Introducción a los métodos Kernel

Universidad Autónoma de Madrid

29 de abril de 2008

Manel Martínez Ramón

Universidad Carlos III de Madrid

Departamento de Teoría de la Señal y Comunicaciones

Transparencias disponibles en www.tsc.uc3m.es/∼manel

Kernels

- **Transformaciones no lineales**
- **Truco de los kernels**
- **Ejemplos**
- **Kernels compuestos**

Introducción

- Métodos Kérnel.
  - Se basan en el viejo truco de los Kernels (Teorema de Mercer).
  - Permiten versiones no lineales de los algoritmos lineales.

- Teoría del aprendizaje estadístico
  - Introducida por Vapnik y Chervonenkis.
  - Porporcionan buena generalización a través del control de la complejidad.
  - Las SVM se derivan de esta teoría.

  Es posible formular máquinas no lineales que reducen drásticamente los inconvenientes de los métodos clásicos de aprendizaje no lineal:
  - Tienen menor coste computaciona y menos heurísticos.
  - Hay existencia y unicidad de soluciones.
  - Se establece un contro explícito del sobreentrenamiento.

Transformaciones no lineales

Para construir un estimador no lineal, debemos transformar los datos de entrada no linealmente.
La transformación no lineal implica una correspondencia hacia un espacio de mayor dimensión, posiblemente infinita:

$$\boldsymbol{x} : \mathbb{R}^n \mapsto \varphi(\boldsymbol{x}) : \mathcal{H}$$

Un ejemplo de transformación no lineal a un espacio de mayor dimensionalidad es una transformación polinómica: sea un conjunto de datos unidimiensional x_i. Aplicamos la siguiente transformación no lineal:

$$\varphi(x) = \{x^2, \sqrt{2}x, 1\}^T : \mathbb{R}^3$$

La pregunta es: existe un producto escalar en ese espacio que pueda ser expresado como función de los datos de entrada x?

Transformaciones no lineales

El producto escalar explícito es:

$$\varphi(x_1)^T \varphi(x_2) = \{x_1^2, \sqrt{2}x_1, 1\}\{x_2^2, \sqrt{2}x_2, 1\}^T = x_1^2 x_2^2 + 2x_1 x_2 + 1$$

Que puede ser escrito como

$$\varphi(x_1)^T \varphi(x_2) = x_1^2 x_2^2 + 2x_1 x_2 + 1 = (x_1 x_2 + 1)^2$$

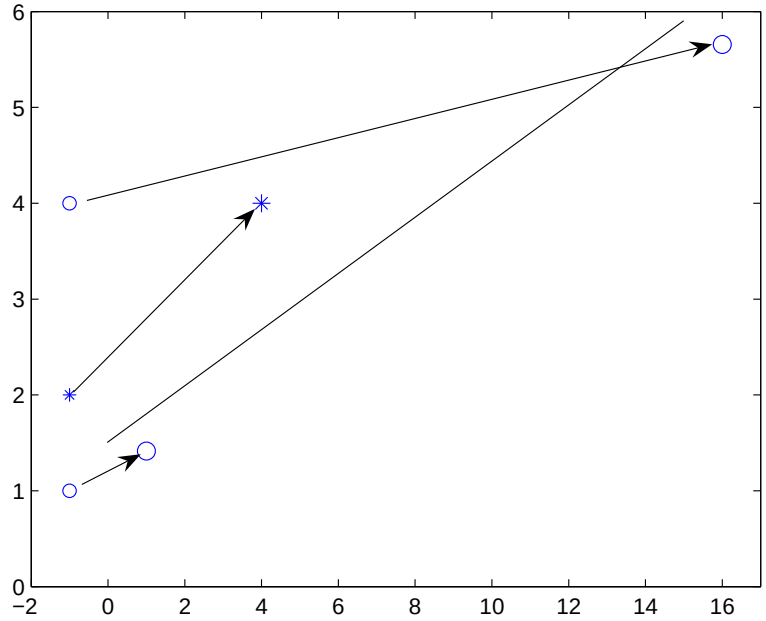
- Existe una expresión del producto escalar en función del espacio de entrada.
- Ese producto escalar se denomina Kernel, y el espacio de mayor dimensionalidad (espacio de características) es un espacio de Hilbert (Reproducing Kernel Hilbert Space, RKHS).
- No necesitamos la expresión de las componentes del vector en el espacio de características.

Transformaciones no lineales

$$\varphi(x_1)^T \varphi(x_2) =$$

$$x_1^2 x_2^2 + 2x_1 x_2 + 1 = (x_1 x_2 + 1)^2$$

Esta transformación incrementa la posibilidad de que haya separabilidad lineal.
Este es un ejemplo de correspondencia en un espacio de Hilbert.
En una dimensión, no es posible clasificar los datos linealmente.



Este es un ejemplo en dos dimensiones; en dimensión infinita, siempre se puede clasificar linealmente cualquier conjunto de datos. *Y eso es bueno y es malo:* La complejidad es infinita, y hay que controlarla.

El truco de los kernels

El hecho de necesitar sólo los productos escalares nos permite reproducir cualquier algoritmo lineal en un espacio de Hilbert. Esto es:

Existe una versión no lineal de cualquier algoritmo lineal basado en datos.

Si encontramos una transformación no lineal

$$\varphi(\boldsymbol{x})$$

a un espacio de mayor dimensionalidad provisto de un producto escalar que puede ser expresado como (kernel):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j) \tag{1}$$

entonces podremos construir una versión no lineal del mismo algoritmo donde la transformación no lineal es φ. Este es el truco de los kernels.

El truco de los kernels

Recordemos que no se necesitan los vectores, así que no debemos preocuparnos por la dimensión del espacio en cuanto a coste computacional. Lo que necesitamos es conocer el kernel.

Los más comunes son:

- Lineal: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$
- Gausiano: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\left(\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}\right)$
- Polinómico: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\boldsymbol{x}_i^T \boldsymbol{x}_j + 1\right)^n$

Son kernels todas aquellas funciones $K(\mathbf{u}, \mathbf{v})$ que verifican el teorema de Mercer, es decir, para las cuales

$$\int_{\mathbf{u},\mathbf{v}} K(\mathbf{u}, \mathbf{v}) g(\mathbf{u}) g(\mathbf{v}) d\mathbf{u} d\mathbf{v} > 0 \tag{2}$$

para toda función $g(\cdot)$ de cuadrado integrable.

Ejemplos

El algoritmo de mínimos cuadrados admite una versión no lineal inmediata. Sea un estimador lineal

$$y_i = \mathbf{w}^T \mathbf{x}_i + e_i \tag{3}$$

Se desea minimizar el error cuadrático medio de la estimación, es decir:

$$\min \mathrm{E} \left(y_i - \mathbf{w}^T \mathbf{x}_i \right)^2 \tag{4}$$

La solución a esta optimización es

$$\mathbf{w} = \mathbf{R}^{-1} \mathbf{p} \approx (\mathbf{X}\mathbf{X}^T)^{-1} \mathbf{X}\mathbf{y} \tag{5}$$

donde $\mathbf{X}$ es una matriz conteniendo un conjunto de vectores columna $\mathbf{x}_i$, e $\mathbf{y}$ es un vector con los valores deseados de y_i.

Ejemplos

Primer paso: encuéntrese una formulación basada en productos escalares. Para ello, téngase en cuenta el siguiente hecho:

$\mathbf{w}$ *es una combinación lineal de los datos de entrada.*

Por lo tanto:

$$\mathbf{w} = \sum_i \alpha_i \mathbf{x}_i = \mathbf{X}\boldsymbol{\alpha} \tag{6}$$

¿Cuánto vale el vector $\boldsymbol{\alpha}$? Se encuentra relacionando (5) (primal) con (6) (dual):

$$\mathbf{X}\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y} \tag{7}$$

y de ahí

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y} = \mathbf{y} \tag{8}$$

$$\boldsymbol{\alpha} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{y} = \mathbf{K}^{-1}\mathbf{y} \tag{9}$$

Ejemplos

La matriz $\mathbf{K}$ contiene los productos escalares de los datos. Una vez hallado $\boldsymbol{\alpha}$, el estimador puede reescribirse como:

$$y_i = \sum_j \alpha_j \mathbf{x}_j^T \mathbf{x}_i \tag{10}$$

Segundo paso: Aplíquese una transformación no lineal $\varphi(\cdot)$ a los datos hacia un espacio de Hilbert provisto de un kernel $K(\cdot,\cdot)$. El estimador (ahora no lineal) se reescribe como

$$y_i = \sum_j \alpha_j \varphi(\mathbf{x}_j)^T \varphi(\mathbf{x}_i) = \sum_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) \tag{11}$$

donde las variables duales α_i se calculan como

$$\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y} \tag{12}$$

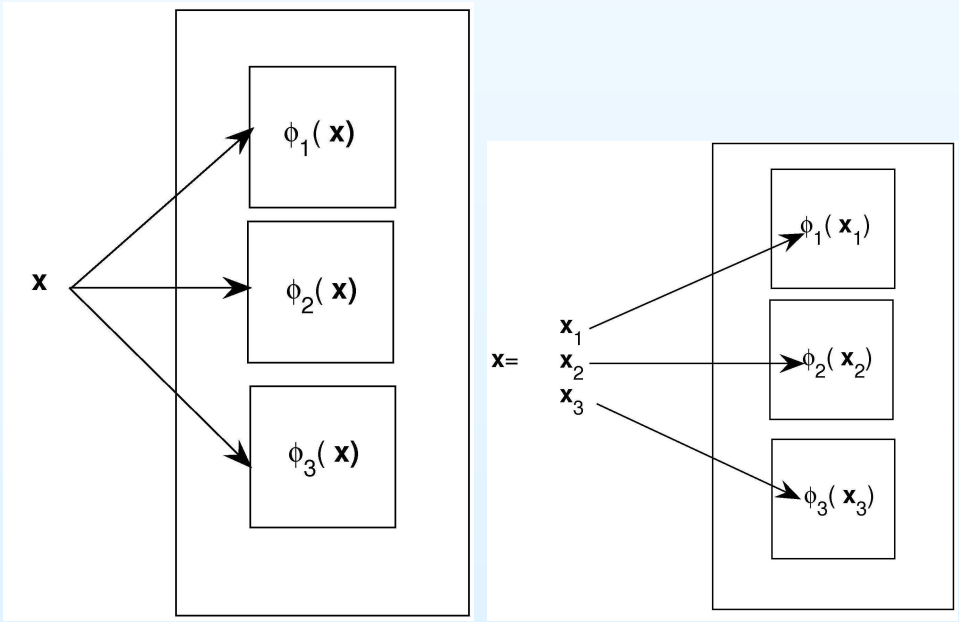
con

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) \tag{13}$$

Kernels compuestos

- Diferentes kernels proporcionan diferentes características a los estimadores.
- Los datos pueden transformarse hacia varios espacios de Hilbert (con diferentes kernels) a la vez.
- Alternativamente, diferentes fragmentos de cada patrón se pueden transformar a diferentes espacios de Hilbert

La concatenación de estos espacios de Hilbert es un nuevo espacio de Hilbert.



Kernels compuestos

En efecto, dado un vector

$$\mathbf{x} = \begin{pmatrix} \mathbf{x^a} \\ \mathbf{x^b} \end{pmatrix}$$

una transformación de la forma

$$\varphi(\mathbf{x}) = \begin{pmatrix} \varphi^a(\mathbf{x^a}) \\ \varphi^b(\mathbf{x^b}) \end{pmatrix}$$

lo sitúa en un espacio de Hilbert compuesto de los espacios $\mathcal{H}^a$ y $\mathcal{H}^b$ cuyo producto escalar es

$$\begin{aligned} \varphi(\mathbf{x}_1)^T \varphi(\mathbf{x}_2) &= \varphi(\mathbf{x^a}_1)^T \varphi(\mathbf{x^a}_2) + \varphi(\mathbf{x^b}_1)^T \varphi(\mathbf{x^b}_2) \\ &= K(\mathbf{x^a}_1, \mathbf{x^a}_2) + K(\mathbf{x^b}_1, \mathbf{x^b}_2) \end{aligned} \tag{14}$$

Lo que demuestra que una suma de kernels es un kernel. Esta técnica se denomina *suma directa de Espacios de Hilbert.*

Kernels compuestos

Existen aproximaciones aún más generales basadas en las propiedades de los espacios de Hilbert.
De la misma forma se puede demostrar que si se construye un vector con el producto tensorial de los vectores $\varphi(\mathbf{x^a})$ y $\varphi(\mathbf{x^b})$, se consigue el kernel tensorial, de la forma

$$K(\mathbf{x}_1, \mathbf{x}_2) = K(\mathbf{x^a}_1, \mathbf{x^a}_2) K(\mathbf{x^b}_1, \mathbf{x^b}_2) \tag{15}$$

Máquinas de vectores soporte

- Complejidad
- Clasificador lineal
- Interpretación Geométrica
- Optimización práctica de la SVM
- Condiciones de Karush Kuhn Tucker
- Solución
- Ejemplos

Complejidad

- Vapnik y Chervonenkis demostraron que el riesgo de error en test de una máquina de clasificación aumenta con su complejidad (Véase tambien Thikonov).
- Demostraron que la complejidad de una máquina lineal disminuye con el módulo de su vector de pesos.

$$min\{Complejidad\} \Leftrightarrow min\{||\boldsymbol{w}||^2\}$$

- Así que tenemos un método sencillo de controlar el sobreentrenamiento de una máquina lineal
- ...y un método sencillo para encontrar versiones no lineales.

Clasificador SVM lineal

- The SVM idea:
  Dado un clasificador lineal $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ minimizamos el error en entrenamiento y, a la vez, su complejidad, minimizando el módulo de su vector de pesos.

- Algoritmo:
  Minimizar

$$||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i \tag{16}$$

  sujeto a

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \tag{17}$$

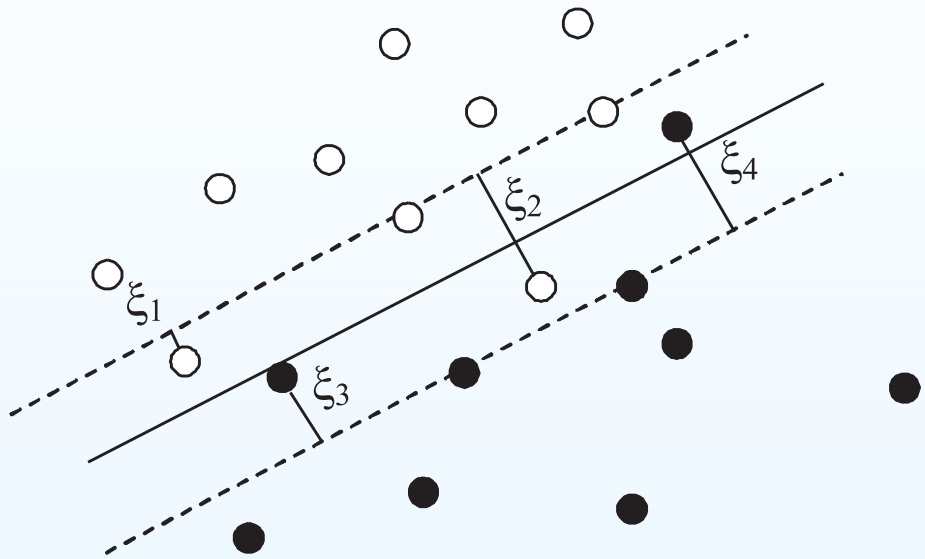
$\xi_i \geq 0$ es la *pérdida* para la muestra $\boldsymbol{x}$. C es el compromiso entre error empírico y complejidad.

Interpretación Geométrica

Llamamos margen al área entre los planos $y(\boldsymbol{w}^T \boldsymbol{x} + b) \geq 1$.

Para una muestra dentro del margen definimos una pérdida ξ_i. Queremos minimizar la suma de pérdidas.

A la vez, minimizamos el módulo de los pesos



Esto es equivalente a maximizar el margen: es inmediato demostrar que

$$d^2 = \frac{1}{||\mathbf{w}||^2}$$

Optimización prctica de la SVM

La minimización del funcional primal

$$\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i \tag{18}$$

sujeto a

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \xi_i \geq 1 \tag{19}$$

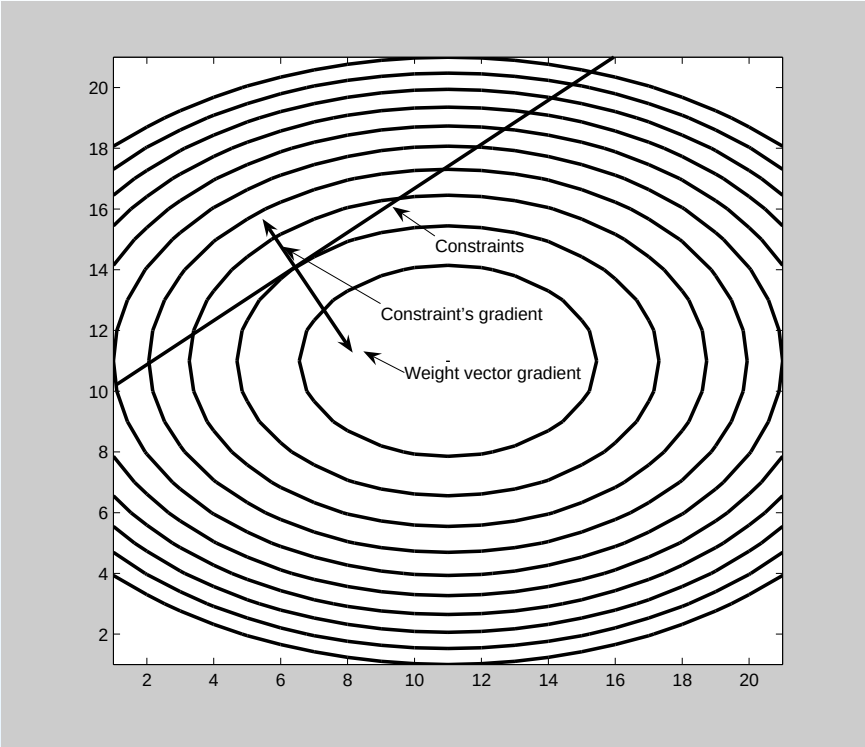
con $\xi_i \geq 0$ es un problema con restricciones.

Debemos usar multiplicadores de Lagrange

El lagrangiano es

$$L = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \xi_i - 1\right] - \sum_{i=1}^{N}\mu_i\xi_i \tag{20}$$

Optimización prctica de la SVM



$$L = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \xi_i\right] - \sum_{i=1}^{N}\mu_i\xi_i \qquad (21)$$

Condiciones de Karush Kuhn Tucker

Las condiciones para este problema son:

$$\frac{\partial L_{pd}}{\partial \boldsymbol{w}} = 0$$
$$\frac{dL_{pd}}{db} = 0 \tag{22}$$
$$\frac{\partial L_{pd}}{\partial \xi_i} = 0$$

$$\alpha_i, \mu_i \geq 0 \tag{23}$$

$$\alpha_i y_i [(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 + \xi_i] = 0 \tag{24}$$

$$\mu_i \xi_i = 0 \tag{25}$$

Solución

Si se aplican las condiciones (22) se obtiene

$$\frac{\partial L_{pd}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \tag{26}$$

$$\frac{dL_{pd}}{db} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{27}$$

$$\frac{\partial L_{pd}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \tag{28}$$

El resultado (26) da la solución para los parámetros

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{29}$$

Solución

Combinando esas ecuaciones con el lagrangiano se obtiene el dual

$$L_d = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i{}^T\boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i \tag{30}$$

que puede escribirse como

$$L_d = -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{K}\boldsymbol{Y}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha} \tag{31}$$

siendo $\boldsymbol{\alpha}$ un vector columna conteniendo los multiplicadores de Lagrange α_i, $\boldsymbol{Y}$ una matriz diagonal $\boldsymbol{Y}_{ii} = y_i$, y $\boldsymbol{K}$ la matroz de productos escalares.

$$\boldsymbol{K}_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j \tag{32}$$

Solución

- La solución es una combinación lineal de muestras $\boldsymbol{x}$. Los parámetros de la combinación son los multiplicadores de Lagrange α_i.
- Sólo un subconjunto será diferente de cero. Sus muestras asociadas son los llamados **vectores soporte**. La solución será dispersa.
- A partir de (29) la máquina puede expresarse como:

$$f(\boldsymbol{x}_j) = \boldsymbol{w}^T \boldsymbol{x}_j + b = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_j + b \tag{33}$$

Resumen del algoritmo

- Tómese un conjunto de muestras etiquetadas $\{\boldsymbol{x}_i, y_i\}$.
- Calcúlese la matriz $\boldsymbol{K}$ de productos escalares.
- Optimícese el funcional

$$L_d = -\frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Y}\boldsymbol{K}\boldsymbol{Y}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha}$$

- Constrúyase la máquina usando la expresión (29) del vector de pesos:

$$f(\boldsymbol{x}_j) = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_j + b \tag{34}$$

Notas

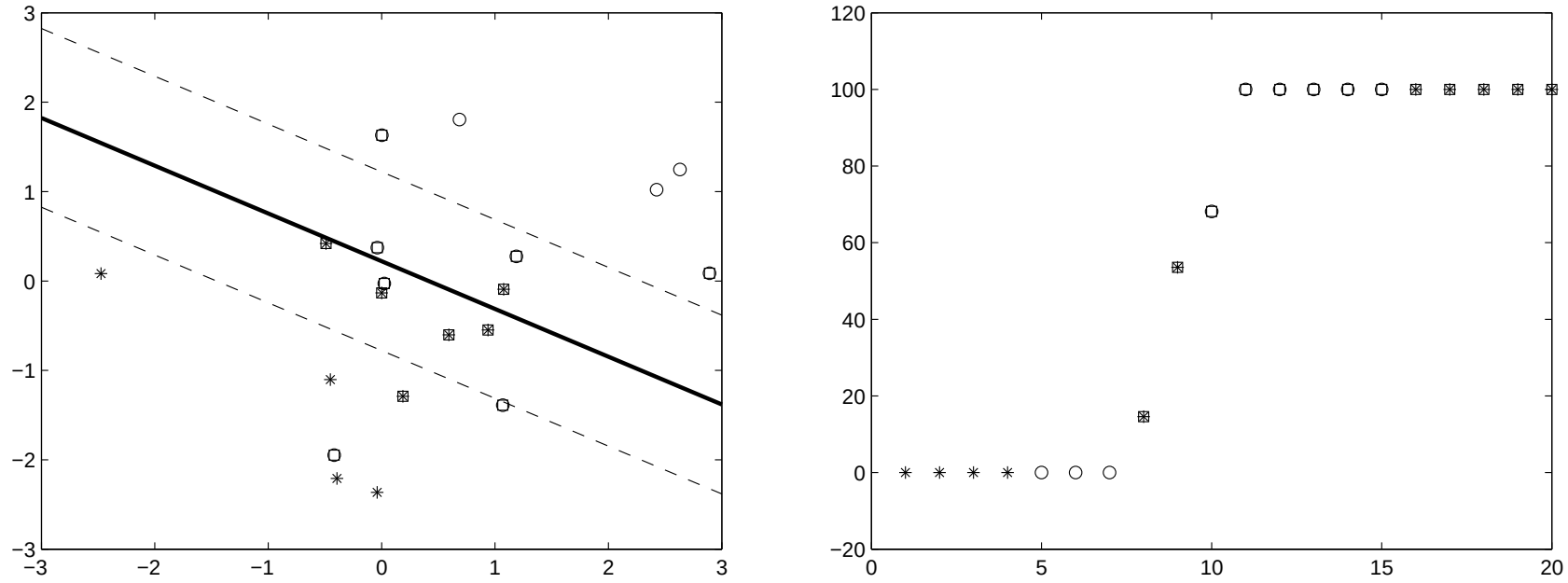
- **SÓLO SE NECESITAN PRODUCTOS ESCALARES**.
- La solución es dispersa.
- Hay existencia y unicidad de solución, ya que

$$L_d = -\frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Y}\boldsymbol{K}\boldsymbol{Y}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha}$$

es una forma cuadrática.

Ejemplos

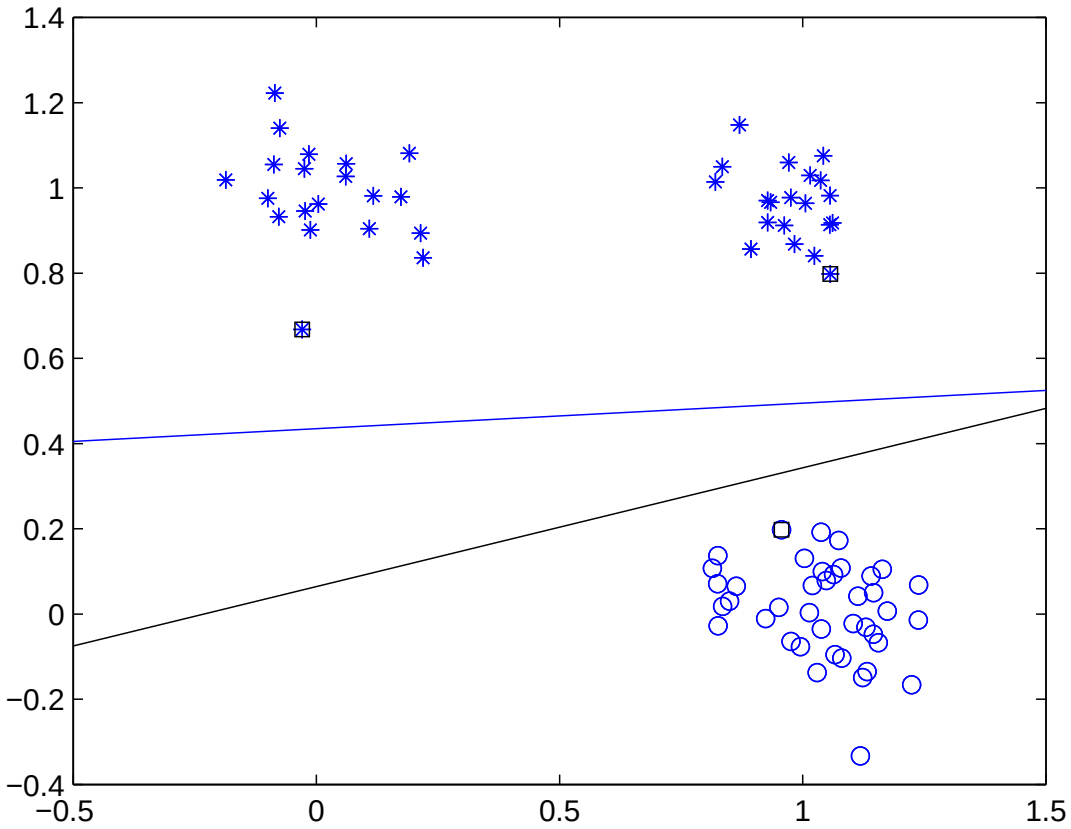
Clasificador lineal



Clasificador SVM lineal y los valores de los multiplicadores de Lagrange.

Ejemplos

Comparación del clasificador lineal SVM y el MMSE



Ejemplos

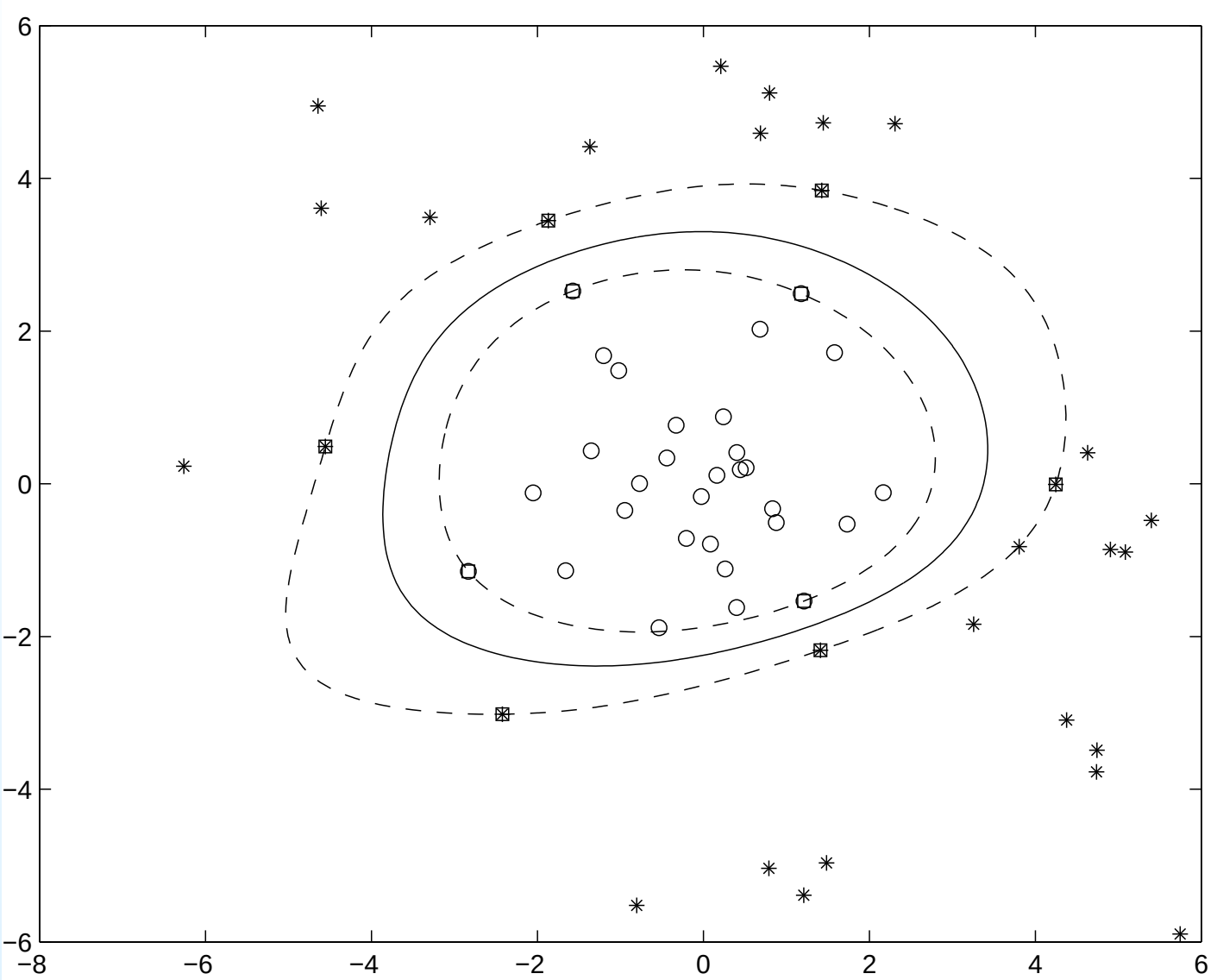
Construir una máquina no lineal es inmediato:
Aplíquese una transformación no lineal $\varphi(\cdot)$ a los datos. El clasificador será

$$y_i = \sum_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \tag{35}$$

La optimización se lleva a cabo de forma idéntica al clasificador lineal, con la salvedad de que

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) \tag{36}$$

Ejemplos



Conclusiones

- El truco de los kernels permite construir versiones no lineales de los algoritmos lineales, dotándolos de mayor potencia.
- Los kernels se interpretan como productos escalares en espacios de alta dimensionalidad.
- Hay kernels con gran versatilidad y se pueden construir kernels compuestos.
- Con el número de dimensiones aumenta la complejidad, y con ella ella el riesgo del estimador.
- Las máquinas de vectores soporte son una alternativa a los algorimos MMSE que controlan la complejidad mediante regularización.
- Presentan existencia y unicidad, al igual que el MMSE
- Se pueden realizar versiones no lineales usando el truco de los kernels, obteniendo máquinas de gran capacidad expresiva, pero con control del sobreentrenamiento.