OPENCOURSEWARE
APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS
GRADO EN ESTADÍSTICA Y EMPRESA
Ricardo Aler

Máquinas de Vectores de Soporte

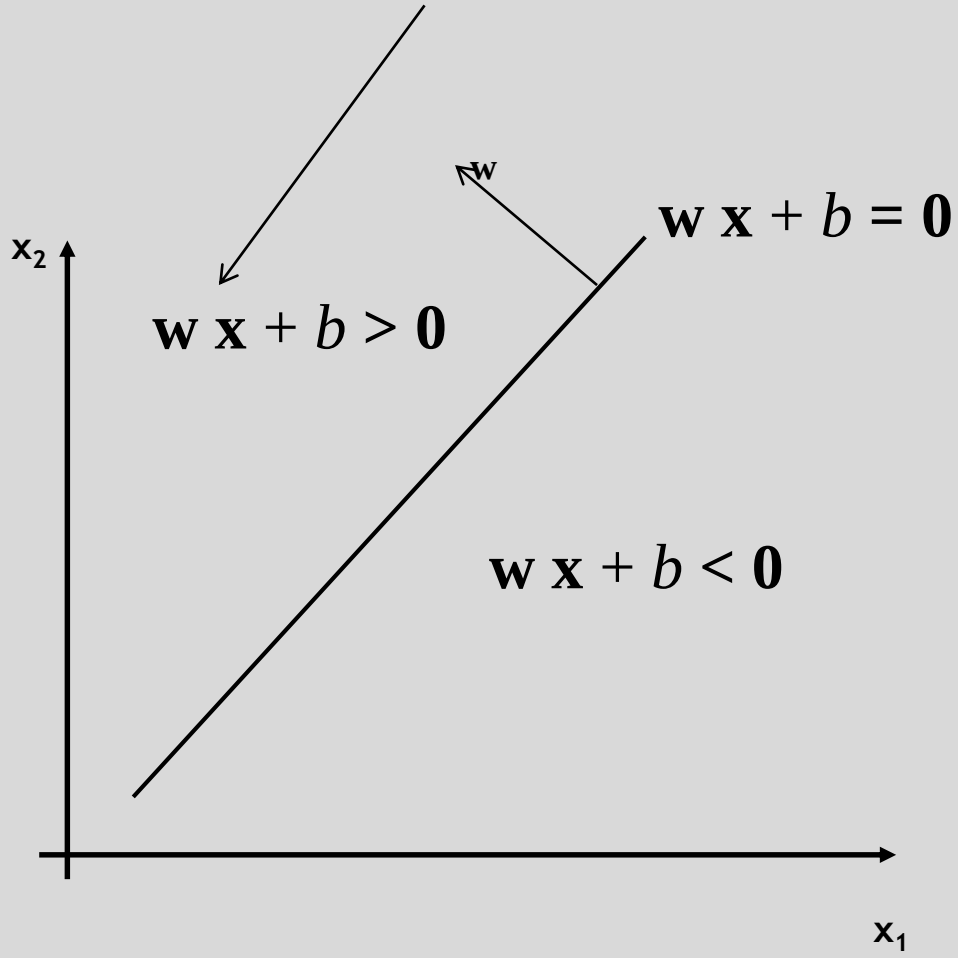
SVMs: Support Vector Machines

Support Vector Machines (SVMs)

- Las SVMs son modelos que pertenecen a una clase de funciones matemáticas
- Establecen fronteras entre distintas clases en el espacio de instancias (también se pueden usar para regresión).

Modelos lineales

Además, cuanto más grande es este valor, más lejos estamos de la recta (es proporcional a la distancia a la recta)



2-D:

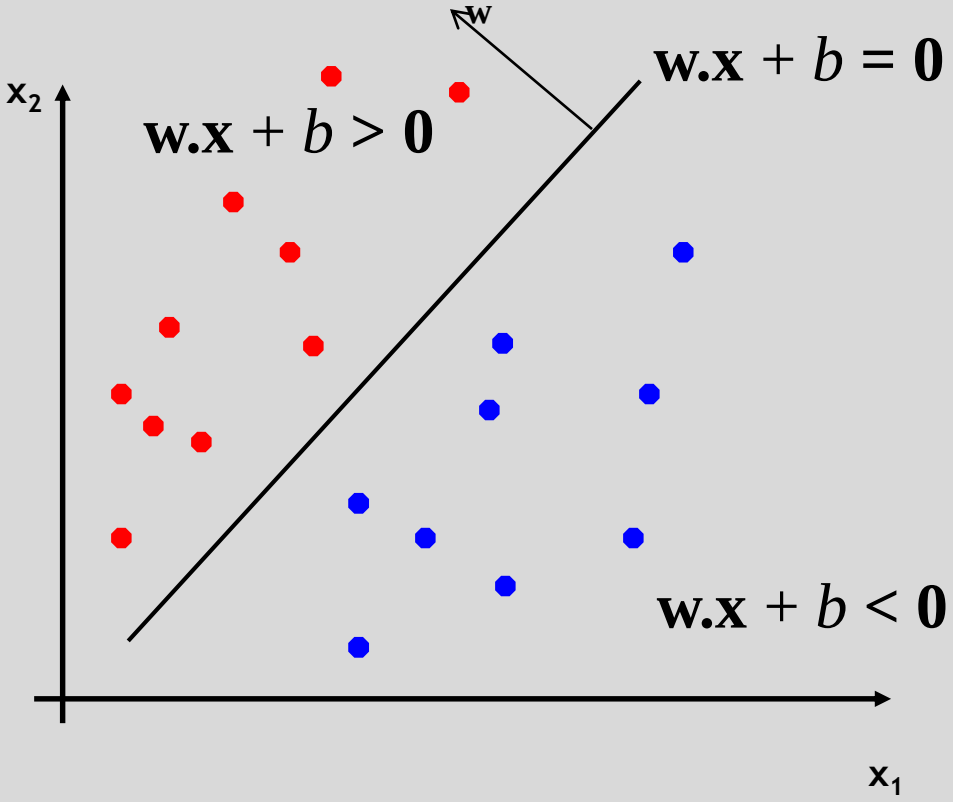
- $y = a * x + b$
- $x_2 = a * x_1 + b$
- $0 = a*x_1 - x_2 + b$
- $w_1*x_1 + w_2*x_2 + b = 0$
- $(w_1, w_2) . (x_1, x_2) + b = 0$
- $\mathbf{w} . \mathbf{x} + b = 0$

N-D:

- $\mathbf{w} . \mathbf{x} + b = 0$
- $(w_1, w_2, \ldots, w_n) . (x_1, x_2, \ldots, x_n) + b = 0$
- $w_1 * x_1 + w_2*x_2 + \ldots + w_n * x_n + b = 0$

Clasificación lineal biclase

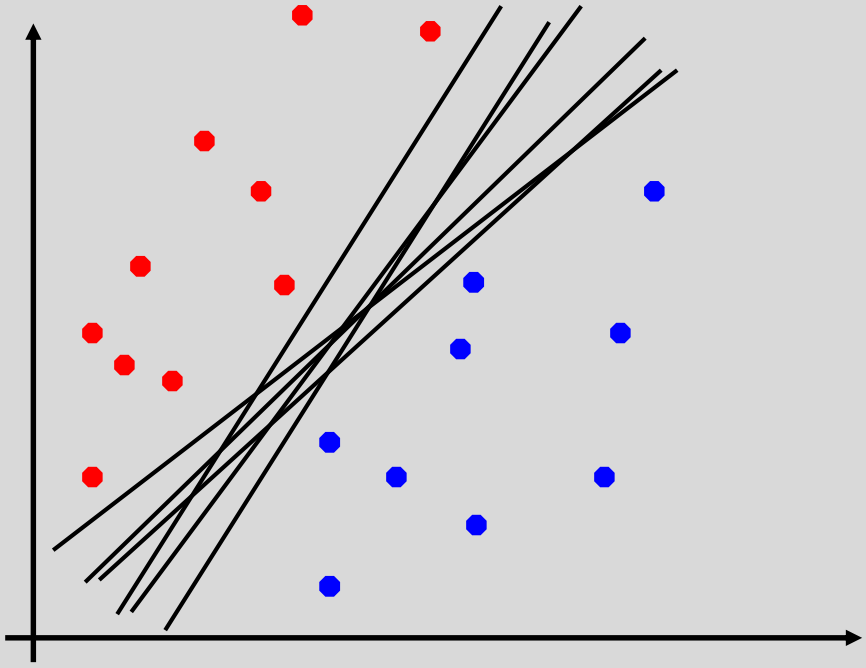
$$f(\mathbf{x}) = \text{sign}(\mathbf{w}.\mathbf{x} + b)$$



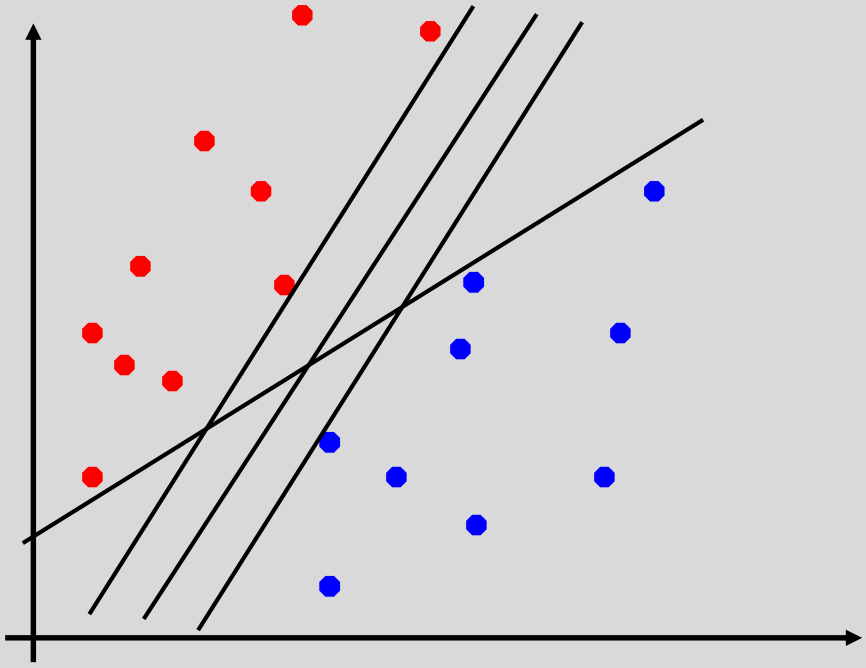
Aprendizaje de modelos lineales

- Existen muchas otras maneras:
  - Discriminante de Fisher (LDA: Linear Discriminant Analysis)
  - Perceptrón simple
  - Adaline
  - Red de neuronas con una neurona oculta
  - Support Vector Machines con kernel lineal
  - …

¿Qué hiperplano elegir?

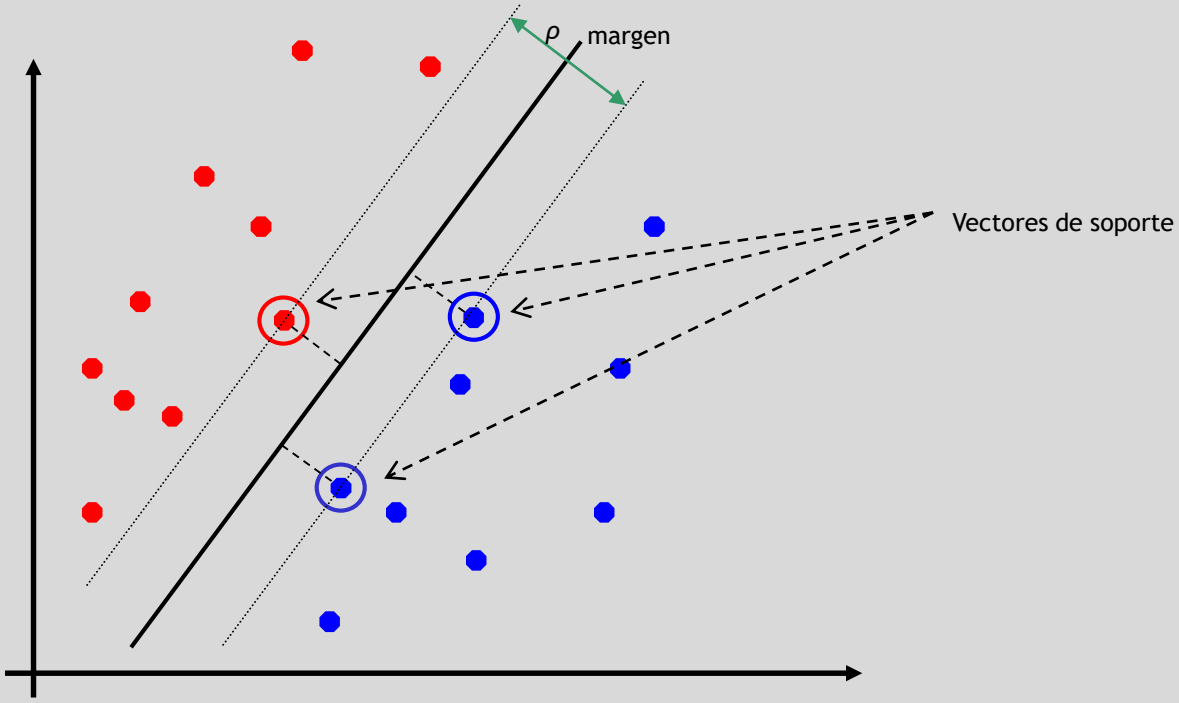


¿Qué hiperplano elegir?



Margen de clasificación

- Las SVMs construyen el hiperplano de separación, de manera que ocupe la posición donde el márgen entre las dos clases es máximo. Eso da lugar a la frontera con mejor capacidad de generalización.
- Los $\mathbf{x}_i$ en la frontera entre las clases son los **vectores de soporte**.
- El margen ρ es la distancia perpendicular al hiperplano (la frontera), entre los vectores de soporte de las dos clases.



Problema de optimización planteado por las SVMs

$$\mathbf{w}\,\mathbf{x}+b = w_1*x_1+w_2*x_2+b = 0$$

- Encuentra **w** y b que:
  - Maximice el margen (en realidad, minimiza **1/margen^2**)
  - Separe las instancias pertenecientes a distintas clases
  - El proceso también encuentra los vectores de soporte
- Este problema de optimización se puede resolver eficientemente mediante programación cuadrática (no entramos en los detalles)
- Si los datos son linealmente separables, tiene solución única.
- Si los datos no son separables linealmente, este problema de optimización no tiene solución.

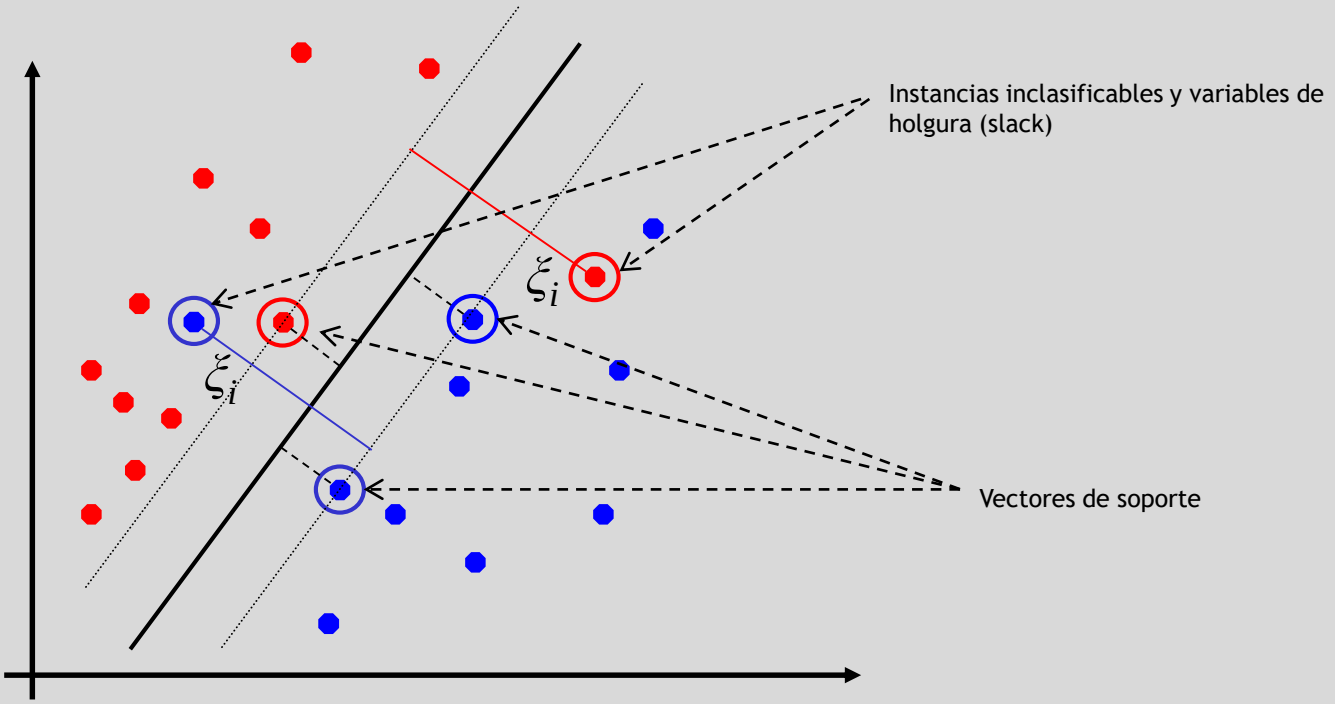
Support Vector Machines

- Casos:
  - Modelo lineal:
    - Los datos son separables linealmente (hard margin)
    - **Los datos no son separables linealmente (soft margin)**
  - Modelo no lineal (kernels)

Soft Margin, slack variables

- Si las clases solapan, pueden existir instancias inclasificables linealmente
- Solución: permitir que algunos puntos estén mal clasificados
- Las variables de holgura ξ_i son las distancias desde el margen opuesto a la instancia inclasificable

Minimizar: $(1/margen^2) + C*\Sigma\xi_i$



*Problema de optimización planteado por las **soft** SVMs (con **slack** variables)*

$$\mathbf{w}\,\mathbf{x}+b = w_1 * x_1 + w_2 * x_2 + b = 0$$

- Encuentra **w** y b que:
  - Maximizar el margen y minimizar las holguras.
    - O sea, minimizar $(1/margen^2) + C * \Sigma \xi_i$

- Este problema de optimización se puede resolver eficientemente mediante programación cuadrática (no entramos en los detalles) y tiene solución única
- Este problema de optimización tiene siempre solución, aunque los datos no sean separables linealmente

Soft Margin, slack variables

- Ahora se permiten “márgenes blandos” con instancias mal clasificadas y variables de holgura. Minimizar:

$$(1/margen^2) + C*\Sigma\xi_i$$

- El parámetro *C* controla el peso que le damos a uno o a otro objetivo y permite controlar el sobreaprendizaje:
  - Si *C* tiene un valor grande …
  - Si *C* tiene un valor pequeño …

Soft Margin, slack variables

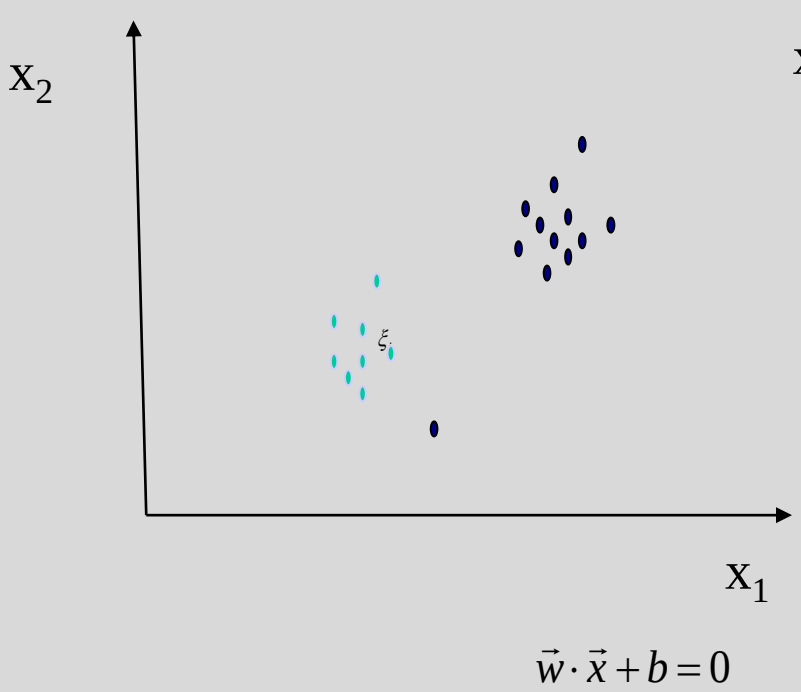
- El parámetro *C* controla el peso que le damos a uno o a otro componente y permite controlar el sobreaprendizaje:
  - Si *C* tiene un valor grande, se le da mucha importancia a que todos los datos de entrenamiento se clasifiquen correctamente (slack variables tienden a cero). Si hay ruido, puede haber overfitting
  - Si *C* tiene un valor pequeño ocurre lo contrario. Si es demasiado pequeño, puede ocurrir que haya demasiados datos de entrenamiento mal clasificados (muchas slack variables con valores altos)

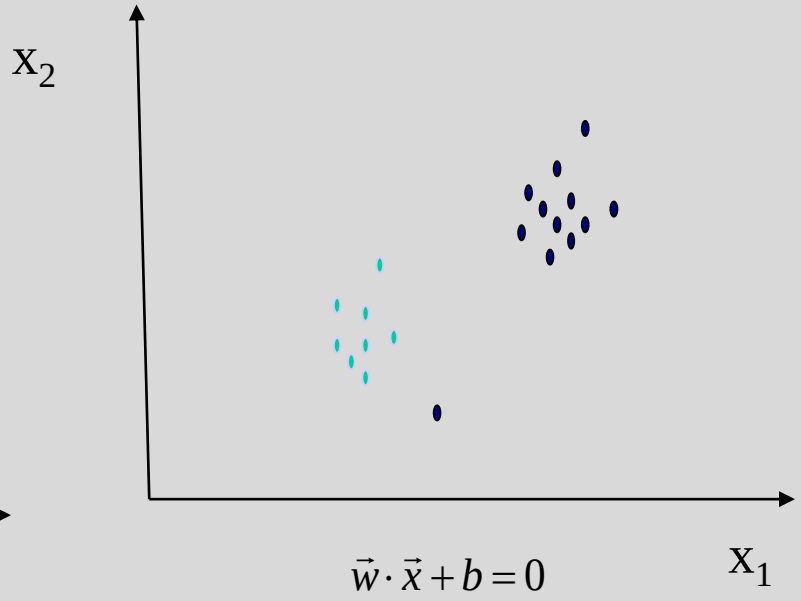
$$(1/margen^2) + C^*\Sigma\xi_i$$

$(1/margen^2) + C^*\Sigma\xi_i$

Soft Margin vs. Hard Margin

Los mismos datos, resueltos con variables de holgura (izquierda) y sin variables de holgura (derecha)





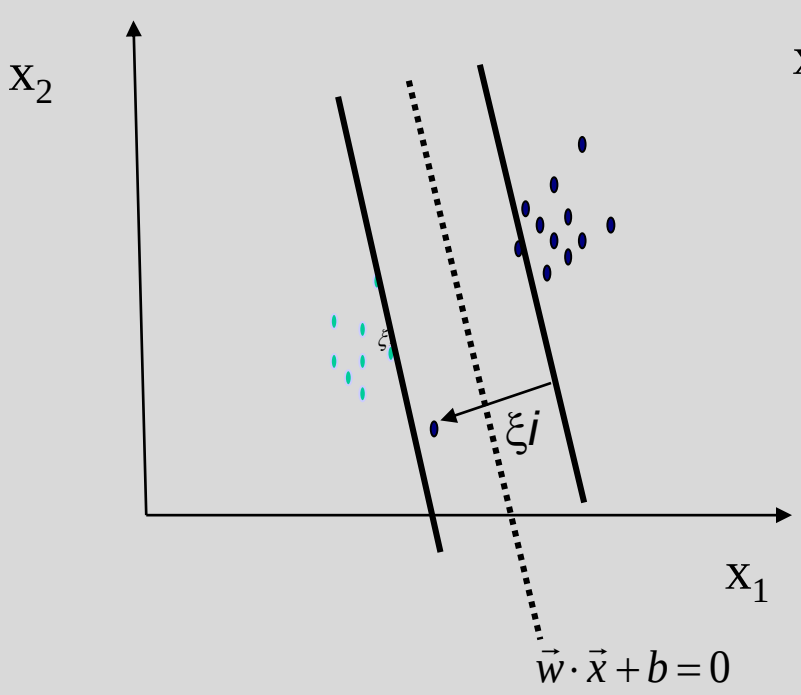
Soft Margin SVM (para un valor apropiado del coste *C)*

Soft Margin con *C* muy grande

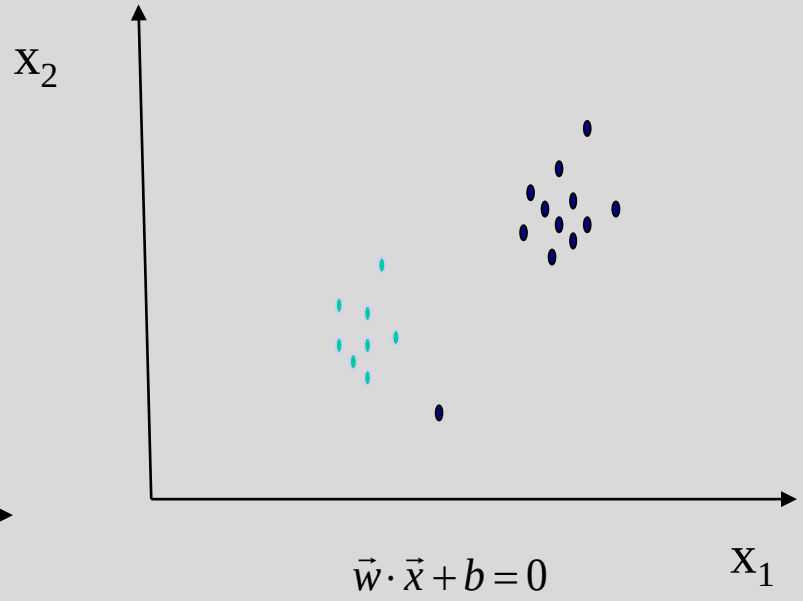
$(1/margen^2) + C*\Sigma\xi_i$

Soft Margin vs. Hard Margin

Los mismos datos, resueltos con variables de holgura (izquierda) y sin variables de holgura (derecha)



Soft Margin SVM (para un valor apropiado del coste *C)*

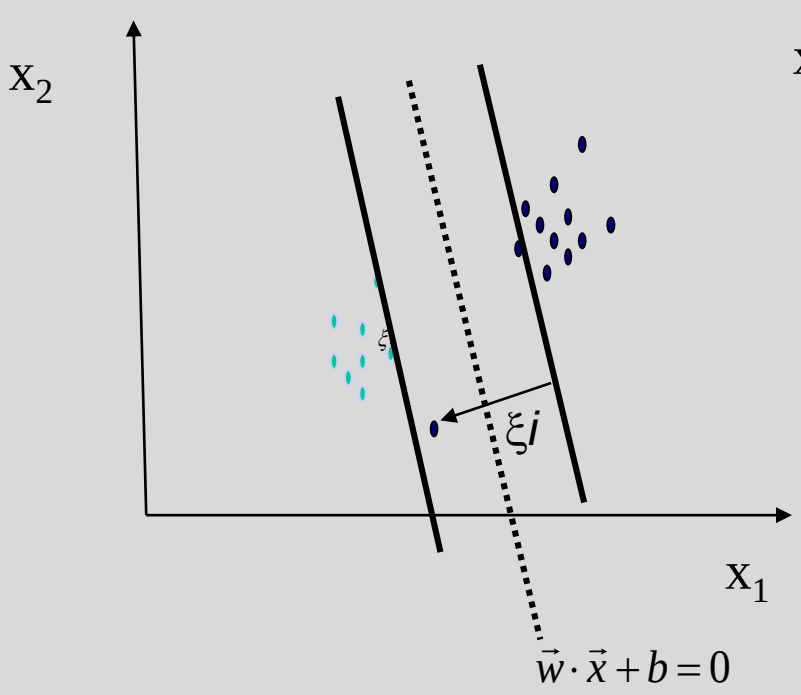


Soft Margin con *C* muy grande

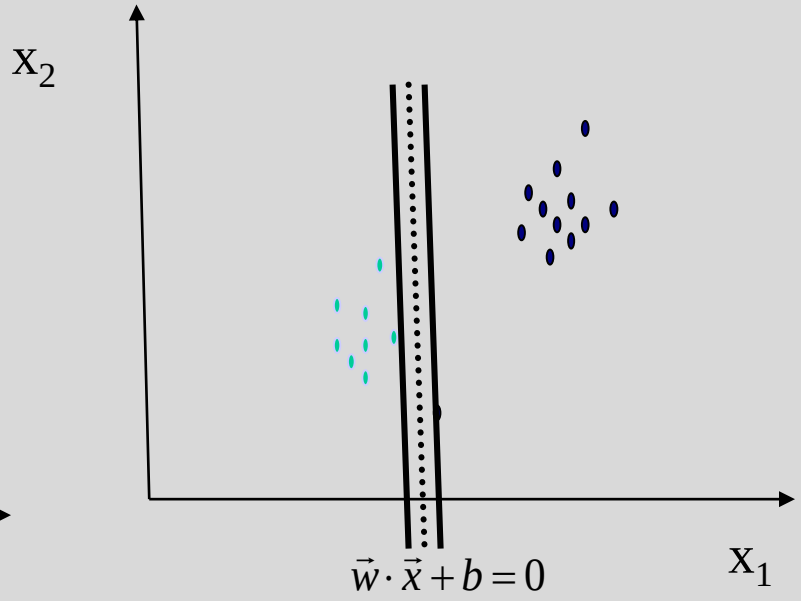
$(1/margen^2) + C*\Sigma\xi_i$

Soft Margin vs. Hard Margin

Los mismos datos, resueltos con variables de holgura (izquierda) y sin variables de holgura (derecha)



Soft Margin SVM (para un valor apropiado del coste *C)*

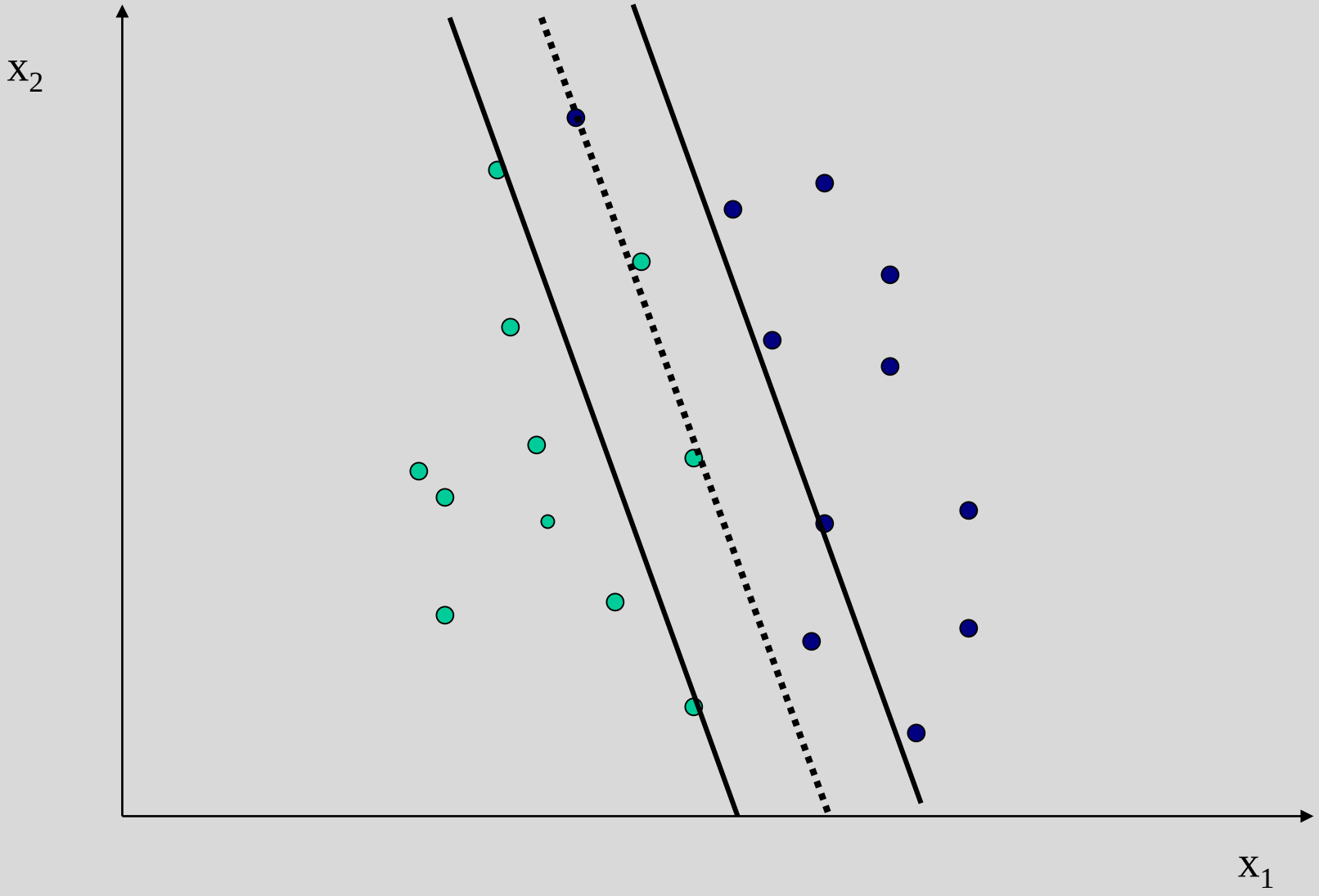


Soft Margin con *C* muy grande

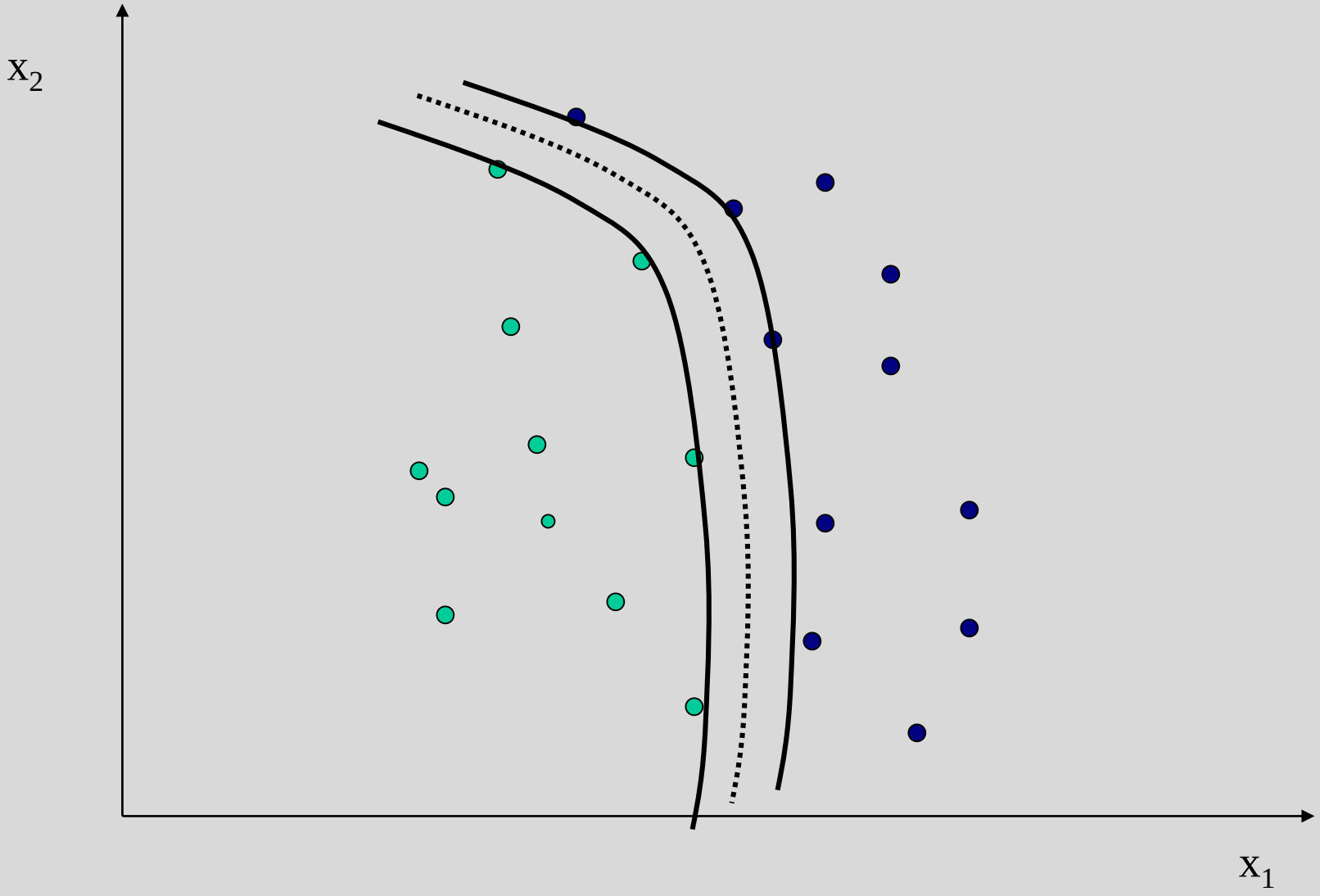
Support Vector Machines

- Casos:
  - Modelo lineal:
    - Los datos son separables linealmente
    - Los datos no son separables linealmente (soft margin)
  - **Modelo no lineal (kernels)**

Modelo lineal vs. Modelo no lineal



Modelo lineal vs. Modelo no lineal



SVMs no lineales

- Matemáticamente, las SVMs no-lineales tienen la siguiente forma:

$$\left(\sum_i (\alpha_i y_i) K(\vec{x}_i, \vec{x}) \right) + b = 0$$

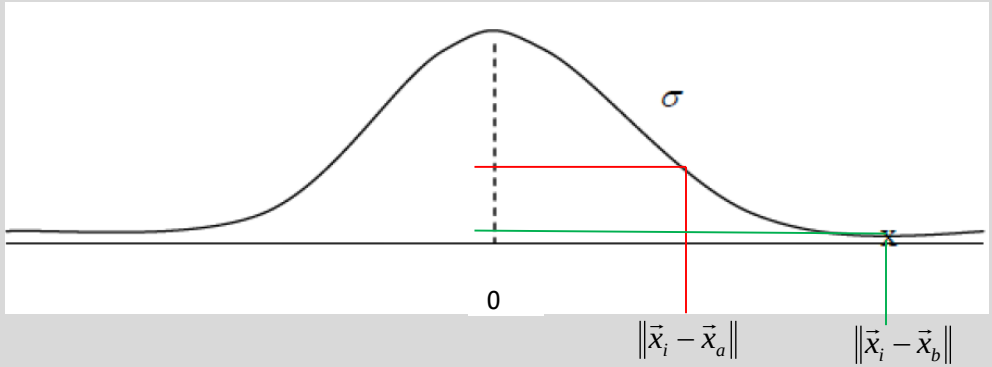
- $\boldsymbol{x}_i$ es cada uno de los vectores de soporte
- $K(\boldsymbol{x}_i, \boldsymbol{x})$ es la función de kernel o de "similitud" (mide cuanto se parecen el vector de soporte $\boldsymbol{x}_i$ y la instancia a clasificar $\boldsymbol{x}$)
- y_i = +1 o -1, si el vector de soporte pertenece a la clase positiva o negativa, respectivamente.
- a_i es un número real que representa el peso de ese vector de soporte en la clasificación final

Kernels

- $K(\boldsymbol{x}_i, \boldsymbol{x})$ es la función de kernel o de “similitud”.
- Existen diferentes kernels, dependiendo de como se quiera medir la similitud.
- Pero el más común es el gausiano: cuanto la distancia entre ambos puntos es más cercana a cero (caso $\boldsymbol{x}_a$), más grande el parecido, pero si están muy lejos, pero más allá de cierta distancia (caso x_b), se considera que los puntos tienen poco parecido.
- Es como si cada vector de soporte tuviera una zona de influencia, dentro de la cual se computa el parecido, pero más allá, el vector de soporte tiene poca influencia.

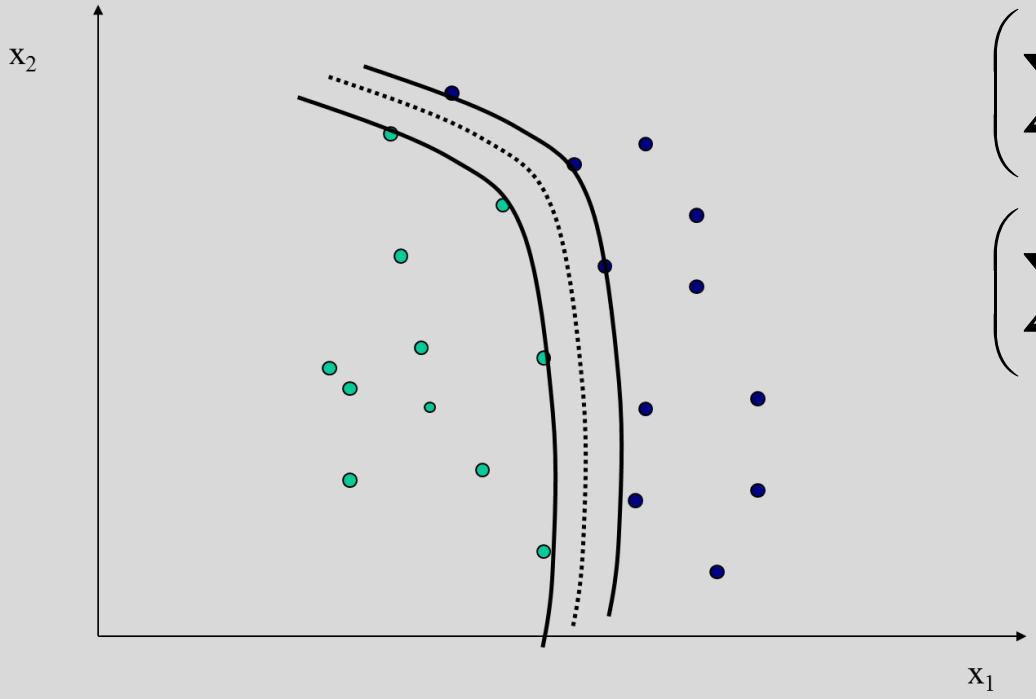
$$K(\vec{x}_i, \vec{x}) = \exp(-\left\|\vec{x}_i - \vec{x}\right\|^2 / 2\sigma^2)$$

$$K(\vec{x}_i, \vec{x}) = \exp(-\gamma\left\|\vec{x}_i - \vec{x}\right\|^2)$$



SVMs no lineales

- Una SVM no lineal con kernel gausiano, se puede ver como una suma ponderada de “similaridades” a los vectores de soporte de la clase positiva (y_i = +1) y los de la clase negativa (y_i = -1).
- Gana la clase que tenga una suma mayor.
- No lo vamos a demostrar, pero el resultado es una función de separación como la de la figura.



$$\left(\sum_i (\alpha_i y_i) K(\vec{x}_i, \vec{x}) \right) + b = 0$$

$$\left(\sum_i (\alpha_i y_i) \exp(-\gamma \left\| \vec{x} - \vec{x}_i \right\|^2) \right) + b = 0$$

SVMs no lineales

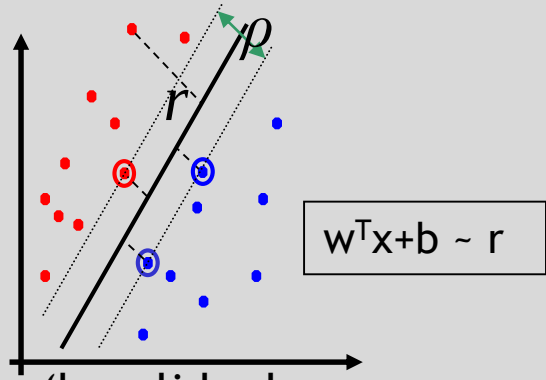
- Las svm no lineales son lineales, pero en un espacio transformado
- Por ejemplo, las svm gausianas son lineales en un espacio de dimensiones infinitas, y es en ese espacio transformado donde se encuentra el hiperplano de máximo margen y se minimizan las holguras
- Esa frontera lineal en el espacio transformado es no lineal en el “feature space” o espacio de instancias original.
- Por ejemplo, un conjunto de datos cuya frontera de separación es una circunferencia en el espacio de instancias (x_1, x_2), tiene una frontera lineal en el espacio transformado (v_1, v_2), si se hace la transformación $v_1=x_1^2$, $v_2=x_2^2$

De biclase a multiclase

- Si el clasificador es capaz de manejar varias clases, no hay que hacer nada
- Si el clasificador es binario (biclase):
  - **One versus all** (o one versus rest): se utilizan tantos hiperplanos separadores como clases. Cada hiperplano separa una clase de todas las demás. Al final, el hiperplano del cual está mas lejos la instancia a clasificar, es el que gana (cuanto mas lejos está un punto de un hiperplano, mas "inmerso" está la instancia en la clase)
  - **One versus one**: se utilizan tantos hiperplanos como parejas de clases hay. Cada hiperplano separa una clase de otra. Por ejemplo, si hay 4 clases, habrá 6 hiperplanos: 1|2, 1|3, 1|4, 2|3, 2|4, 3|4. Al final, los hiperplanos votan. Esta aproximación suele ser demasiado costosa, si el número de clases es grande.

Clasificación con multiples clases

- One-versus-all:
  - Ej con 3 clases, generar:
    - Separador 1 vs. 2+3
    - Separador 2 vs. 1+3
    - Separador 3 vs. 1+2
    - En test, escoger el separador que de mayor valor (la salida de una SVM es la distancia al hiperplano)



- One-versus-one:
  - Ej con 3 clases, generar:
    - Separador 1 vs. 2
    - Separador 1 vs. 3
    - Separador 2 vs. 3
    - En test, por votación
- Hay SVMs que permiten multiclase

Metodología para el uso de SVMs

1. Normalizar (escalado) los atributos
2. Ajuste de hiper-parámetros:
   - Parámetro *C* (coste): (valores grandes tienden a sobreajustar, valores pequeños a infra-ajustar)
   - Kernel:
     - Lineal (también llamado polinómico con exponente 1): ninguno
     - Gaussiano: gamma o sigma (sigmas pequeñas (o gammas grandes) tienden a sobreajustar)

- Nota: probar primero con un kernel lineal, y después con uno Gaussiano.