

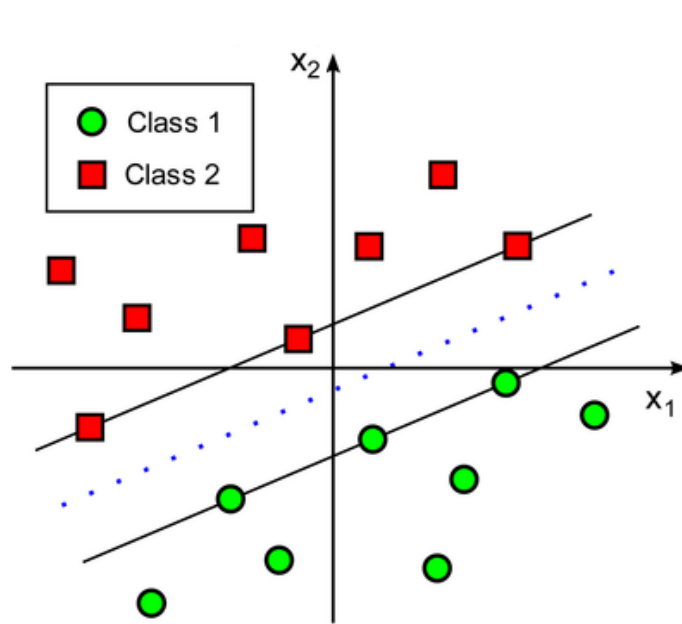
# **Máquinas de soporte vectorial**

Dr. Raimundo Sánchez  
raimundo.sanchez@uai.cl  
@raimun2

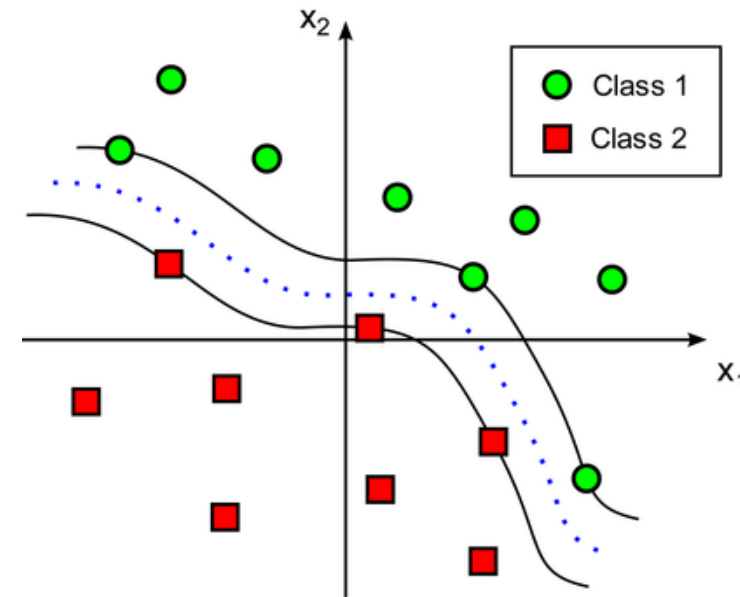
# Support Vector Machines (SVM)

SVM es un modelo discriminativo que busca la mejor manera de dividir un espacio, basándose en algunos puntos de este (support vectors), para separarlo en 2 zonas con clases distintas.

Luego se construye un clasificador sobre esa división.

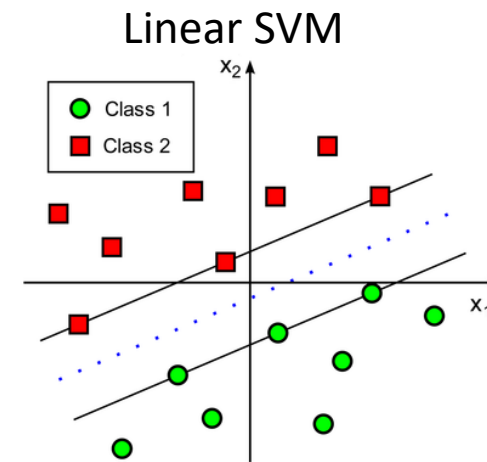
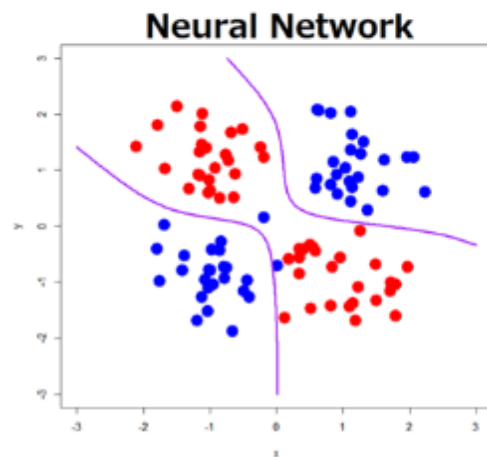
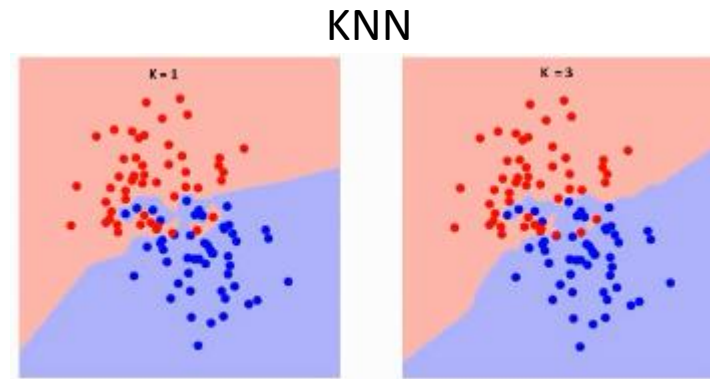
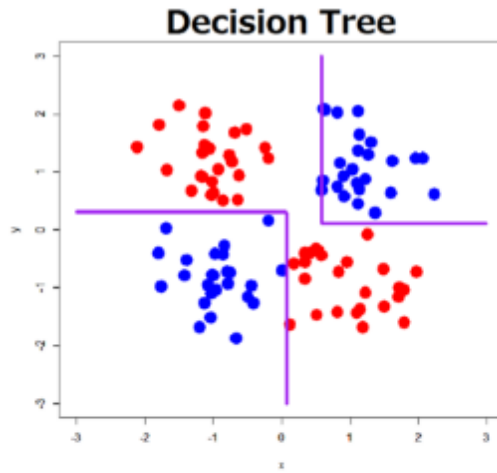


Linear SVM



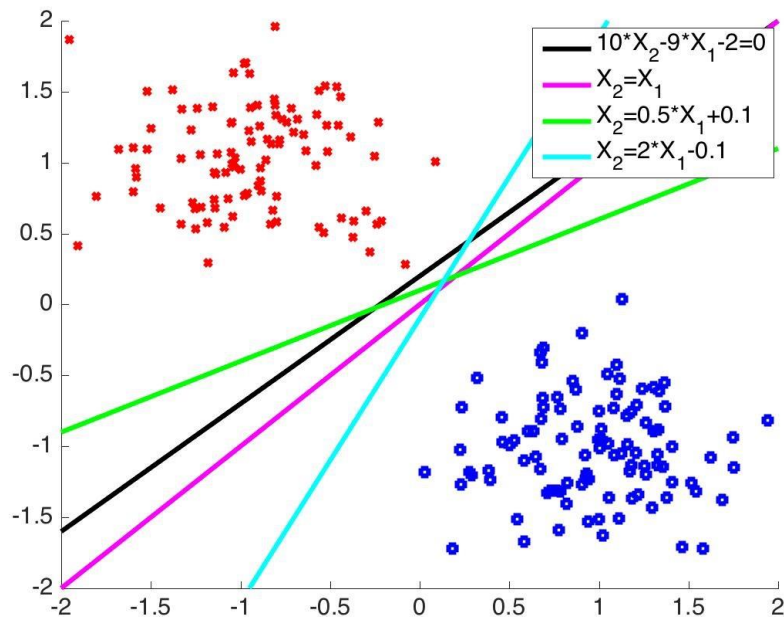
Non linear SVM

# Comparación con otros métodos



# Descripción

Para separar el espacio nosotros tenemos que buscar un hiperplano (de una dimensión menor a las dimensiones del espacio). En un espacio 2D, como en el próximo ejemplo, el hiperplano será una línea.



Cuando llegue un nuevo punto, podemos evaluar a que lado del hiperplano esta. Dependiendo de a que lado este, lo podemos clasificar (como rojo o azul en este caso).

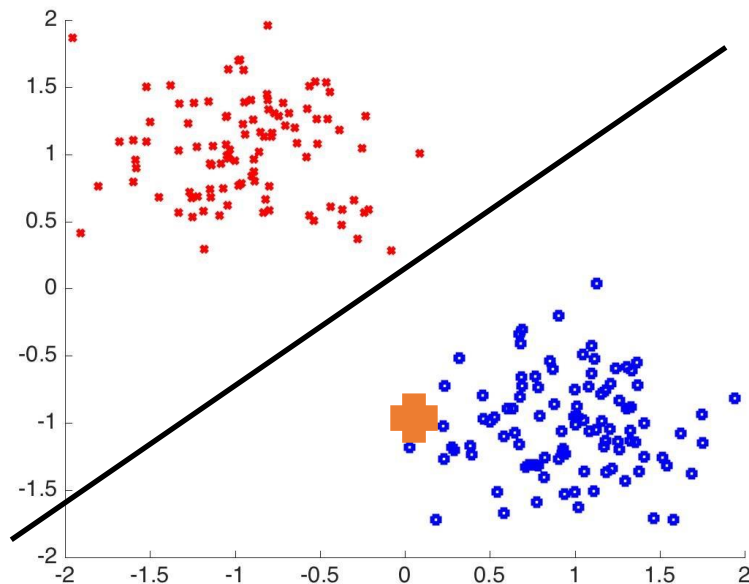
- El problema es que muchos hiperplanos logran dividir el espacio en 2. ¿Cuál será el mejor? ¿Cómo lo logramos encontrar? ¿Qué factores afectan su posición?

# Ejemplo

Supongamos que hemos ya terminado con SVM y el mejor hiperplano esta dado por la recta:

$$-9x + 10y - 2 = 0$$

Nos llega un nuevo punto en (0,-1), el cual deseamos clasificar.



Si evaluamos:

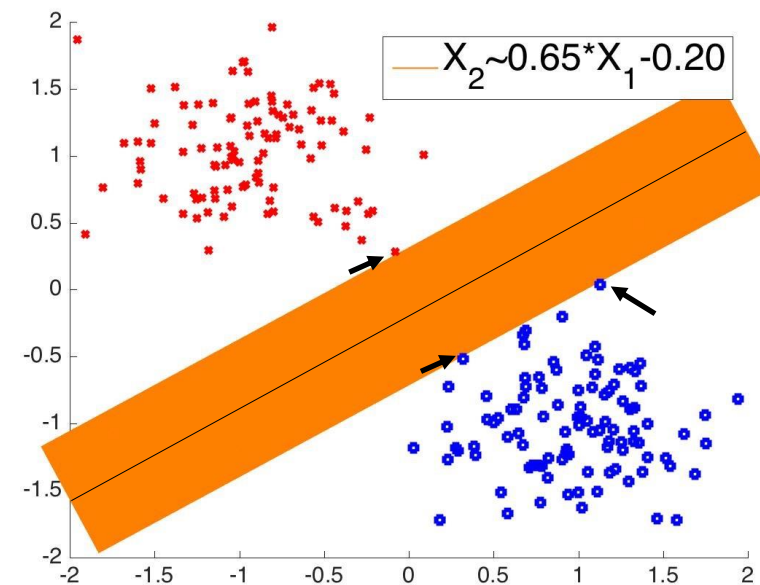
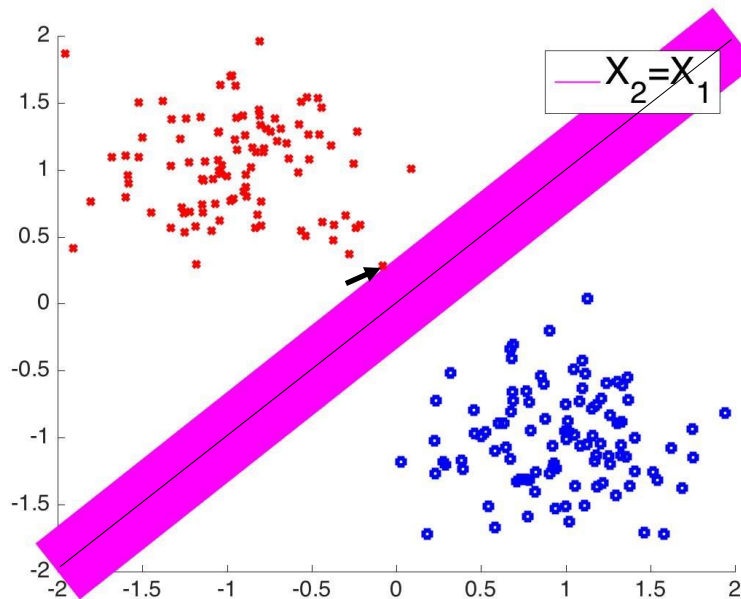
$-9*(0) + 10*(-1) - 2 \Rightarrow -13$ ; como el resultado es negativo, el nuevo punto debe corresponder a la parte inferior de la recta, o sea, la clase azul.

Si fuera un resultado positivo, seria de clase roja.

# Frontera de decisión

A cada posible hiperplano que cumpla las condiciones, le dibujaremos un margen equidistante a ambos lados hasta tocar el primer punto del espacio.

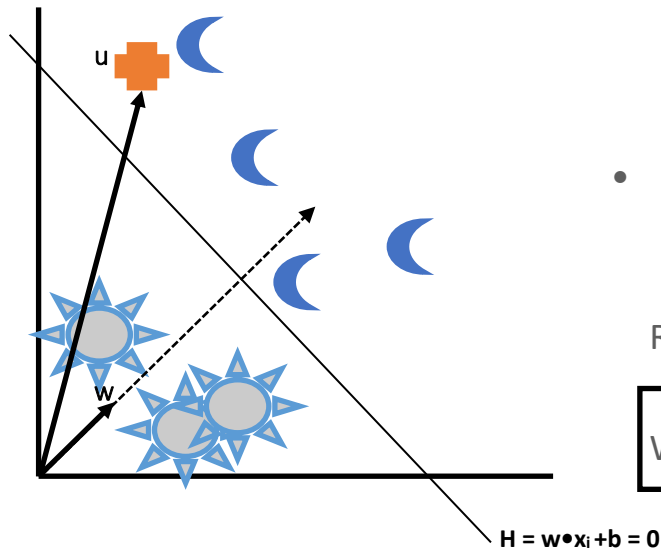
Diremos que el mejor hiperplano es aquel que tenga el **margen mas grande posible**. Esto lo podemos denominar el método del “ancho de avenida”.



- Entre el hiperplano representado por la ecuación de la recta de color naranja tiene un margen mas grande que el rosado; por lo tanto esta dividiendo el espacio de mejor forma. Los “vectores de soporte” son los puntos que tienen contado con los márgenes del hiperplano.

# Aprendizaje

Supongamos un espacio con algunos pocos puntos (entidades) que poseen 2 clases distintas (Lunas y Soles). Vamos a definir matemáticamente la regla de decisión que impone el supuesto hiperplano.



- $W$  es un vector perpendicular al hiperplano.  $U$  es el vector de un punto desconocido que queremos clasificar.  $b$  es una constante (bias o umbral) que escala el vector proyectado.
- Si proyectamos  $U$  sobre  $W$  podemos luego observar “a que lado de la avenida se encuentra”.

Regla de decisión:

$$W \bullet U + b \geq 0 \rightarrow \text{Entonces } U \text{ es de clase + (sobre hiperplano)(Luna)}$$

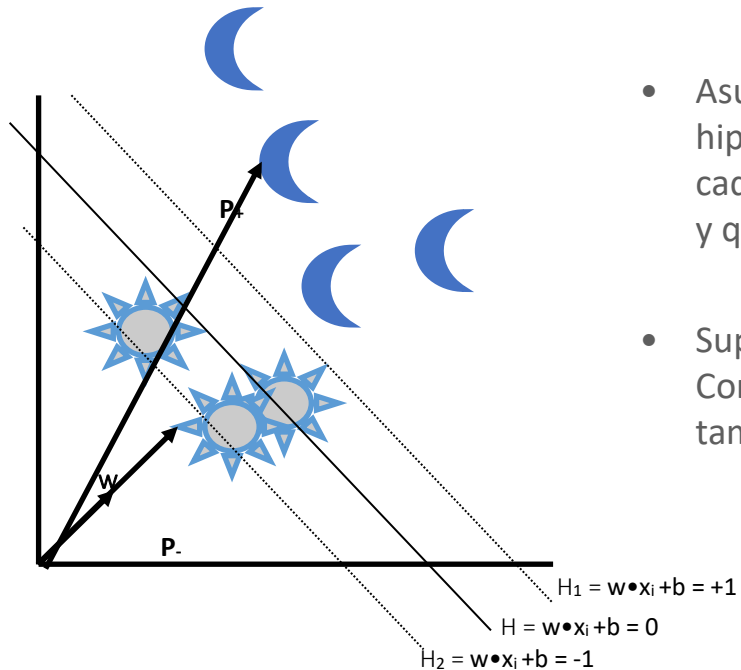
Tenemos que buscar el  $W$  y  $b$  que definan el mejor hiperplano.

Existen muchos  $W$  y  $b$  que cumplen con los requerimientos ¿Cuál es mejor?  $\rightarrow$  Márgenes

$W$  ajusta el “ángulo de inclinación” del hiperplano.  $b$  mueve el hiperplano desde el punto  $0,0$  a otra posición del espacio (afecta el umbral de decisión)

# Aprendizaje

Supongamos un espacio con algunos pocos puntos (entidades) que poseen 2 clases distintas (Lunas y Soles). Vamos a definir matemáticamente la regla de decisión que impone el supuesto hiperplano.



- Asumamos que hemos encontrado los mejores márgenes para este hiperplano y se encuentran a una unidad de distancia (del eje X) a cada lado. Cumplen que son equidistantes del centro del hiperplano y que no sobrepasan ningún punto.
- Supongamos un punto de clase Luna (clase positiva)  $P_+$ . Consideremos también un punto clase Sol (clase negativa)  $P_-$ . Y también los márgenes que hemos creado.

Con la regla de decisión anterior y usando los márgenes, crearemos una restricción basándonos en los puntos que tenemos.

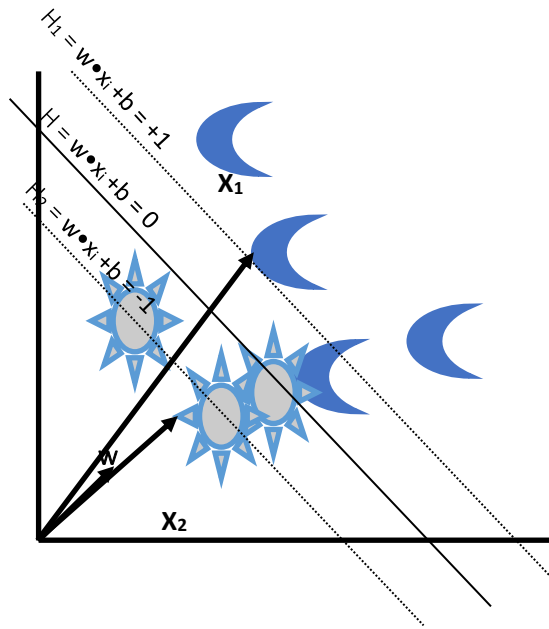
$w \cdot P_+ + b \geq 1 \rightarrow$  Nuestros puntos Luna “deben estar al lado superior de la calle”

$w \cdot P_- + b \leq -1 \rightarrow$  Nuestros puntos Sol “deben estar al lado inferior de la calle”



# Aprendizaje

Supongamos un espacio con algunos pocos puntos (entidades) que poseen 2 clases distintas (Lunas y Soles). Vamos a definir matemáticamente la regla de decisión que impone el supuesto hiperplano.



- Creemos una nueva variable  $y_i$  la cual será la clase de la entidad  $i$ . Definiremos que  $y$  vale +1 si la clase es positiva (Luna) y -1 si la clase es negativa (Sol).
- Definamos  $x_i$  como el vector de posición de la entidad  $i$

Si ajustamos la restricción anterior:

$$1(w \bullet x_i + b) \geq 1 \text{ si } y_i \text{ es } +1 \text{ (Luna)}$$

$$-1(w \bullet x_i + b) \leq -1 \text{ si } y_i \text{ es } -1 \text{ (Sol)} \rightarrow 1(w \bullet x_i + b) \geq 1$$

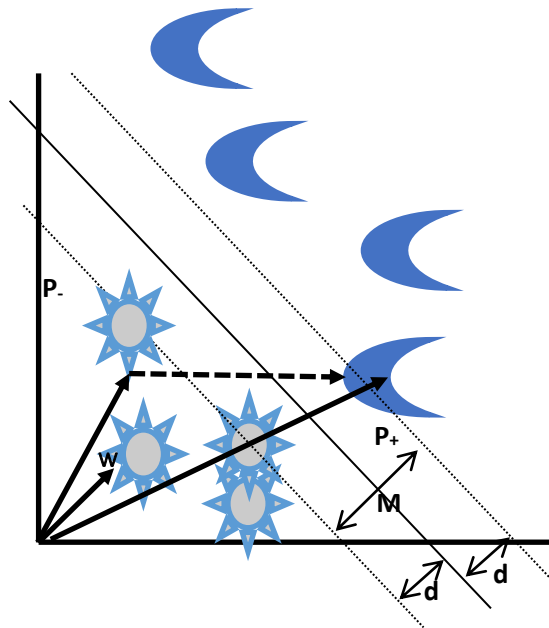
Restricción del espacio, independiente de la clase; gracias a usar  $y_i$

$$y_i(w \bullet x_i + b) - 1 \geq 0$$

$y_i(w \bullet x_i + b) - 1 = 0$  en caso de puntos que esta justo sobre algún margen

# Aprendizaje

Ahora que tenemos la regla de decisión y la restricción, vamos a ver como se comporta el ancho del hiperplano.



Vector entre dos  
puntos que están en  
márgenes opuestos

$$M = \text{Ancho de margen} = (P_+ - P_-) \cdot \underbrace{(W / ||w||)}_{\text{Vector unitario perpendicular a hiperplano}}$$

Vector unitario  
perpendicular a hiperplano

- Si reemplazamos  $P_+$  y  $P_-$  con las restricciones anteriores.

$$P_+ \Leftrightarrow 1(w \cdot x_i + b) - 1 = 0 \quad \text{y} \quad P_- \Leftrightarrow -1(w \cdot x_i + b) - 1 = 0$$

$$M = \text{Ancho de margen} = 2 / ||w||$$

El ancho del margen es 2 dividido en la magnitud de  $W$ , lo anterior se cumple si consideramos la posición de los márgenes dados y el hecho de incluir la variable  $y_i$  específicamente con los valores  $+1$  y  $-1$  para las dos clases a dividir.

$$\text{MAX } 2 / ||w|| \Leftrightarrow \text{MIN } ||w|| \Leftrightarrow \text{MIN } (1/2) * ||w||^2$$

# Aprendizaje

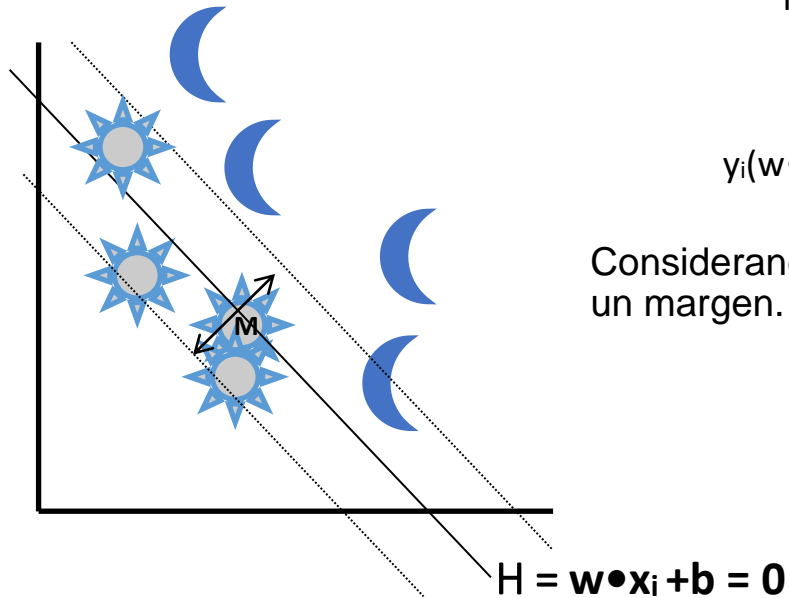
Debemos encontrar un  $\mathbf{w}$  y  $\mathbf{b}$  que nos brinde el hiperplano con el  $M$  mas grande dado los puntos del espacio y sus clases:

$$\text{Max } 2 / ||\mathbf{w}|| \Leftrightarrow \text{Min } ||\mathbf{w}|| \Leftrightarrow \text{Min } (1/2) * ||\mathbf{w}'||^2$$

Sujeto a:

$$y_i(\mathbf{w} \bullet \mathbf{x}_i + b) - 1 \geq 0 \text{ para todos los puntos } i, \text{ con posición } \mathbf{x} \text{ y clase } y$$

Considerando que:  $y_i(\mathbf{w} \bullet \mathbf{x}_i + b) - 1 = 0$  en caso de que el punto  $i$  este sobre un margen.



$$\begin{aligned} &\text{minimize: } \frac{1}{2} ||\mathbf{w}'||^2 \\ &\text{subject to: } y_i * (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

# Optimización

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{2} \|\mathbf{w}^2\| \\ \text{subject to:} \quad & y_i * (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

El problema es de minimización de tipo convexo cuadrático, con muchas restricciones lineales de desigualdades. Con  $p+1$  parámetros ( $p$  es la dimensionalidad de la data) y  $n$  restricciones (una restricción por cada entidad).

En este caso podemos usar el método de optimización de Lagrange para maximizar una nueva función y no preocuparnos de las restricciones. Las restricciones serán reemplazadas por multiplicadores de Lagrange y el proceso de aprendizaje será dictado por productos puntos.

$$\begin{aligned} \text{maximize:} \quad & f(x, y) \\ \text{subject to:} \quad & g(x, y) = 0 \\ & \mathcal{L} = f(x, y) - \alpha g(x, y) \end{aligned}$$

# Lagrangiano

## Definición de L:

Multiplicadores de Lagrange

$$\mathcal{L} = \underbrace{\frac{1}{2} \mathbf{w}^T \mathbf{w}}_{\text{Problema de optimización original}} + \sum_{i=1}^n \alpha_i \underbrace{(1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))}_{\text{Restricción del problema para cada } i}$$

$\frac{1}{2} \|\mathbf{w}^2\|$   
 $y_i(\mathbf{w} \bullet \mathbf{x}_i + b) - 1 \geq 0$

Configuración de gradientes de L respecto a incógnitas (W y b):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} + \sum_{i=1}^n \alpha_i (-y_i) \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$
$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0$$

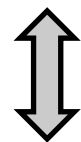
Restricción de no-negatividad de los multiplicadores de Lagrange:

$$\alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots, n\}$$

# Lagrangiano

Transformamos el problema:

$$\begin{array}{ll}\text{minimize:} & \frac{1}{2} \|\mathbf{w}^2\| \\ \text{subject to:} & y_i * (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 \quad \forall i \in \{1, 2, \dots, n\}\end{array}$$



$$\begin{array}{ll}\text{maximize:} & \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^n \alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)) \\ \text{subject to:} & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots, n\}\end{array}$$

# Lagrangiano

maximize:  $\frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^n \alpha_i (1 - y_i (\mathbf{w}^T \mathbf{x}_i + b))$

subject to:  $\sum_{i=1}^n \alpha_i y_i = 0$   
 $\alpha_i \geq 0 \quad \forall i \in \{1, 2, \dots, n\}$

Cada entidad va quedar a asociada a un peso. Los puntos con alfa mayor a 0 van a ser puntos de soporte del vector (hiperplano).

El peso total aportado por los puntos de ambas clases a la definición del hiperplano es el mismo.

- Al resolver el problema por un método de resolución cuadrático (o similar) obtendremos del Alpha asociado a cada entidad (los que sean mayores a 0 son soportes S). Con esos valores podemos calcular W y b:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

Clase del punto (+1 o -1)

Posición del punto en el espacio

Peso del punto en "construir" el hiperplano.

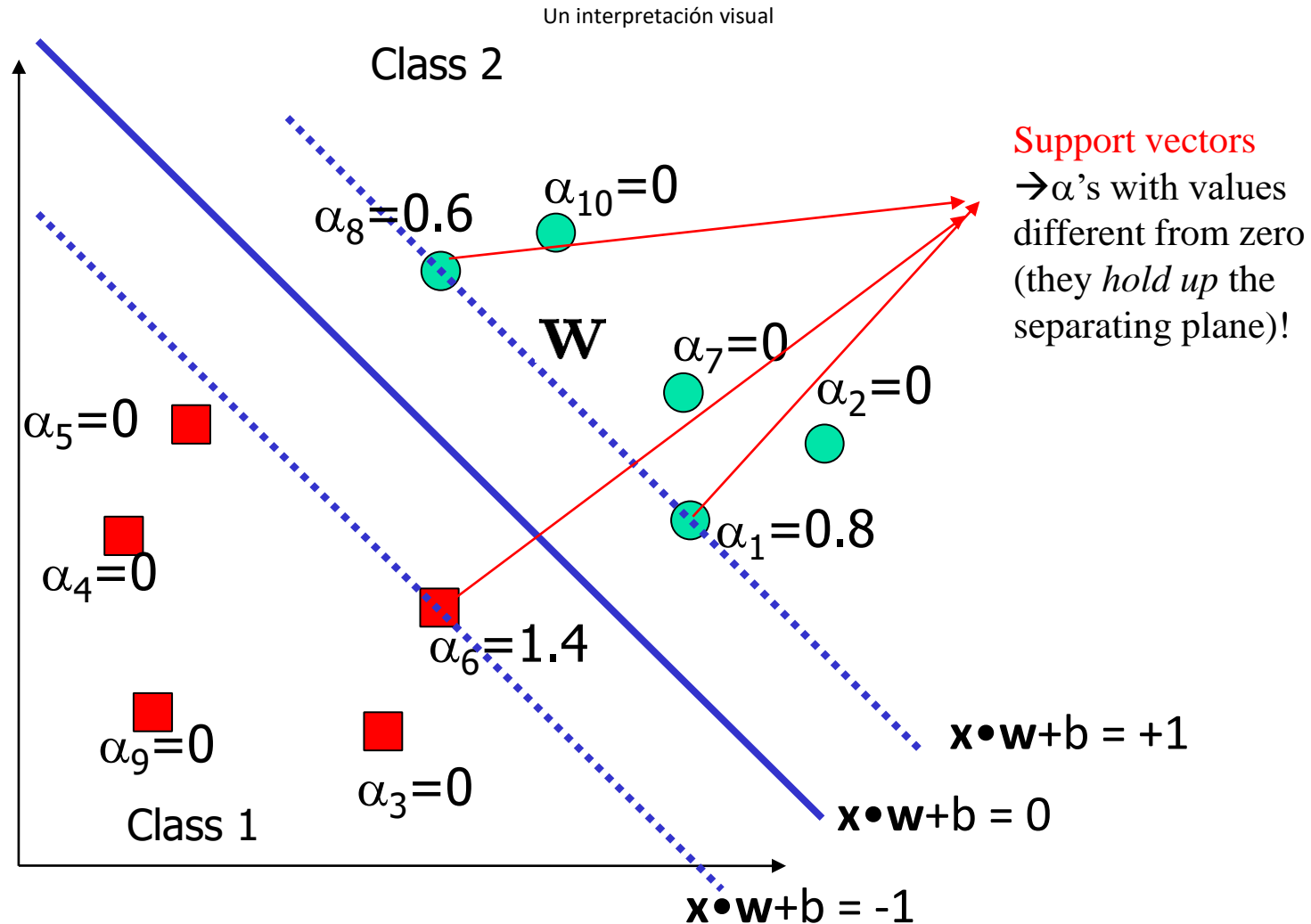
$$b = \frac{1}{S} \sum_{i=1}^S (y_i - w \cdot x_i)$$

Soportes generados

- El bias b queda relacionado a los soportes y sus posiciones.

El vector W es una suma lineal de las entidades multiplicadas por un peso asociado y por una clase asociada.

# Interpretación geométrica





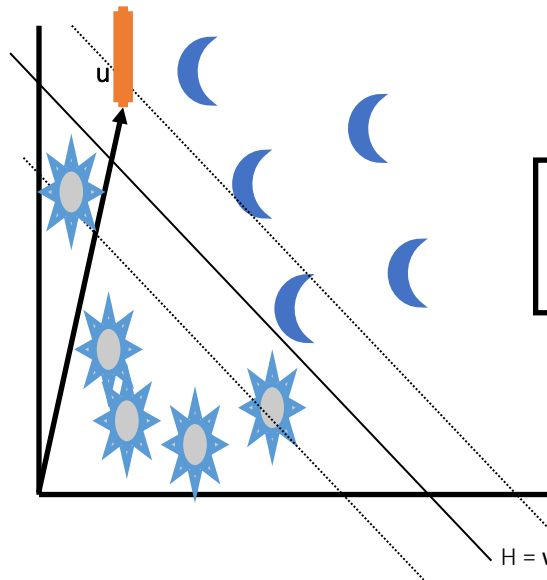
# Lagrangiano

- Dado  $W$  y  $b$  gracias a los Alphas:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

$$b = \frac{1}{S} \sum_{i=1}^S (y_i - w \cdot x_i)$$

- Reemplazando en el problema original, para clasificar un punto nuevo  $U$ :

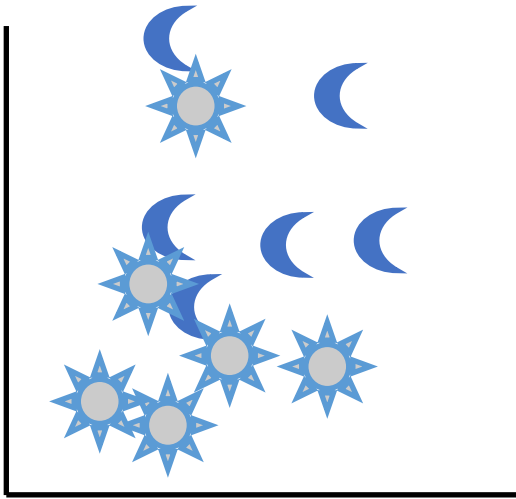


$W \cdot U + b \geq 0 \rightarrow$  Entonces  $U$  es Clase (+) Luna

$$\sum_{i=1}^n \alpha_i y_i x_i \cdot U + \frac{1}{S} \sum_{i=1}^S (y_i - w \cdot x_i) \geq 0 \rightarrow \text{Entonces } U \text{ es Clase (+) Luna}$$

# Caso no separable

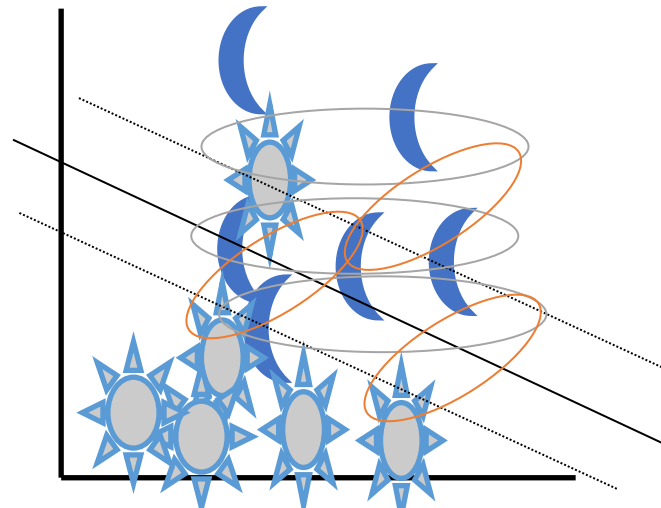
SVM, como lo hemos visto hasta ahora tiene un primer problema. No siempre se puede dividir el espacio con un hiperplano que cumpla las condiciones:



$$y_i(w \cdot x_i + b) - 1 \geq 0$$

- Para solventar esto, podemos agregar a SVM una “holgura”, que permita soportar una determinada cantidad de puntos en el incorrecto lado de la calle; con el fin de lograr generar el hiperplano.

- ¿Cuáles de estos puntos son parte de la holgura?
- ¿Cuáles de estos puntos son soportes?



# Caso no separable

Esta holgura se añade a la restricción como (“ $\xi_i$ ”)

Original

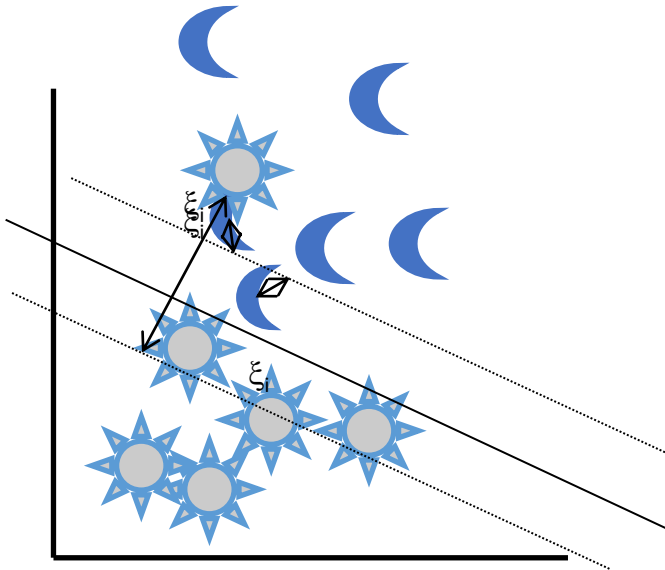
$$1(w \bullet x_i + b) \geq 1 - \xi_i \quad \text{si } y_i \text{ es } +1 \text{ (Luna)}$$

$$-1(w \bullet x_i + b) \leq -1 + \xi_i \quad \text{si } y_i \text{ es } -1 \text{ (Sol)} \rightarrow 1(w \bullet x_i + b) \geq 1 - \xi_i$$

$\xi_i$  :

$$y_i(w \bullet x_i + b) - 1 + \xi_i \geq 0$$

Sa:  $\xi_i \geq 0$

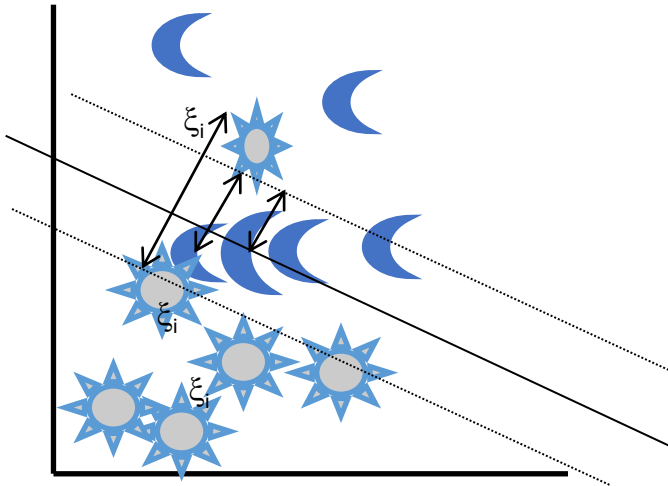


- $\xi_i$  es una distancia, relacionada a cada punto, que indica que tan lejos quedo del lado correcto de la clasificación.
- En caso de ser 0 ese punto se encuentra en el lugar correcto; en caso de ser mayor a 0, ese puntos se encuentra a una distancia opuesta de donde le corresponde según el corte del hiperplano.

# Caso no separable

¿Como afecta al problema de optimización?:

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{2} \|\mathbf{w}^2\| + C \sum_{i=1}^n \xi_i \\ \text{subject to:} \quad & y_i * (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i \quad \forall i \in \{1, 2, \dots, n\} \quad \xi_i \geq 0 \end{aligned}$$



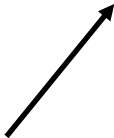
- El problema queda prácticamente igual, solo que al minimizar  $W$  estamos considerando la sumatoria de todas las distancias de los puntos mal divididos.
- $C$  es el “tradeoff” entre el error y el margen. Mientras mas grande  $C$ , mas importancia toma el error ( $\xi_i$ ) al buscar el mejor  $W$ .
- $C$  es un parámetro que nosotros podemos escoger.
- Otra forma de verlo es que  $C$  controla que tanto error le permitimos a SVM tener.

# Caso no separable

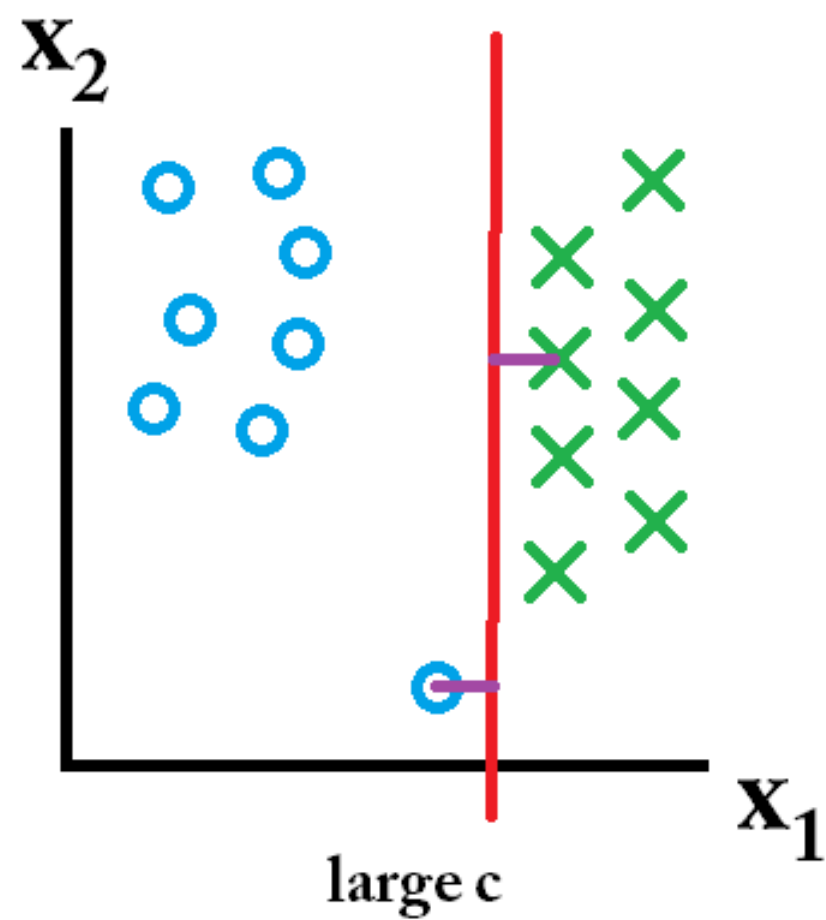
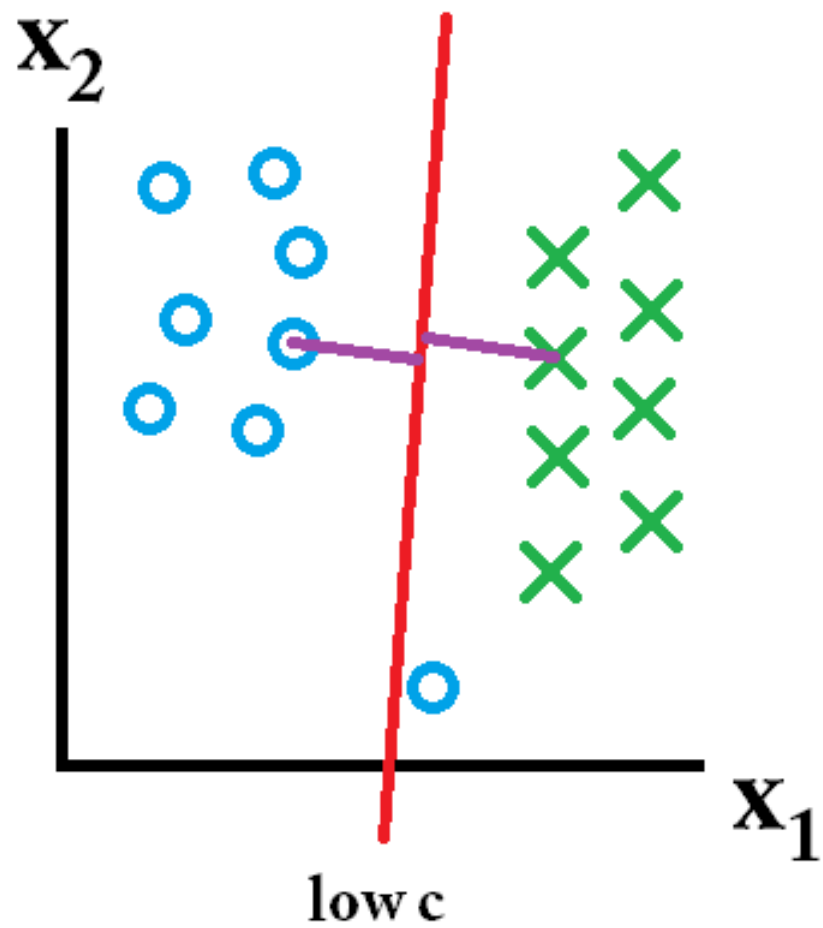
Al someter la nueva función de optimización, con sus restricciones, al Lagrange, obtenemos un resultado prácticamente igual:

$$\begin{aligned} \text{maximize:} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to:} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

Ahora, cada punto va a tener un peso que varia entre 0 (no es soporte) hasta C

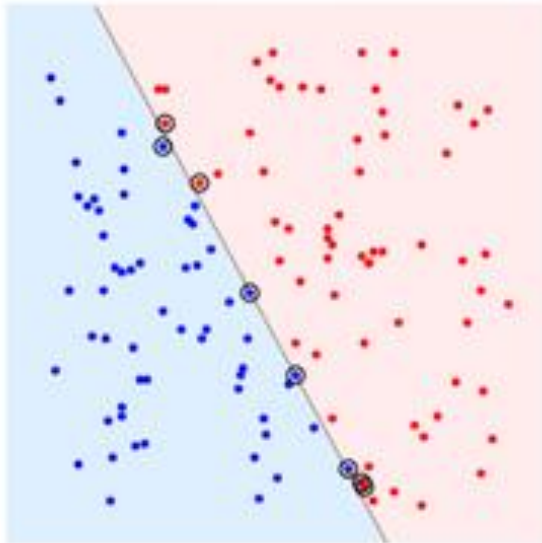


# Sensibilidad de C

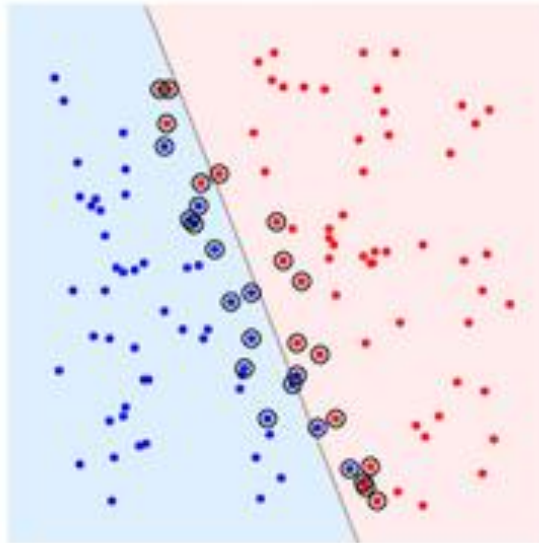


# Ejemplo de C

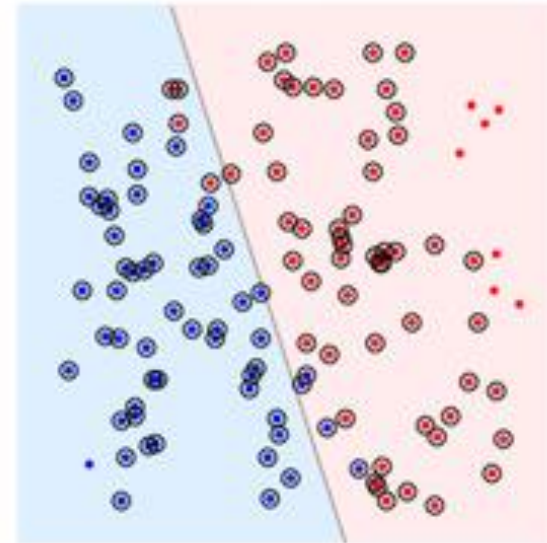
$c=1000$



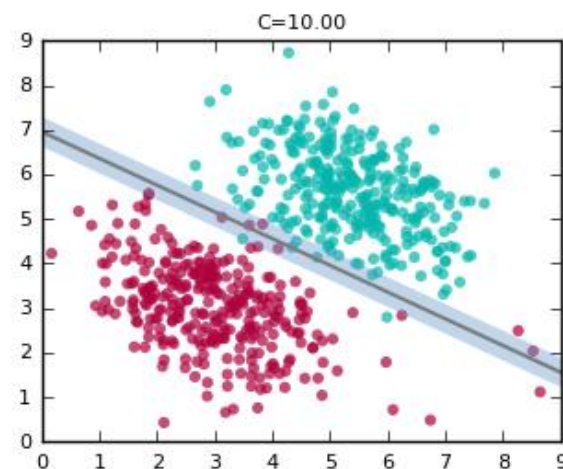
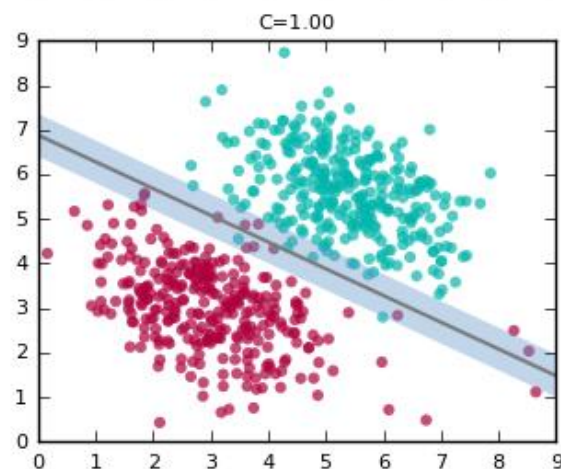
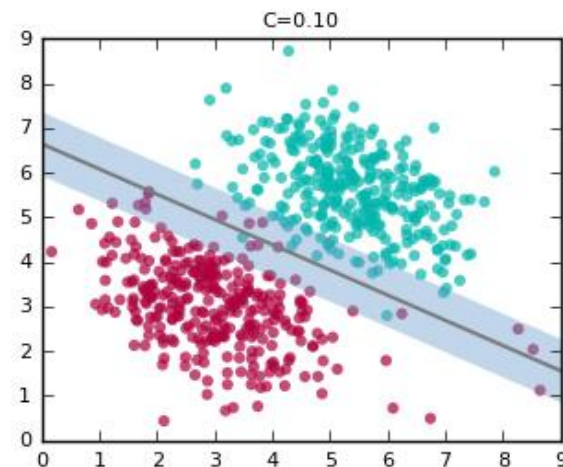
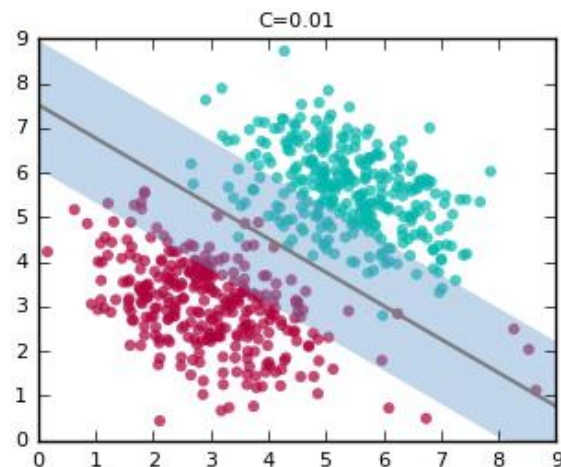
$c=10$



$c=0.1$



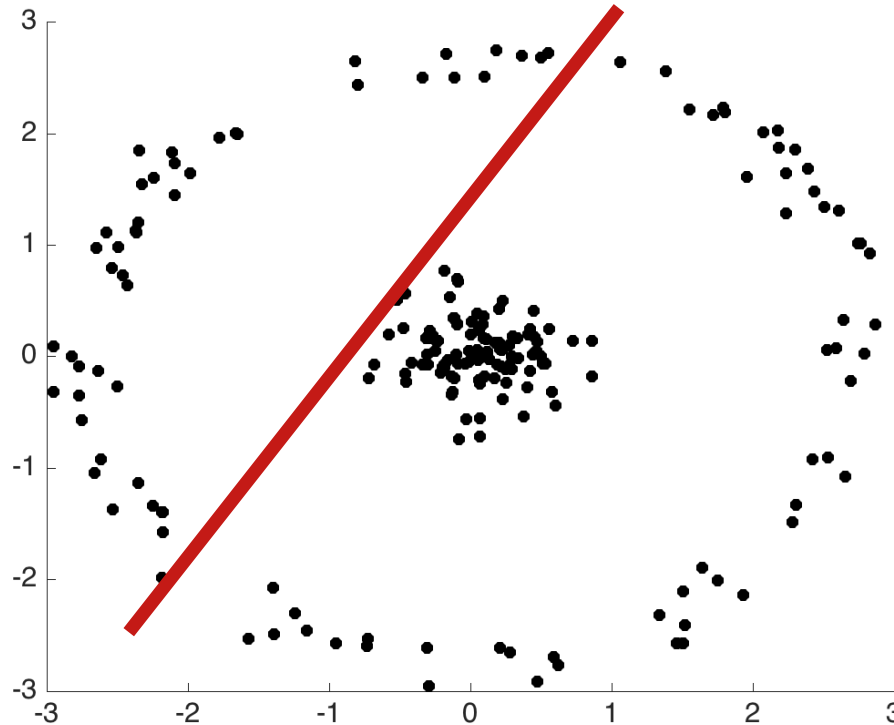
# Ejemplo de C





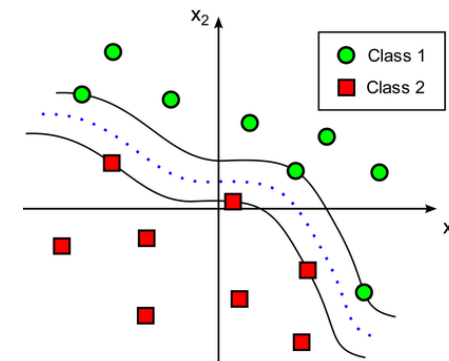
# Limitaciones SVM clásico

Imagina que aplicamos SVM sobre el próximo problema. Puedes seleccionar cualquier  $C$  que quieras. Dibuja la mejor línea de separación.



For any value of  $c > 0$

- Aunque hay claramente 2 clases distintas, no podemos dividir el espacio con el hiperplano de ninguna manera que nos de un buen resultado.
- Es un problema donde necesitamos dividir el espacio de una manera no lineal.

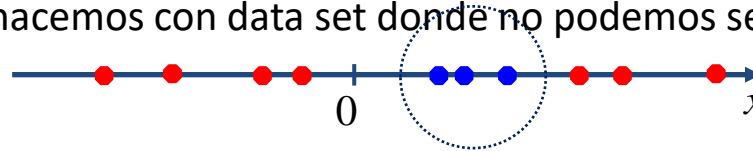


# Caso no lineal

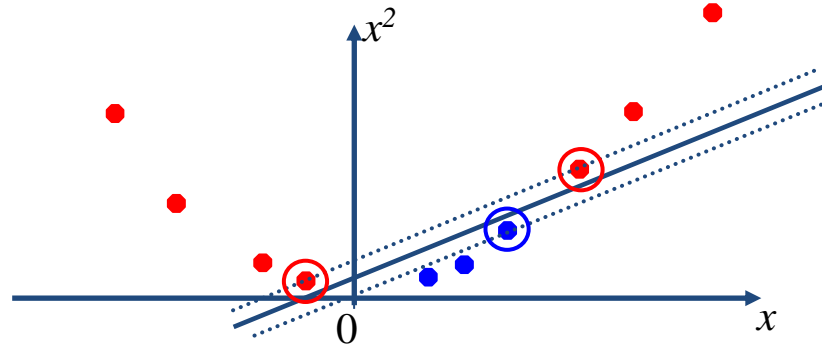
SVM normal funciona bien para DataSet que se pueden dividir linealmente (aunque tengan un poco de ruido, para eso usamos C):



¿Pero que hacemos con data set donde no podemos separar linealmente?



“Mapeamos” o transformamos el DataSet a un dimensión mayor, y en ese lugar buscamos si podemos encontrar un hiperplano y aplicar SVM.



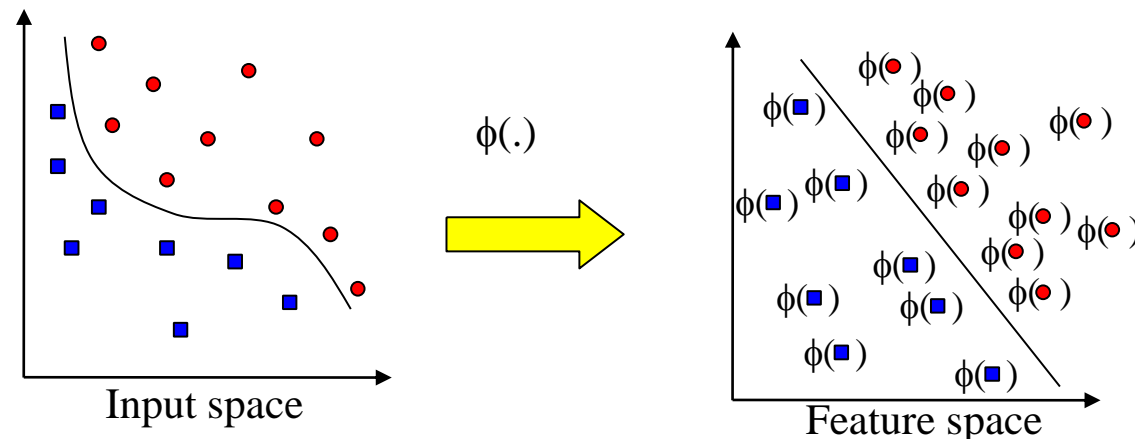
# Transformación del espacio de atributos

La idea general: el input original de datos  $\mathbf{X}$  puede ser mapeado a alguna dimensión superior usando la función  $(\phi(\mathbf{X}))$ , donde podemos entrenar el modelo SVM y encontrar un hiperplano bueno.

Si mapeamos los datos a un espacio con el suficiente numero de dimensiones, entonces generalmente, podrán ser separados linealmente.

El problema es la carga computacional debido al numero de dimensiones y la capacidad de encontrar una manera de separar el espacio.

SVM resuelve estos dos problemas de forma simultanea. Usa “Kernel Tricks” para la transformación del espacio y minimiza  $||\mathbf{w}||^2$  de forma simultanea.



# Truco del kernel

Si nos fijamos en la función de optimización, nos damos cuenta que solo depende del producto punto. Por lo tanto no tenemos que hacer un mapeado explícito, sino utilizar en ese lugar del calculo una función de Kernel.

$$\begin{aligned} \text{maximize: } & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to: } & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

Una función de Kernel  $K$ , es una función que toma dos pares de puntos y evalúa su producto punto en un espacio determinado.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

# Truco del kernel

El siguiente Kernel transforma datos de 2D a 5D:

$$\phi(\mathbf{x}_i) = (1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2})$$

- La función de Kernel  $K(\mathbf{x}_i, \mathbf{x}_j)$  quedaría definida como:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \\ &= 1*1 + \sqrt{2}x_{i1}*\sqrt{2}x_{j1} + \sqrt{2}x_{i2}*\sqrt{2}x_{j2} + x_{i1}^2*x_{j1}^2 + x_{i2}^2*x_{j2}^2 + \sqrt{2}x_{i1}x_{i2}*\sqrt{2}x_{j1}x_{j2} \\ &= 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + x_{i1}^2*x_{j1}^2 + x_{i2}^2*x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \\ &= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2 \end{aligned}$$

- El producto punto puede ser calculado sin tener que hacer literalmente el *mapping* sobre todos los datos.

# Truco del kernel

Nuestro nuevo problema de optimización, considerando una función de Kernel se ve así:

$$\begin{aligned} \text{maximize: } & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to: } & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

Existen cientos de Kernels, distintos (es un área de estudio académico):

Linear

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

Polynomial kernel with degree  $d$

$$\Rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$$

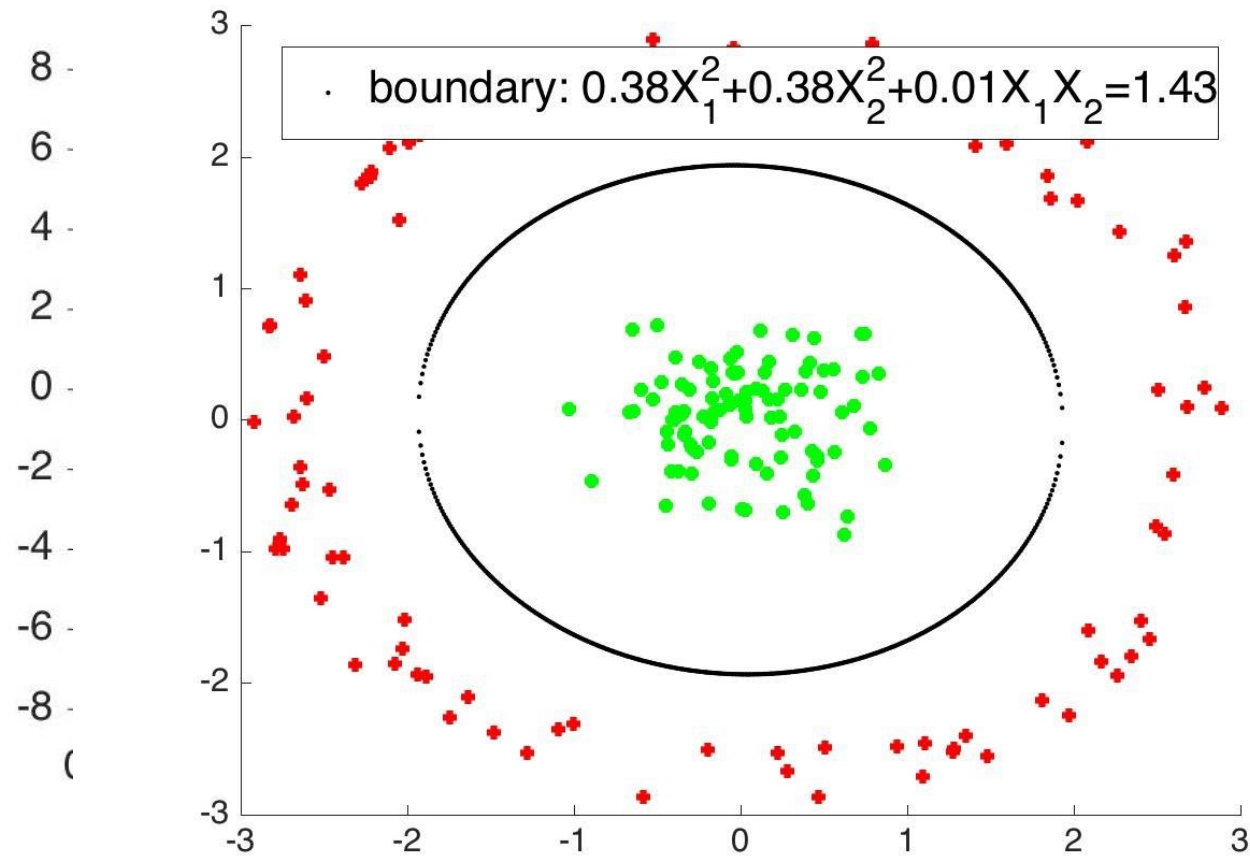
Radial basis function kernel with width  $\sigma$   $\Rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2 / (2\sigma^2))$

Sigmoid with parameter  $\kappa$  and  $\theta$

$$\Rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa * \mathbf{x}_i^T \mathbf{x}_j + \theta)$$

# Ejemplo Kernel polar

En el ejemplo inicial, podemos aplicar la siguiente transformación:  $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$ ; luego aplicamos SVM y luego regresamos los datos al espacio original:



# Debilidades y fortalezas

## Debilidades:

- El proceso de entrenamiento y prueba es lento, debido a tener que solucionar un problema de Lagrange.
- En casi todas sus variedades solo puede clasificar de manera binaria (+1,-1)
- Muy sensitivo al ruido.
- Lo peor: lograr escoger la función de Kernel correcta.

## Fortalezas

- El entrenamiento es fácil, la solución es única y global para todo el espacio.
- SVM no sufre de la maldición de la dimensionalidad !
- No se genera sobretratamiento de manera muy fácil.
- Fácil de entender de manera geométrica.



# **Máquinas de soporte vectorial**

Dr. Raimundo Sánchez  
raimundo.sanchez@uai.cl  
@raimun2