntos $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, en donde los $\mathbf{x}_i$ son vectores n-dimensionales & $y_i = \pm 1$ indica la clase a la que pertenece $\mathbf{x}_i$.

Un hiperplano se puede describir como el conjunto de puntos $\mathbf{x}$ que satisfacen:

$$\mathbf{w}^T\mathbf{x} - b = 0,$$

en donde $\mathbf{w}$ es un vector normal al hiperplano.

El problema de optimización

Para encontrar el hiperplano, SVM requiere un conjunto de entrenamiento que consiste de n puntos $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, en donde los $\mathbf{x}_i$ son vectores n-dimensionales & $y_i = \pm 1$ indica la clase a la que pertenece $\mathbf{x}_i$.

Recordemos que la forma normal de la ecuación de una recta en $\mathbb{R}^2$ es:

$$\mathbf{n}^T(\mathbf{x} - \mathbf{p}) = \mathbf{n}^T\mathbf{x} - \underbrace{\mathbf{n}^T\mathbf{p}}_{b} = 0$$

Un hiperplano se puede describir como el conjunto de puntos $\mathbf{x}$ que satisfacen: $\mathbf{w} \cdot \mathbf{x} - b = 0$

en donde $\mathbf{w}$ es un vector normal al hiperplano.

El problema de optimización

Para encontrar el hiperplano, SVM requiere un conjunto de entrenamiento que consiste de n puntos $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, en donde los $\mathbf{x}_i$ son vectores n-dimensionales & $y_i = \pm 1$ indica la clase a la que pertenece $\mathbf{x}_i$.

Un hiperplano se puede describir como el conjunto de puntos $\mathbf{x}$ que satisfacen:

$$\mathbf{w}^T\mathbf{x} - b = 0,$$

en donde $\mathbf{w}$ es un vector normal al hiperplano.

El problema consiste en encontrar dos hiperplanos paralelos que separen las dos clases de datos, de forma que la distancia entre ellos sea la mayor posible.

El problema de optimización

Estos hiperplanos pueden describirse mediante las ecuaciones:

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

todo lo que esté en o por encima de este límite pertenece a la clase con la etiqueta 1

y

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

todo lo que esté en o por debajo de este límite pertenece a la clase con la etiqueta -1

El problema de optimización

Geométricamente, la distancia entre los hiperplanos es $\frac{2}{\|\mathbf{w}\|}$, así que para maximizar dicha distancia, se requiere minimizar $\|\mathbf{w}\|$.

Además, se tienen las siguientes restricciones:

$$\mathbf{w}^T\mathbf{x} - b \geq 1, \text{ si } y_i = 1 \;\&\; \mathbf{w}^T\mathbf{x} - b \leq -1, \text{ si } y_i = -1$$

Esto último se puede reescribir como:

$$y_i(\mathbf{w}^T\mathbf{x} - b) \geq 1, \forall i = 1,2,\dots,n$$

El problema de optimización

En conclusión, el problema de optimización en forma estándar es:

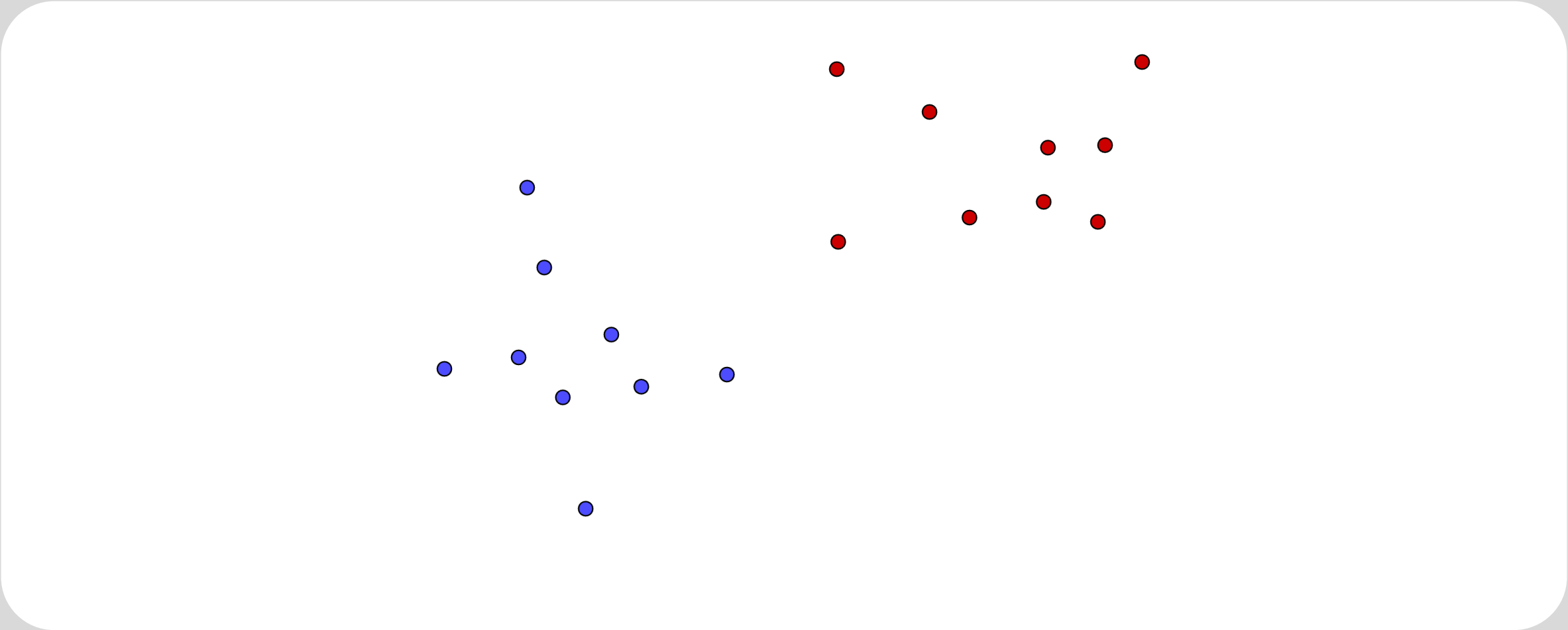
$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 = \min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} = \min_{\mathbf{w},b} \frac{1}{2}\sum_{i=1}^{n} w_i^2$$

sujeto a la restricción:

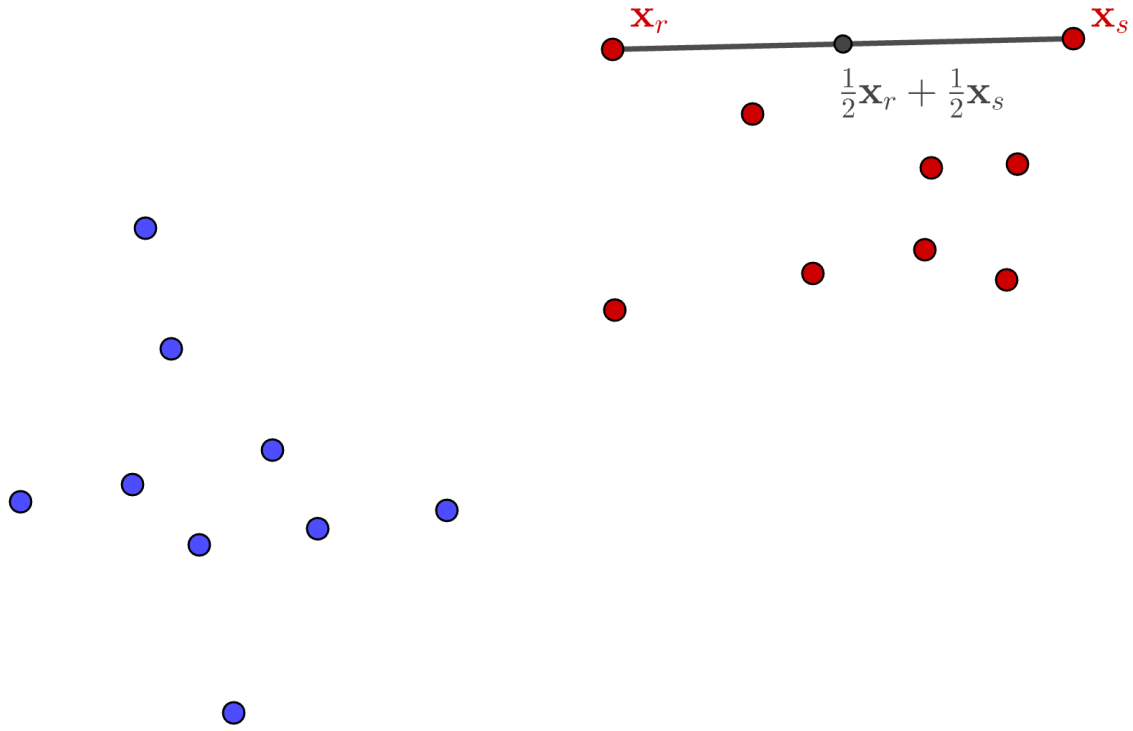
$$y_i(\mathbf{w}^T\mathbf{x} - b) \geq 1, \forall i = 1,2,\dots,n$$

A este tipo de problemas, se les llama **programación cuadrática**.

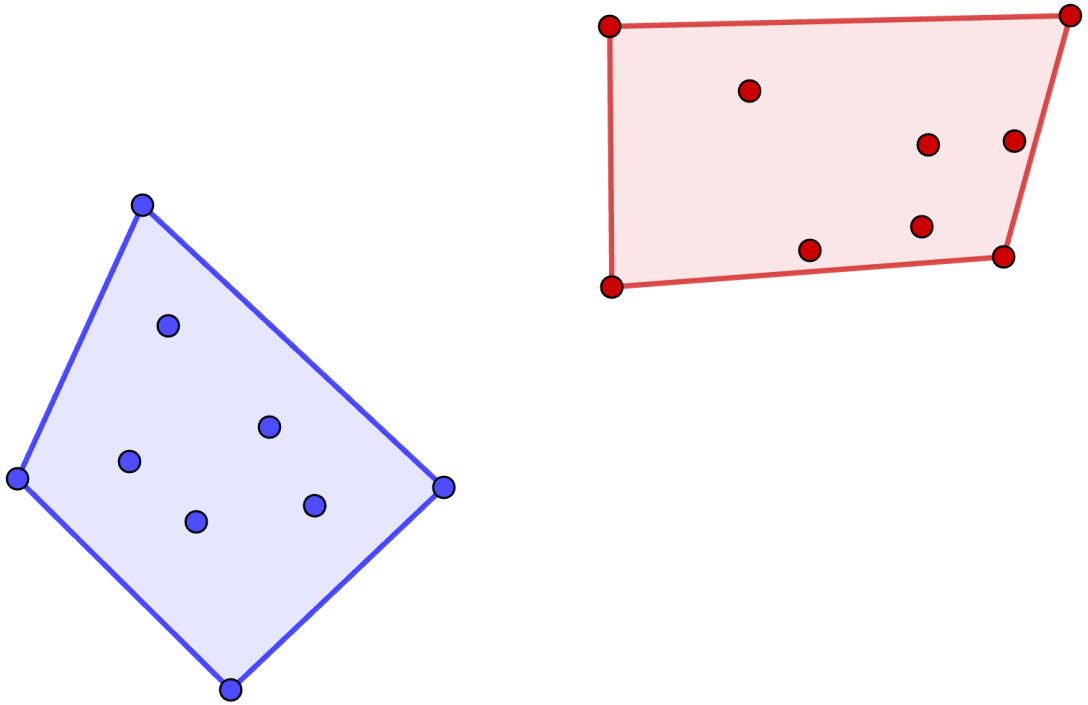
El problema de optimización



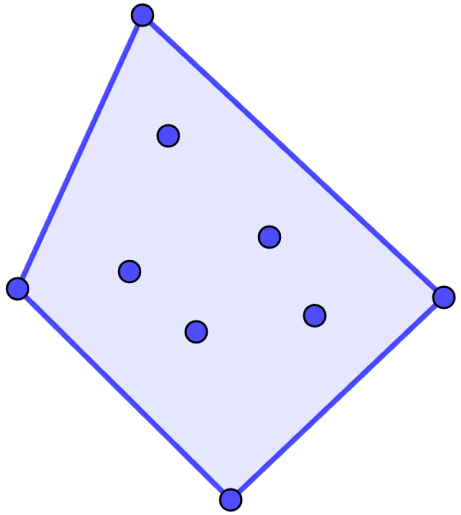
El problema de optimización

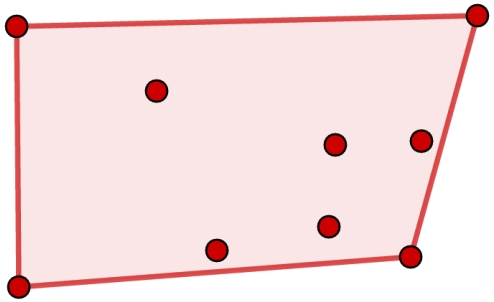


El problema de optimización



El problema de optimización

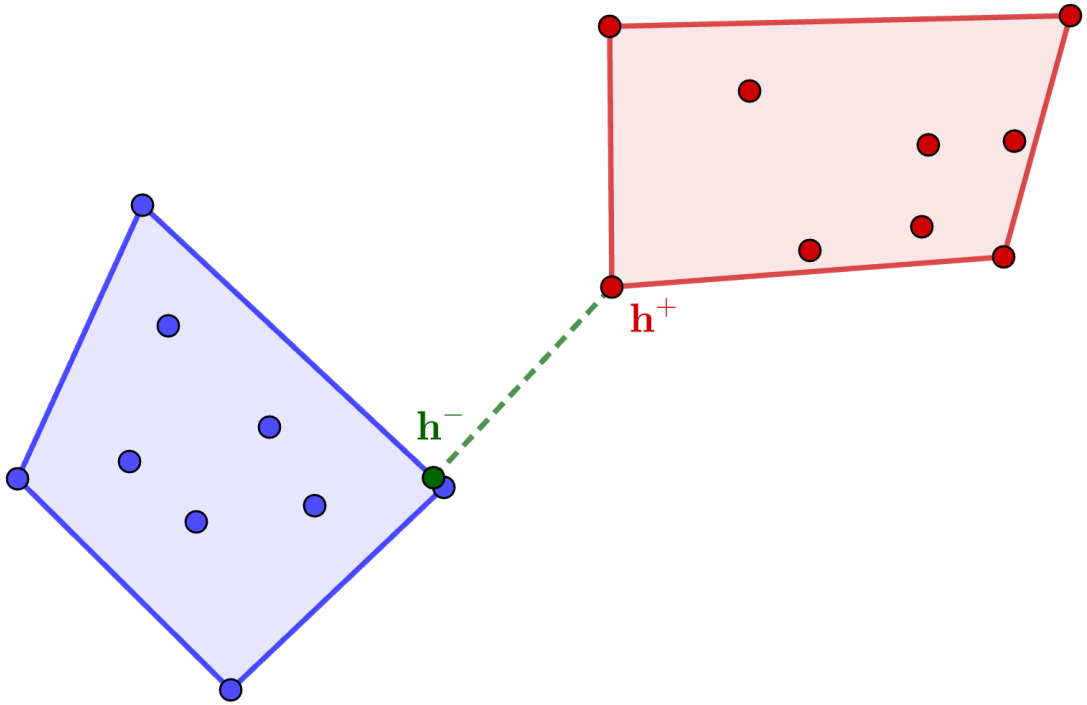




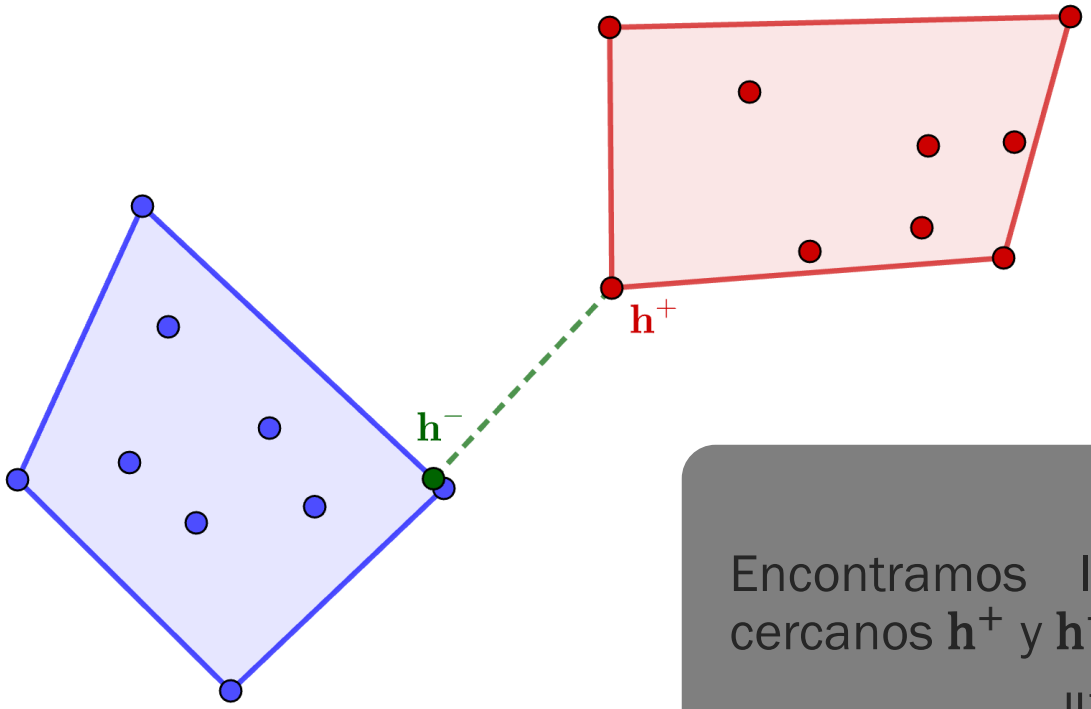
Todo punto $\mathbf{h}$ dentro del polígono es una combinación lineal de los puntos en las esquinas:

$$\mathbf{h} = \sum \alpha_i \mathbf{x}_i$$

El problema de optimización



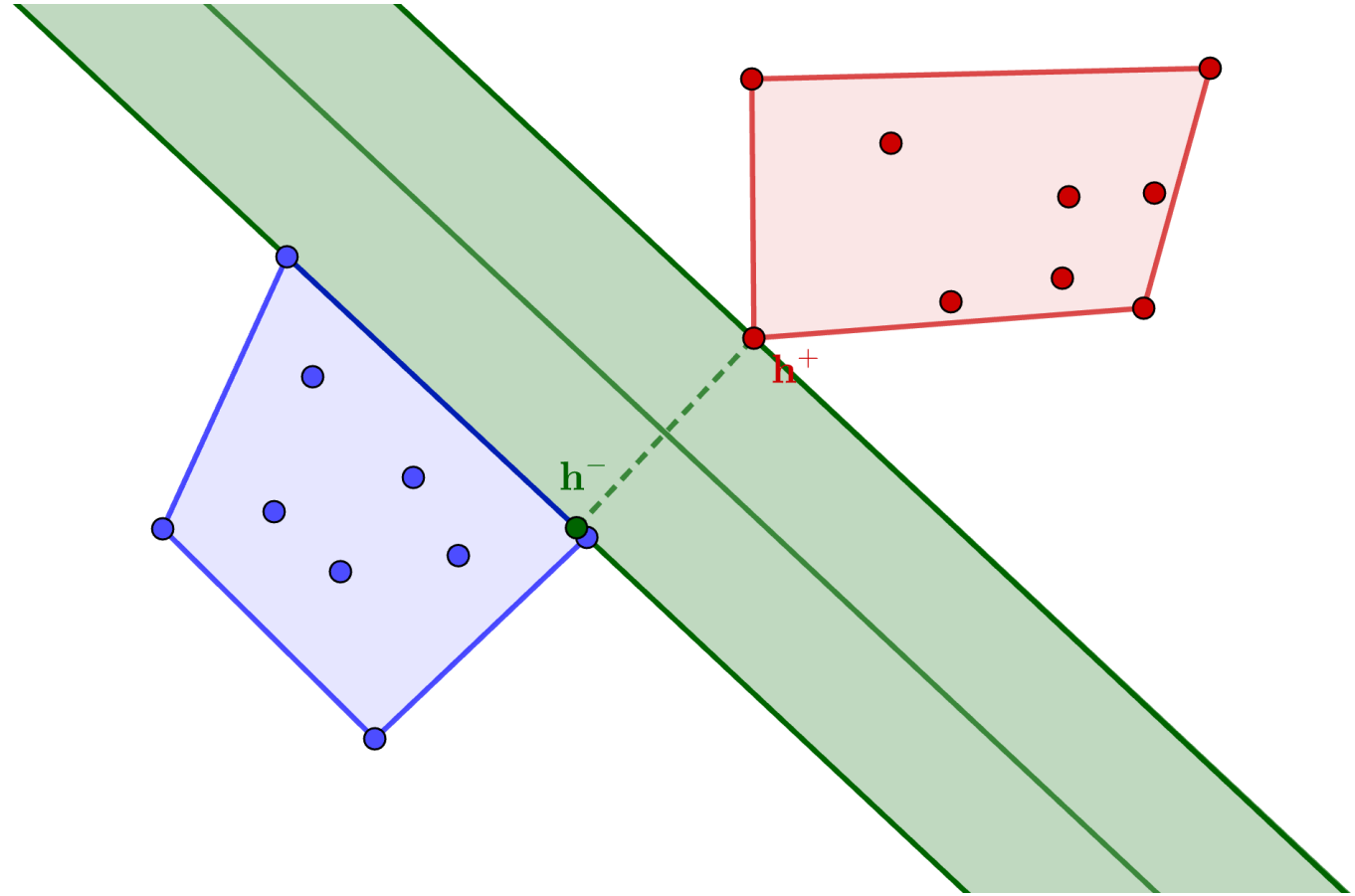
El problema de optimización



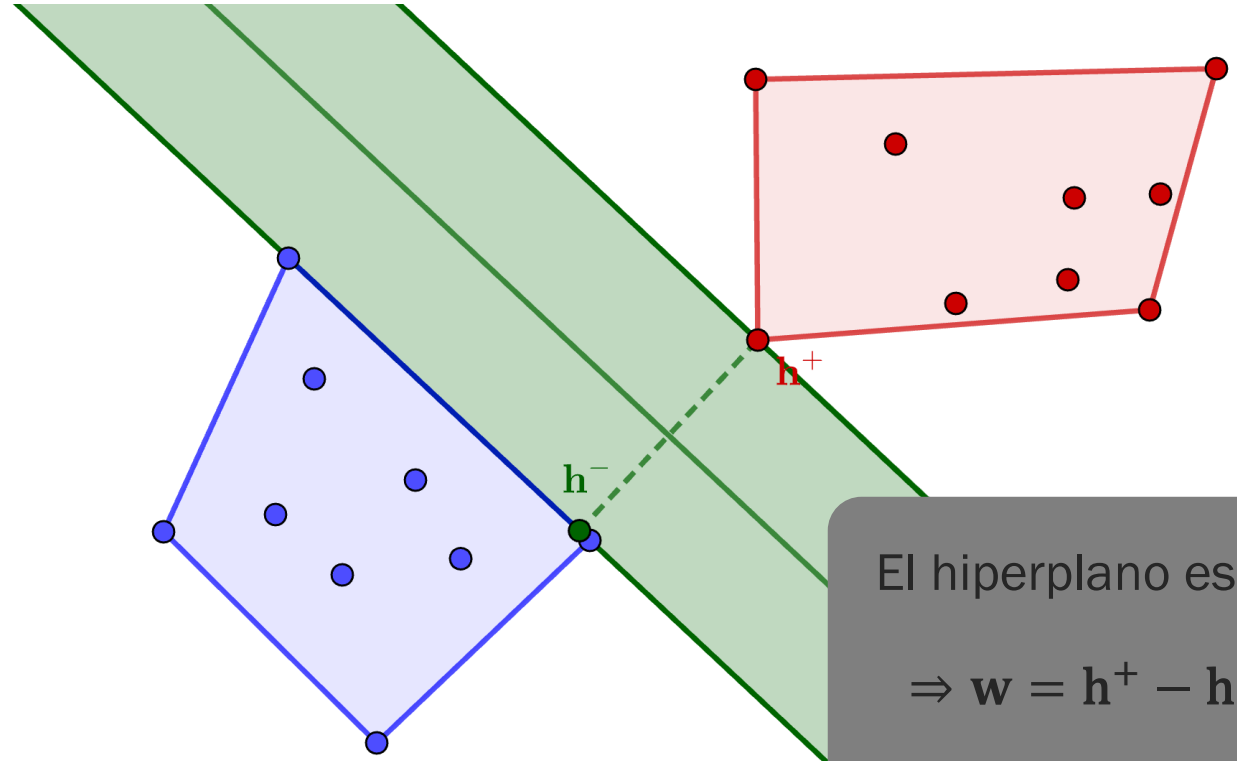
Encontramos los dos puntos más cercanos $\mathbf{h}^+$ y $\mathbf{h}^-$, el margen es:

$$\|\mathbf{h}^+ - \mathbf{h}^-\|$$

El problema de optimización



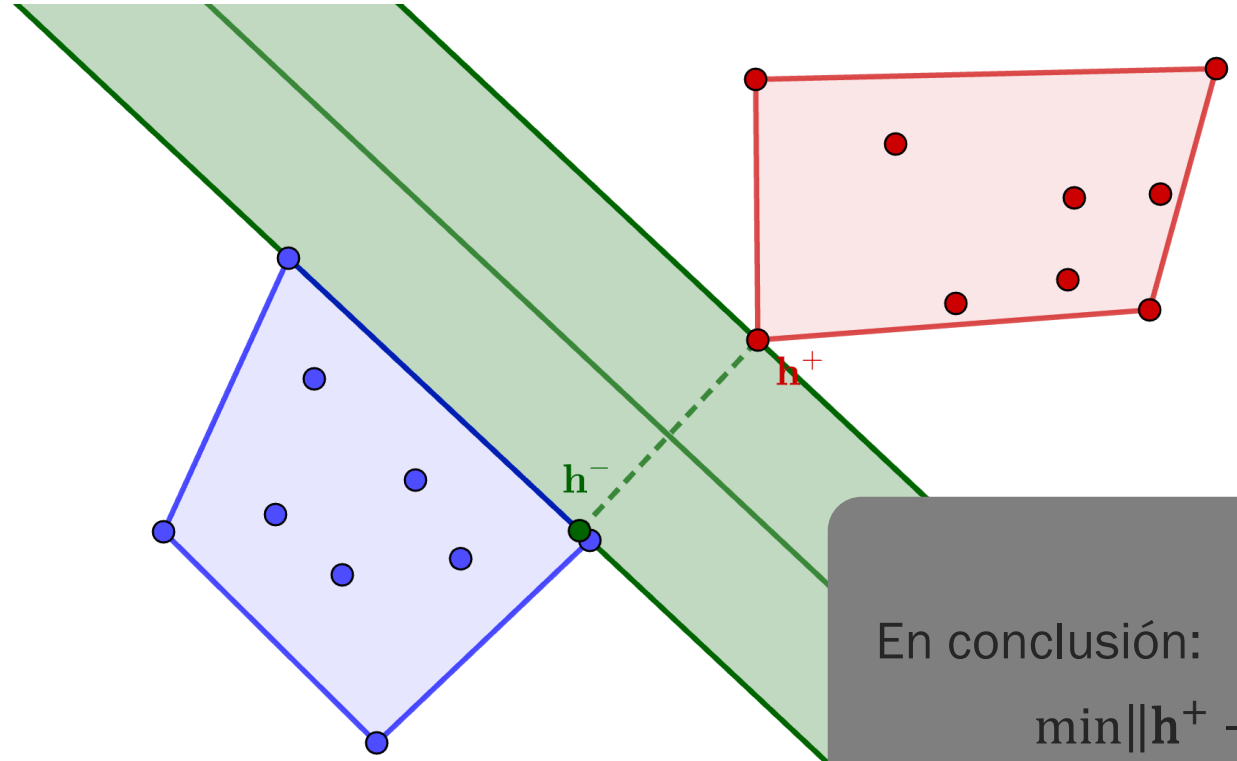
El problema de optimización



El hiperplano es normal a $\mathbf{h}^+ - \mathbf{h}^-$

$$\Rightarrow \mathbf{w} = \mathbf{h}^+ - \mathbf{h}^- = \sum \alpha_i \mathbf{x}_i - \sum \alpha_j \mathbf{x}_j$$

El problema de optimización



En conclusión:

$$\min\|\mathbf{h}^+ - \mathbf{h}^-\| = \min\|\mathbf{w}\|$$

Algoritmo de optimización

Algunos algoritmos para programación cuadrática

- **Punto interior:**

  Optimiza iterativamente dentro del conjunto factible, evitando los bordes de las restricciones.

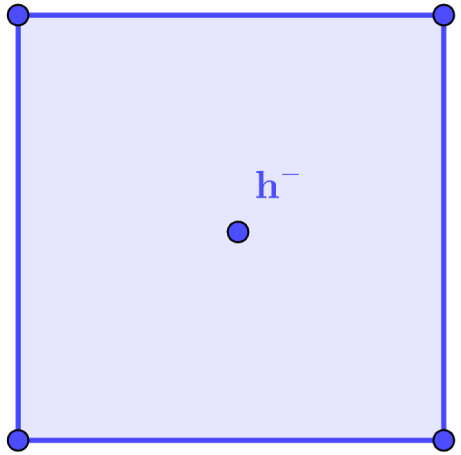
- **Método del Lagrangiano Aumentado:**

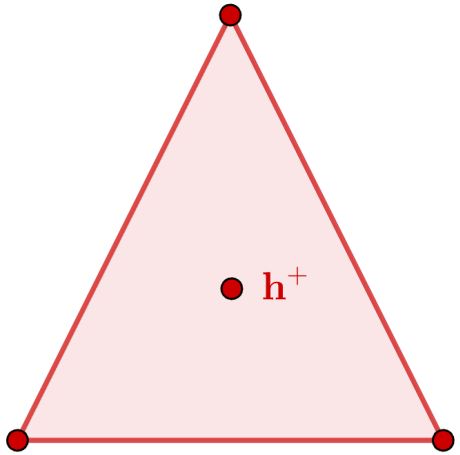
  Utiliza multiplicadores de Lagrange, e introduce un término penalizador en la función objetivo para penalizar violaciones de las restricciones.

Algoritmo SMO

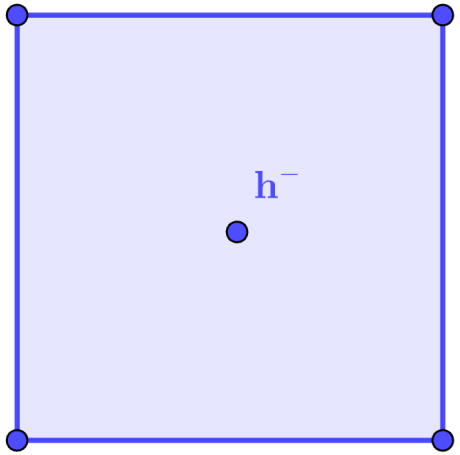
La optimización mínima secuencial – SMO es un algoritmo para resolver el problema de programación cuadrática que surge durante el entrenamiento de SVM.

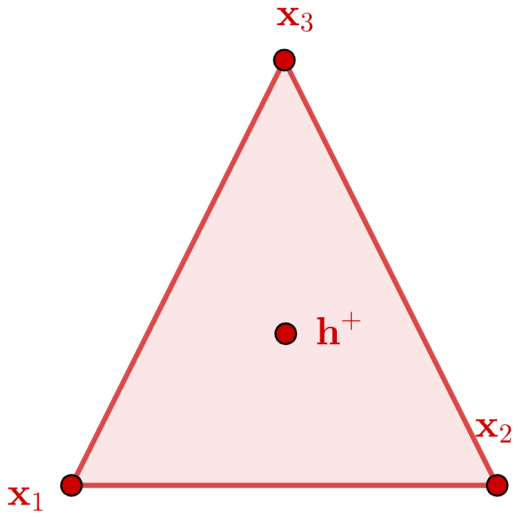
Algoritmo SMO





Algoritmo SMO

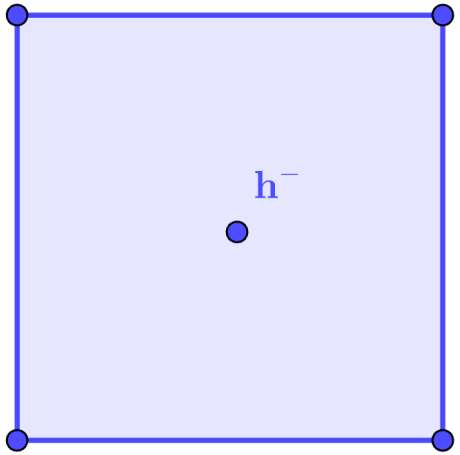


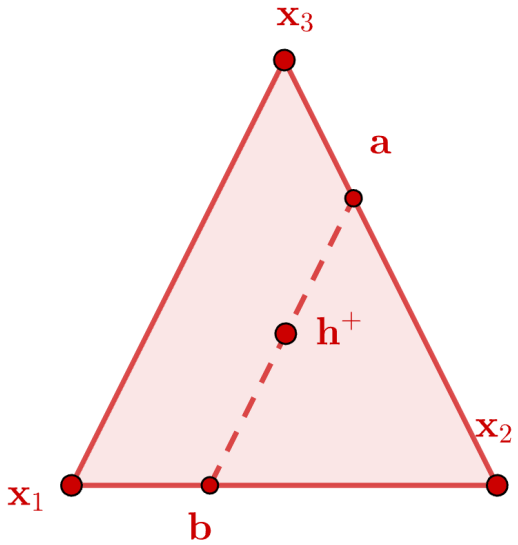


Se comienza con una elección de $\alpha_i = \frac{1}{3}$:

$$\mathbf{h}^+ = \frac{1}{3}\mathbf{x}_1 + \frac{1}{3}\mathbf{x}_2 + \frac{1}{3}\mathbf{x}_3$$

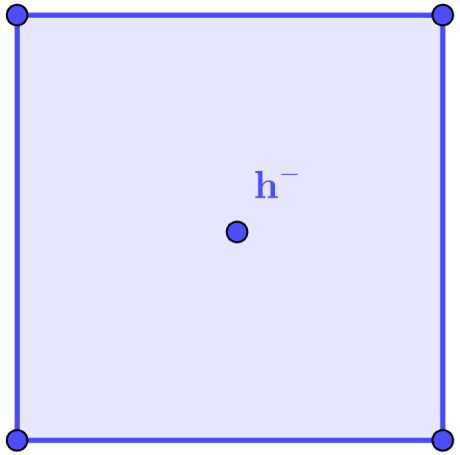
Algoritmo SMO

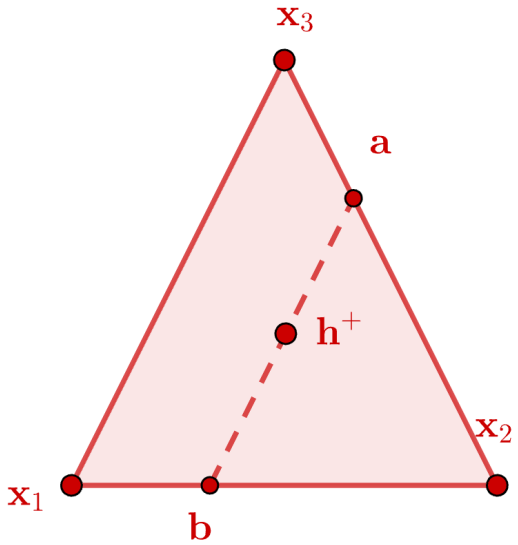




Se escogen dos α, por ejemplo, α_1 y α_3, y se fija α_2.

Algoritmo SMO

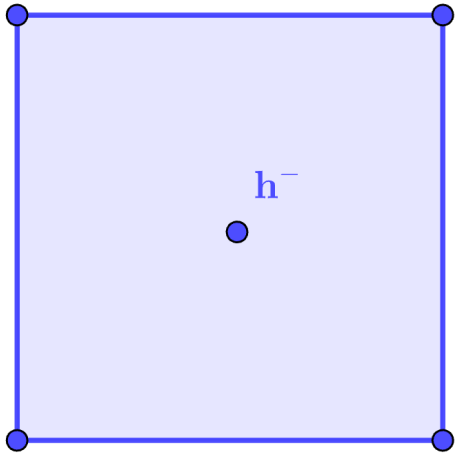


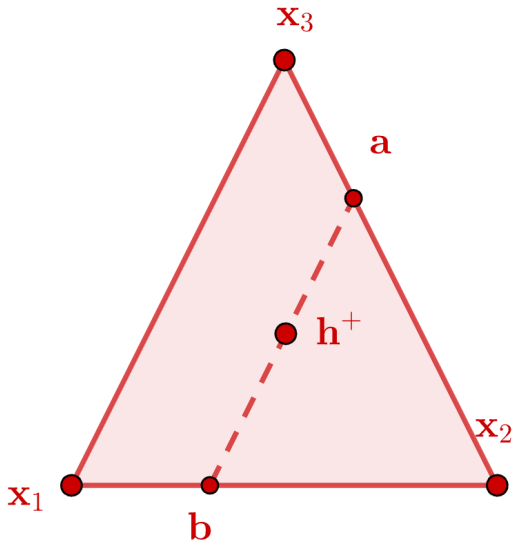


Entonces todos los puntos sobre el segmento que une $\mathbf{a}$ y $\mathbf{b}$:

$$\mathbf{x} = \mathbf{b} + t(\mathbf{a} - \mathbf{b})$$

Algoritmo SMO





Se calcula el t que minimiza:

$$\|\mathbf{b} + t(\mathbf{a} - \mathbf{b}) - \mathbf{h}^-\|$$

Algoritmo SMO

Como $\min\|\mathbf{x}\| = \min \frac{1}{2}\|\mathbf{x}\|^2$, se hace $\Delta = \frac{1}{2}\sum_i \ (b_i + t(a_i - b_i) - h_i^-)^2$, y entonces:

$$\frac{\partial \Delta}{\partial t} = \sum_i (b_i + t(a_i - b_i) - h_i^-)(a_i - b_i) = \sum_i \left(t(a_i - b_i)^2 + (b_i - h_i^-)(a_i - b_i)\right)$$

Al hacer $\frac{\partial \Delta}{\partial t} = 0$ y despejar para t se obtiene:

$$t = \frac{(\mathbf{a} - \mathbf{b})^T(\mathbf{h}^- - \mathbf{b})}{\|\mathbf{a} - \mathbf{b}\|^2}$$

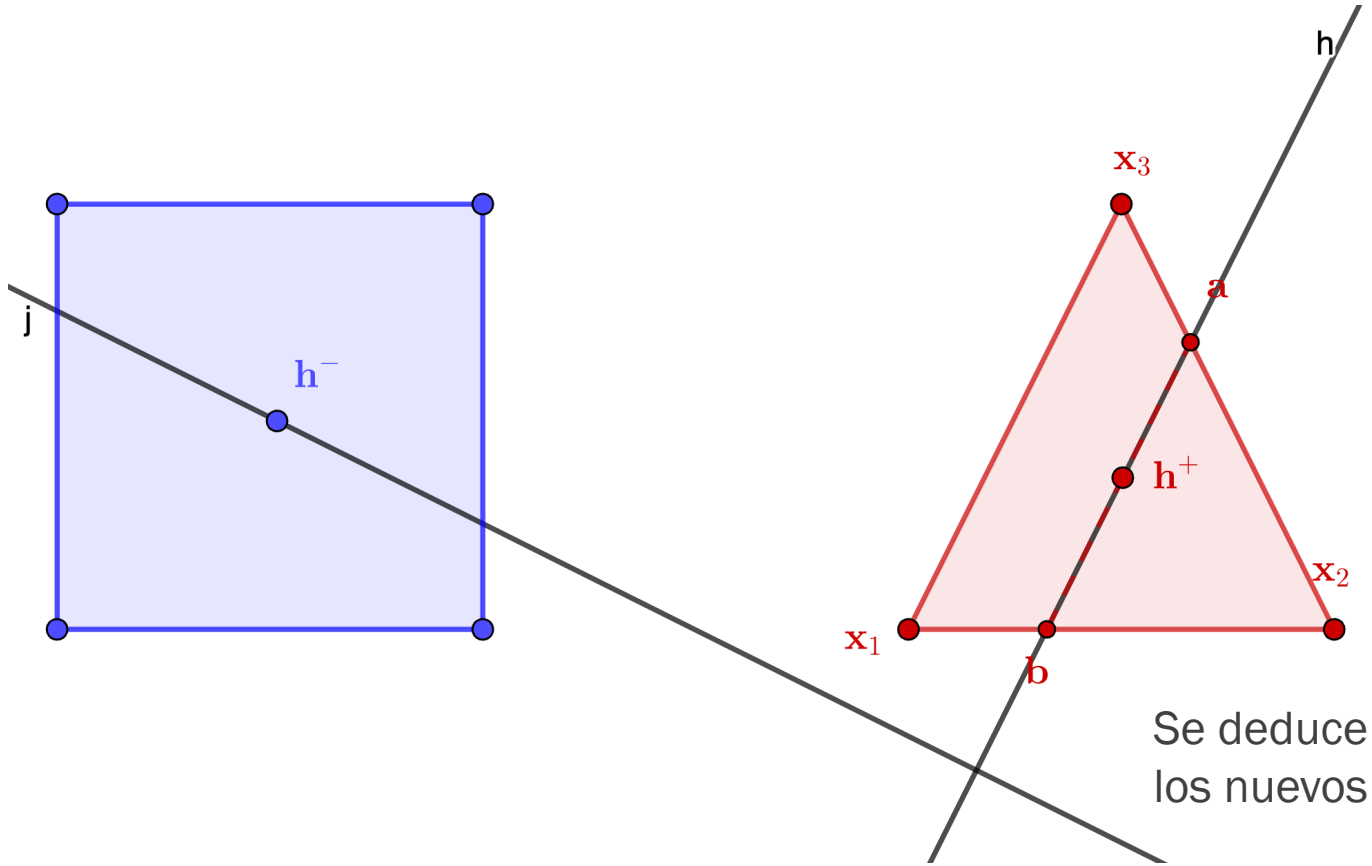
Algoritmo SMO

Se actualizan los α:

$$\alpha_1 = \begin{cases} 0, & t \leq 0 \\ \frac{2}{3}t, & t \in (0,1) \\ \frac{2}{3}, & t \geq 1 \end{cases} \ \& \ \alpha_3 = \frac{2}{3} - \alpha_1$$

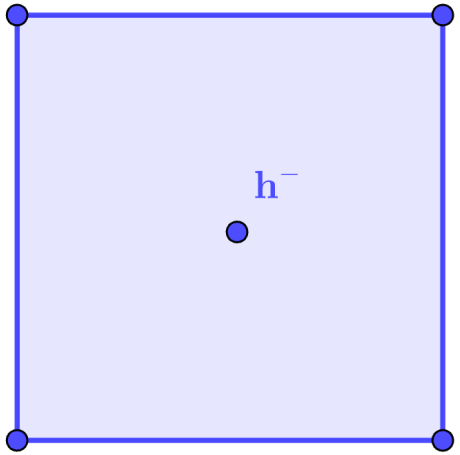
El algoritmo funciona de forma iterativa, actualizando únicamente dos variables a la vez, lo cual simplifica el problema considerablemente y reduce el tiempo de cálculo.

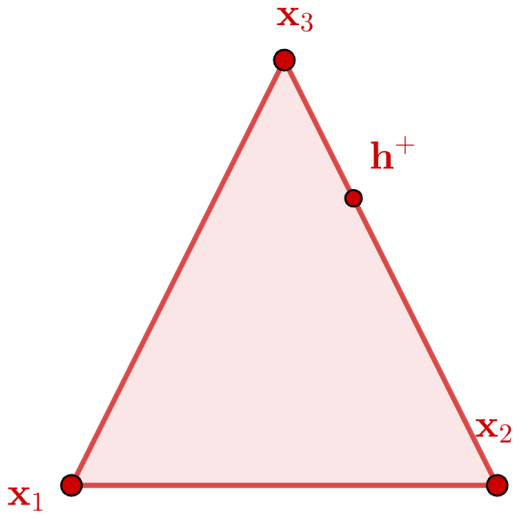
Algoritmo SMO



Se deduce que $t < 0$, por lo que los nuevos $\alpha_1 = 0$ y $\alpha_3 = {}^{2}/_{3}$

Algoritmo SMO





En conclusión:

$$h^+ = 0\mathbf{x}_1 + \frac{1}{3}\mathbf{x}_2 + \frac{2}{3}\mathbf{x}_3$$

Algoritmo SMO

El algoritmo SMO evita cálculos de matriz de alta dimensión, lo que reduce el costo computacional.

Debido a su enfoque en subproblemas de dos variables, el algoritmo puede implementarse sin depender de métodos complejos de optimización.

Su estructura iterativa lo hace ideal para conjuntos de datos grandes, donde los métodos tradicionales de optimización cuadrática serían demasiado costosos.

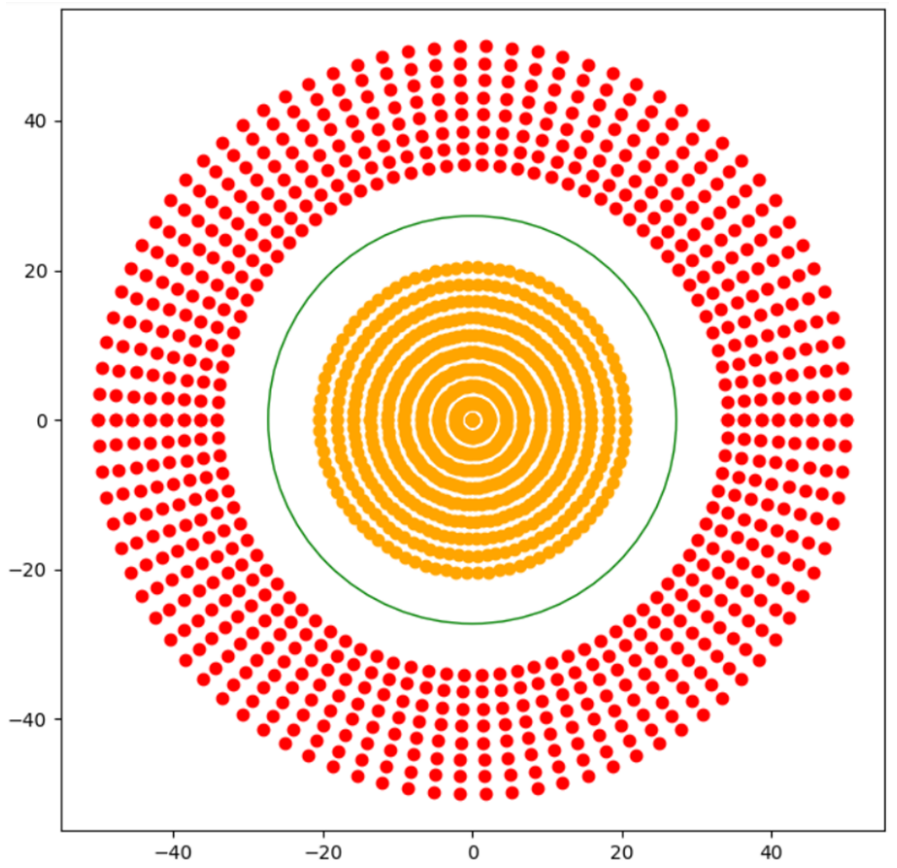
Kernel

Kernel

Cuando no es posible lograr la clasificación con un hiperplano lineal, los datos de entrada se transforman en un espacio de características de mayor dimensión, donde puede ser más fácil encontrar una frontera de decisión lineal que separe las clases.

Kernel

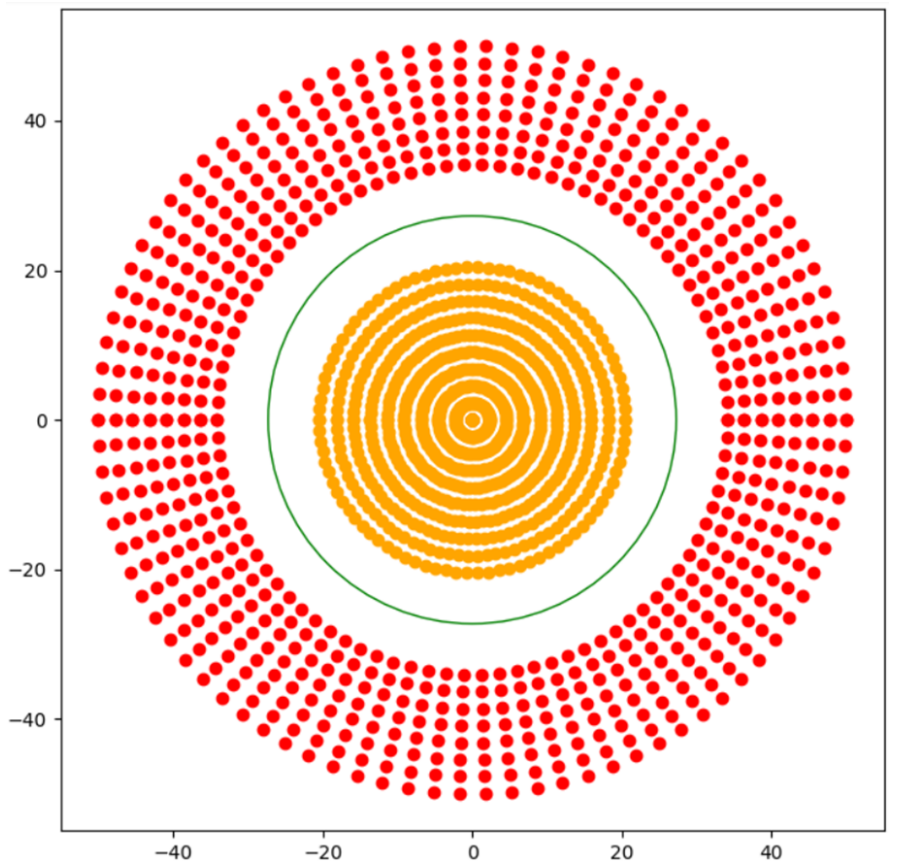
Suponga que se tienen dos características X e Y, y los datos no son linealmente clasificables, como los que se muestran en la figura.



Kernel

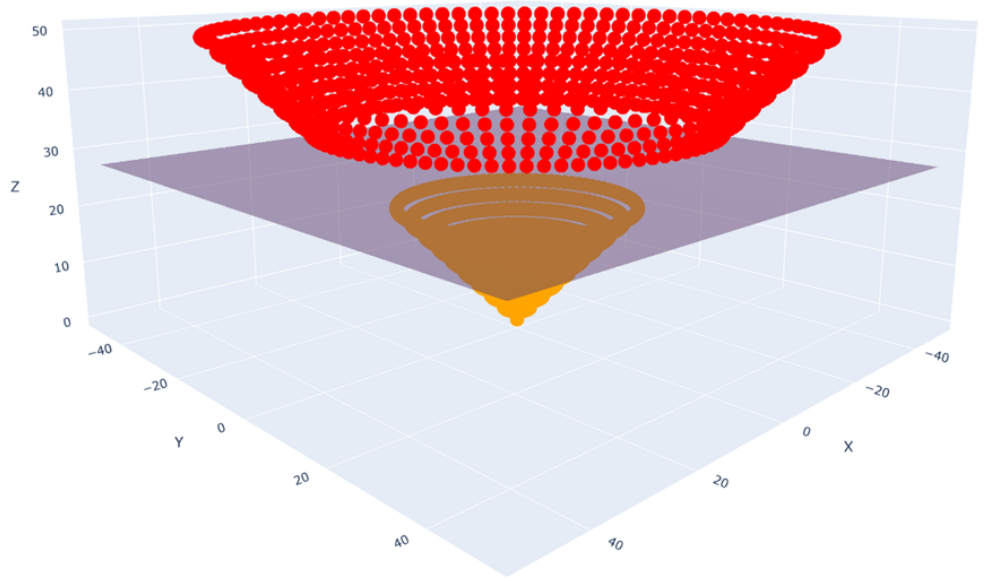
Lo que se tiene que hacer es añadir otra dimensión en la que los datos se vuelven linealmente separables.

La idea es usar una función kernel para transformar las características de los datos $Z = f(X, Y)$.



Kernel

Si $f(X,Y) = X^2 + Y^2$, entonces los datos de la forma (X,Y,Z) ahora son linealmente clasificables.



Tipos de Kernel

Las funciones de kernel asignan los datos a un espacio dimensional diferente, que suele ser superior, con el objetivo de que resulte más fácil separar las clases después de esta transformación. Entre los más conocidos están:

Lineal	$K(x_1, x_2) = x_1^T x_2$	Datos linealmente separables
RBF	$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$	Datos no lineales con relaciones complejas.
Polinómica	$K(x_1, x_2) = (x_1^T x_2 + 1)^p$	Datos con relaciones polinómicas.
Sigmoide	$K(x_1, x_2) = \tanh(\beta_0 x_1^T x_2 + \beta_1)$	Datos que modelan comportamientos similares a redes neuronales o funciones sigmoides.

¿Preguntas?