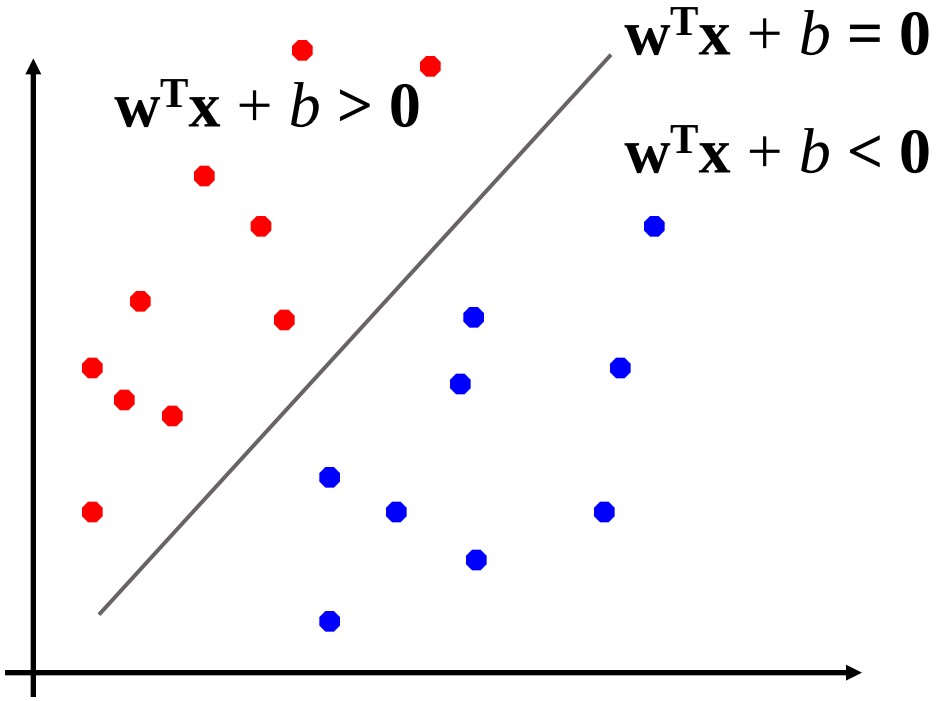
Support Vector Machines

Separadores lineales

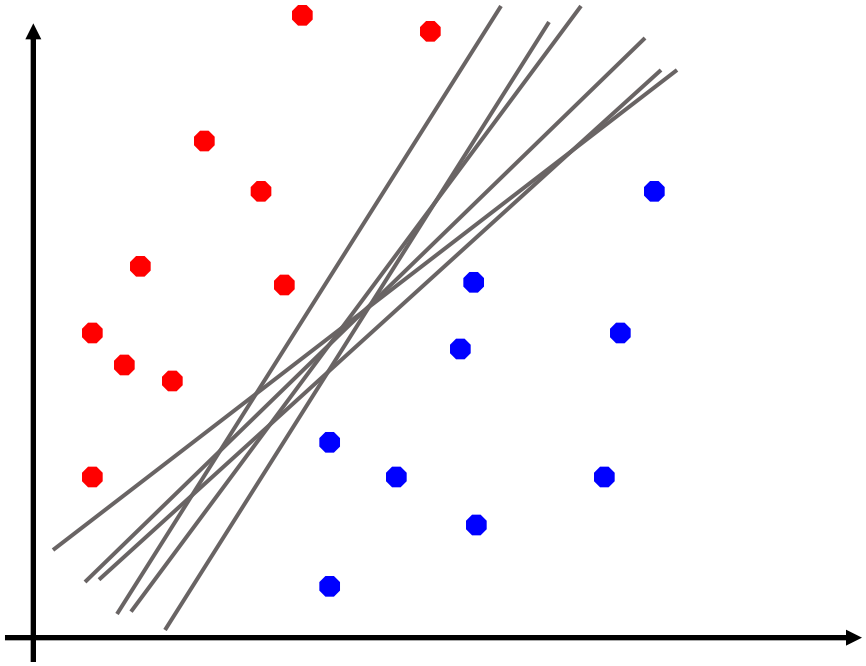
- Clasificacion binaria puede ser vista como la tarea de separar clases en el espacio de caracteristicas



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

Separadores lineales

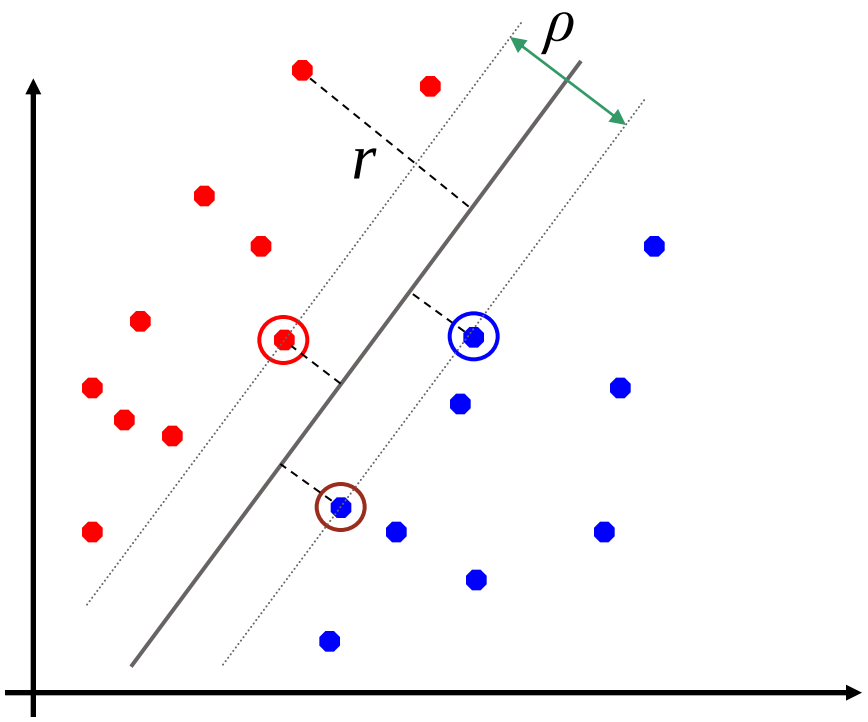
- Cual de todos es optimo?



Margen de clasificacion

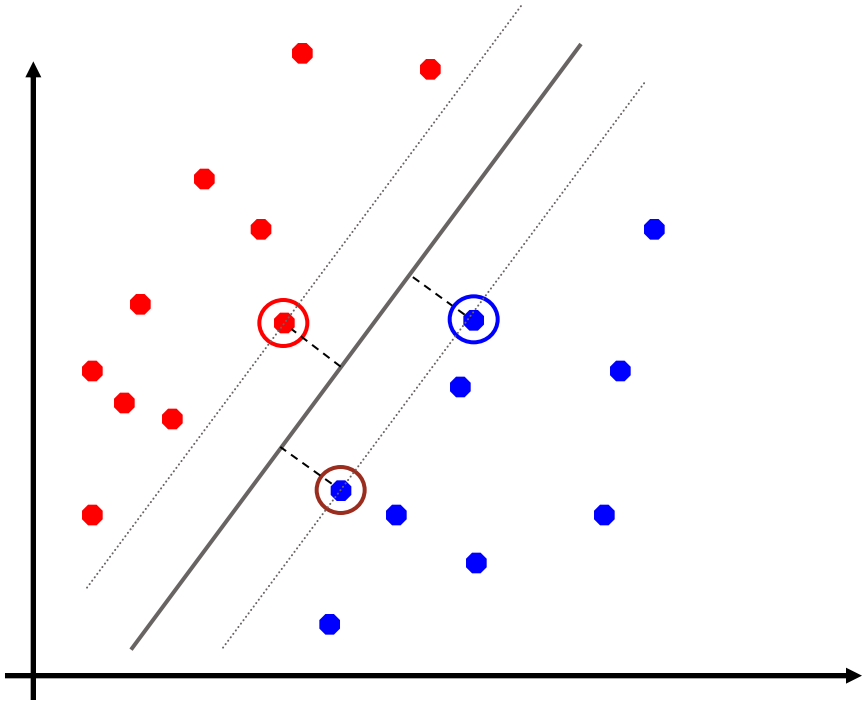
$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$

- Distancia desde un ejemplo $\mathbf{x}_i$ al separador es
- Los ejemplos mas cercanos al hiperplano son los vectores de soporte.
- ***Margen*** ρ del separador es la distancia entre los vectores de soporte



Calsificacion por maximo margen

- Maximizar el margen es bueno de acuerdo a la intuicion
- Implica que solo los vectores de soporte importan los otros ejemplos de entrenamiento son ignorables.



SVM lineal matematicamente

- Consideremos training set $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ separado por un hiperplano. Entonces para cada ejemplo en entrenamiento $(\mathbf{x}_i, y_i)$:

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \text{ if } y_i = -1$$
$$\mathbf{w}^T\mathbf{x}_i + b \geq 1 \text{ if } y_i = 1$$
$$\Leftrightarrow \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

- Para cada vector de soporter $\mathbf{x}_s$ la distancia entre cada $\mathbf{x}_s$ y el hyperplano es

$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- Entonces el margen puede expresarse a traves de **w** y b como:

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

SVM lineal matematicamente

- Entonces podemos formular el problema de optimizacion cuadratica*:*

Encontrar $\mathbf{w}$ y b tales que

$\rho = \frac{2}{\|\mathbf{w}\|}$ se maximiza

y para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Reformulado como:

Encontrar $\mathbf{w}$ y b tal que

$\Phi(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ se minimiza

y para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

Resolviendo el problema de optimizacion

Encontrar $\mathbf{w}$ y b tal que
$\Phi(\mathbf{w}) = \mathbf{w}^T\mathbf{w}$ se minimiza
y para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i\,(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- Se necesita optimizar una funcion cuadratica sujeto a restricciones lineales
- Problemas de optimizacion cuadratica son problemas de programacion matematica conocidos para los cuales existen muchos algoritmos no triviales que proveen soluciones.
- La solucion involucra construir un problema dual donde un *multiplicador* de *Lagrange* α_i se asocia con toda desigualdad que restringe el problema primal (original) :

Encontrar $\alpha_1 \ldots \alpha_n$ tales que
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ ise maximiza y
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ para todo α_i

Solucion del problema de optimizacion

- Dada una solucion $\alpha_1 \ldots \alpha_n$ del problema dual, una solucion del primal es:

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \qquad b = y_k - \Sigma \alpha_i y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_k \quad \text{for any } \alpha_k > 0$$

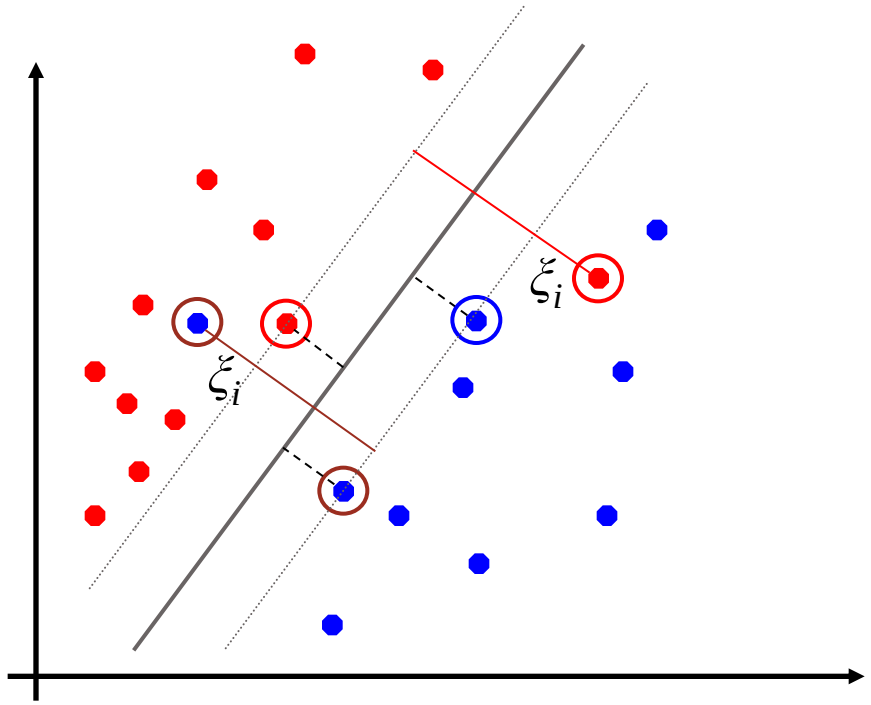
- Cada no zero α_i indica que su correspondiente $\mathbf{x}_i$ es un vector de soporte .
- La funcion de clasificacion es (w no se necesita explicitamente):

$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{x} + b$$

- Notemos que se basa en el *producto interno* entre el punto de test $\mathbf{x}$ y los vectores de soporte $\mathbf{x}_i$
- Tambien recordar que resolver el problema de optimizacion involucra calcular todos los productos internos $\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j$ con todos los puntos de entrenamiento.

Clasificacion con margen soft

- Que pasa s el grupo de entrenamiento no es linealmente separable?
- *Se pueden agregar Slack variables* ξ_i para mejorar la clasificacion de puntos dificiles , lo cual resulta en un *soft margin*.



Clasificacion con margen soft matematicamente

- La formulacion anterior es:

Encuentre un $\mathbf{w}$ y un b tales que
$\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w}$ se minimiza
para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- La formulacion modificacda contiene slack variables:

Encuentre $\mathbf{w}$ y b tal que
$\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w} + C\sum\xi_i$ se minimiza
para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

- Parametro *C* con una forma de controlar sobreentrenamiento: maneja la importancia relativa de maximizar el margen y ajustar los datos de entrenamiento

Solucion para la clasificacion soft

- Problema dual es identico al caso separable (no seria indentico si la penalizacion por norma de las slack variables $C\Sigma\xi_i^2$ se usara en la funcion objetivo primal , se necesitarian mutiplicadores de Lagrange adicionales para las slack variables):

Encontrar $\alpha_1 \ldots \alpha_N$ tal que

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ se maximiza y

(1) $\Sigma\alpha_i y_i = 0$

(2) $0 \leq \alpha_i \leq C$ for all α_i

- Otra vez, $\mathbf{x}_i$ con non-zero α_i serian vectores de soporte.
- La solucion del problema dual es :

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x}_i$

$b = y_k(1 - \xi_k) - \Sigma\alpha_i y_i \mathbf{x}_i^T \mathbf{x}_k$ for any k s.t. $\alpha_k > 0$

Una vez mas no necesitamos calcular **w** explicitamente para clasificacion:

$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$

Justificacion teorica del ancho del margen

- Vapnik probo lo siguiente :

  La clase de separadores lineales optimos tiene dimension VC, h , acotada por arriba

$$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

donde ρ es el margen , D es el diametro de la menor esfera que contiene a todas las muestras de entrenamiento y m_0 es la dimension.

- Intuitivamente esto implica que sin pensar en la dimension m_0 se puede minimizar la dimension VC maximizando el margen ρ.

- Por lo cual la complejidad del clasificador se mantiene reducida a pesar de la dimension de los vectores.

SVM lineales

- El clasificador es un *hiperplano.*
- Los puntos "mas importantes" son los vectores de soporte; ellos definen el hiperplano
- Los algoritmos de optimizacion cuadratica pueden identificar que puntos de entrenamiento $\mathbf{x}_i$ son vectores de soporte con multiplicadores de Lagrange α_i *no nulos*
- En la formulacion dual del problema y en la solucion, los puntos de entrenamiento aparecen solo en productos internos:

Encontrar $\alpha_1 \ldots \alpha_N$ tal que

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$ es maximizada y

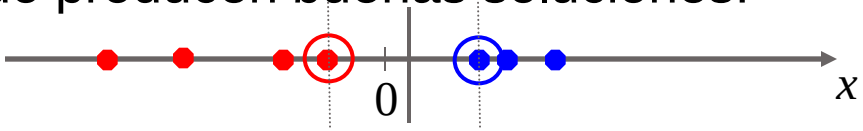
(1) $\Sigma\alpha_i y_i = 0$

(2) $0 \leq \alpha_i \leq C$ para todo α_i

$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^{\mathrm{T}}\mathbf{x} + b$$

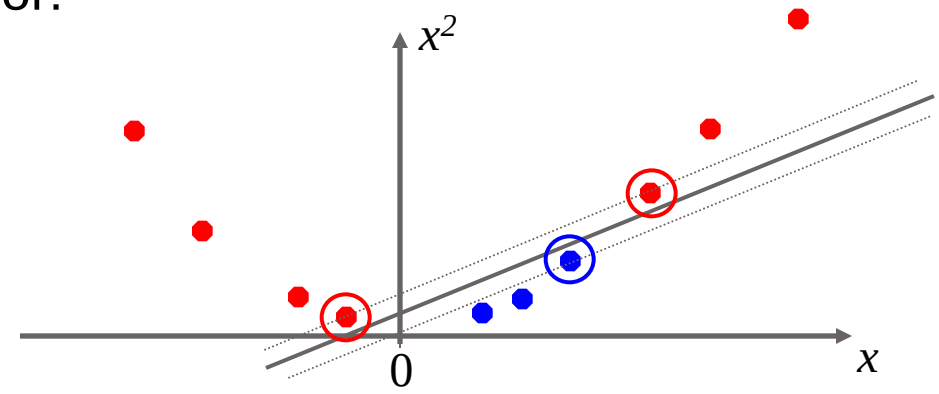
SVM no lineales

- Los grupos de datos que son linealmente separables con algun ruido producen buenas soluciones:



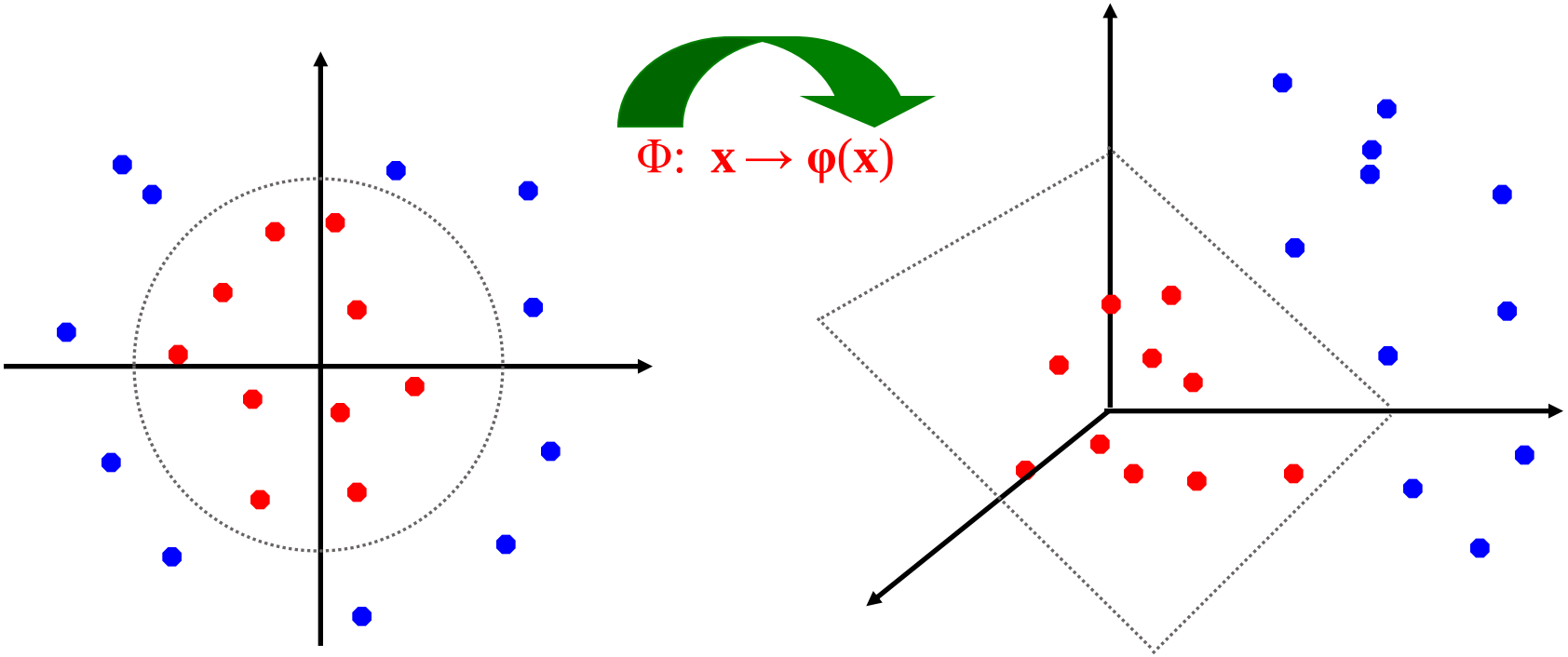
- Pero que pasa con los que son mas dificiles?

- Quizas se prodria proyectar en espacios donde se discrimine mejor:



SVMs no lineales: espacios de caracteristicas

- Idea general: el espacio de caracteristicas inicial puede ser mapeado en otro espacio de mayor dimension donde el grupo de entrenamiento es separable:



El truco del Kernel

- Los clasifcadores lineales se basan en los productos internos $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j$
- Si cada punto se mapea en otro espacio via alguna transformacion $\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$, el producto interno resulta :

$$K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathsf{T}}\boldsymbol{\varphi}(\mathbf{x}_j)$$

- Una funcion de *kernel* es una funcion equivalente a un producto interno en algun espacio de caracteristicas .
- Ejemplo:

  vectores bidimensionales $\mathbf{x}=[x_1 \;\; x_2]$; sea $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j)^2$,

Veamos que $K(\mathbf{x}_i,\mathbf{x}_j)= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathsf{T}}\boldsymbol{\varphi}(\mathbf{x}_j)$:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$$
$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^{\mathsf{T}}\, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}] =$$
$$= \boldsymbol{\varphi}(\mathbf{x}_i)^{\mathsf{T}}\boldsymbol{\varphi}(\mathbf{x}_j), \quad \text{donde } \boldsymbol{\varphi}(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

- Por lo cual una funcion de kernel mapea *implicitamente* datos a un espacio de mayor dimension (sin necesidad de calcular cada $\boldsymbol{\varphi}(\mathbf{x})$ explicitamente).

Que funciones son kernels?

- Para algunas funciones $K(\mathbf{x}_i,\mathbf{x}_j)$, chequear que $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^{\mathsf{T}}\varphi(\mathbf{x}_j)$ puede ser problematico.
- Mercer's theorem:

  Cada funcion simetrica es un kernel

- Funciones simetricas definidas semi-positivas corresponden a una matriz Gram simetrica definida semi-positiva :

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_n)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_n)$
…	…	…	…	…
$K(\mathbf{x}_n,\mathbf{x}_1)$	$K(\mathbf{x}_n,\mathbf{x}_2)$	$K(\mathbf{x}_n,\mathbf{x}_3)$	…	$K(\mathbf{x}_n,\mathbf{x}_n)$

Ejemplos

- Lineal: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
  - MapeoΦ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, donde $\boldsymbol{\varphi}(\mathbf{x})$ es $\mathbf{x}$

- Polynomial de grado p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
  - Mapeo Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, donde $\boldsymbol{\varphi}(\mathbf{x})$ tiene $\binom{d+p}{p}$ dimensiones

- Gaussian (radial-basis function): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$
  - Mapeo Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, donde $\boldsymbol{\varphi}(\mathbf{x})$ es *de dimension infiinita:* todo punto se mapea en una *funcion* (una Gaussiana); la combinacion de funciones de vectores de soporte es el separador .

- Un espacio de dimension mayor tiene su dimensionalidad *intrinsica d* (el mapeo no es sobreyectivo), pero los separadores lineales en el, corresponden a separadores *no-lineales* en el espacio original.

SVM no lineal matematicamente

- Formulacion dual:

Find $\alpha_1 \ldots \alpha_n$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- Solucion:

$$f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$$

- Tecnicas de optimizacion para α_i's son las mismas!

Aplicaciones de SVM

- SVMs fueron propuestos por Boser, Guyon and Vapnik en 1992 y ganaron mas popularidad al final de los 90s.
- SVMs son actualmente uno de los mejores clasificadores para un gran numero de problemas que van desde clasificacion de textos a datos genomicos.
- SVMs pueden ser aplicados a datos muy complejos mas alla de los vectores reales (e.g. grafos, secuencias, datos relacionales) si se diseñan kernels para esos datos.
- las tecnicas de SVM han sido extendidos a otros problemas como regresion [Vapnik *et al.* '97], y componentes principales [Schölkopf *et al.* '99], etc.
- Los algoritmos mas populares de SVMs usan *descomposicion para* mejorar un subconjunto de α_i's en cada paso, e.g. SMO [Platt '99] y [Joachims '99]
- Ajustar SVMs es un arte: seleccionar un kernel especifico y parametros es usualmente realizado manualmente observando los resultados.