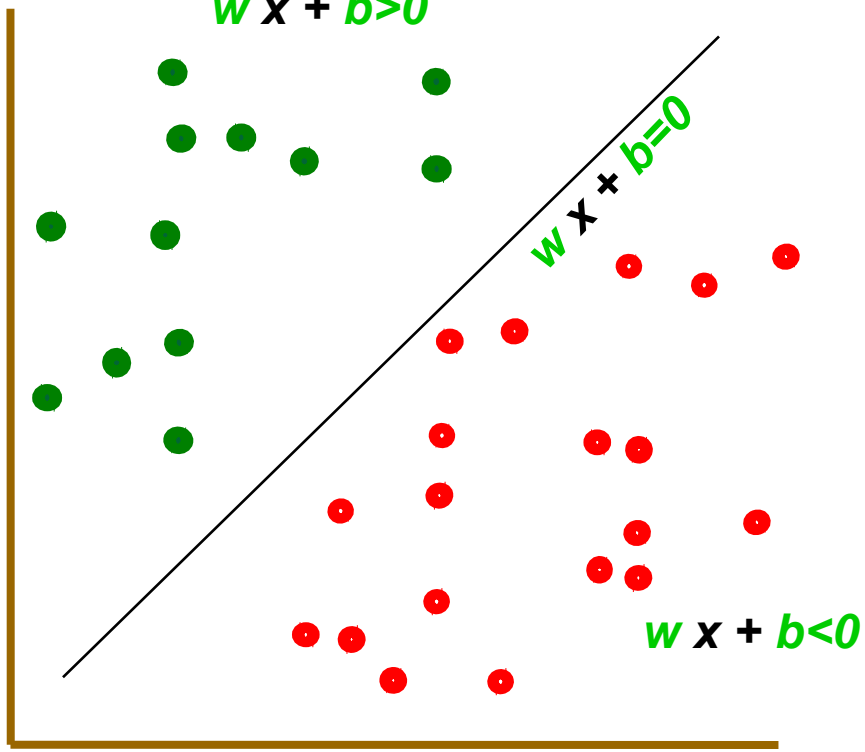
Support Vector Machines

SVM

Support Vector Machines (SVM) - Clasificador Lineal

representa +1

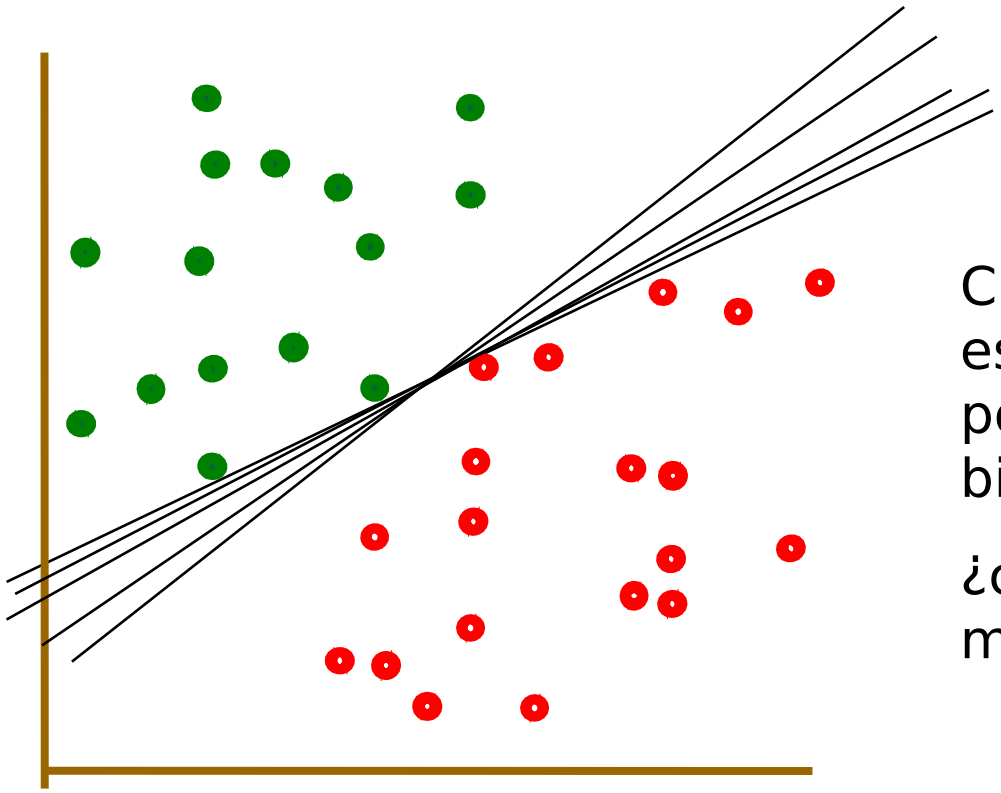
representa -1



$f(x, w, b) = signo(w \cdot x + b)$

Cómo se pueden clasificar estos datos?

SVM- Clasificador lineal



Cualquiera de estas rectas podría estar bien..

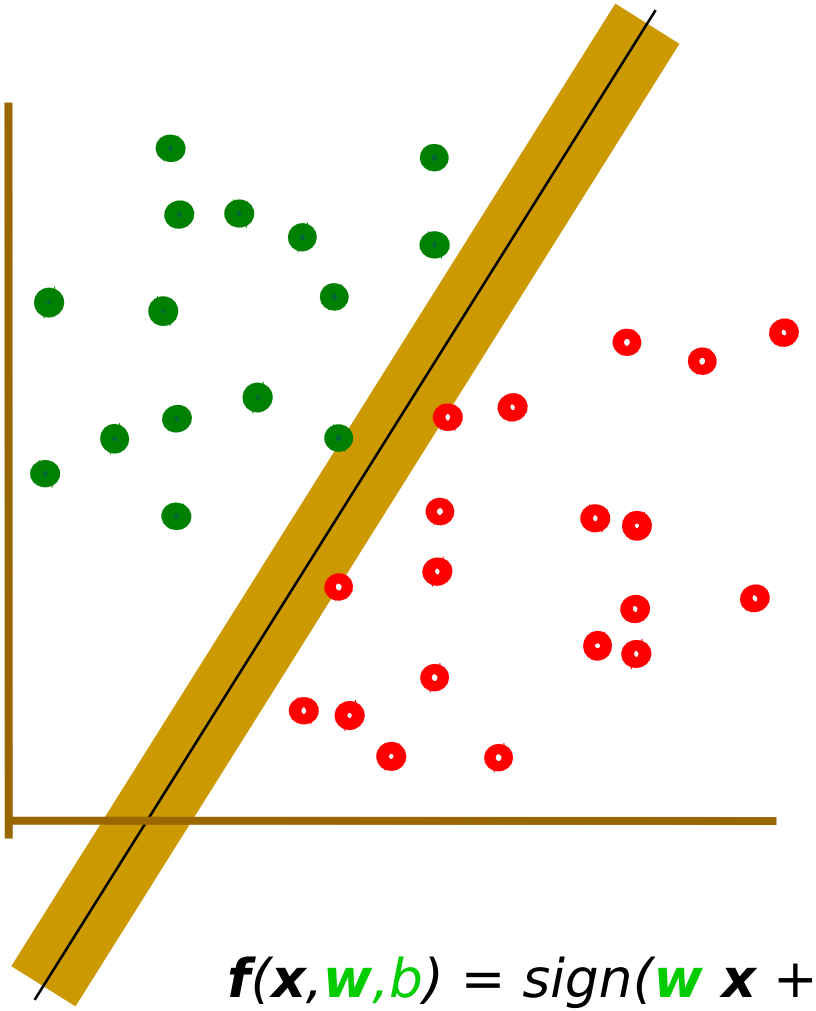
¿cuál es mejor?

$$\boldsymbol{f}(\boldsymbol{x},\boldsymbol{w},b) = signo(\boldsymbol{w}\ \boldsymbol{x} + b)$$

SVM- Clasificador Lineal
Clasificador de máximo margen

representa +1

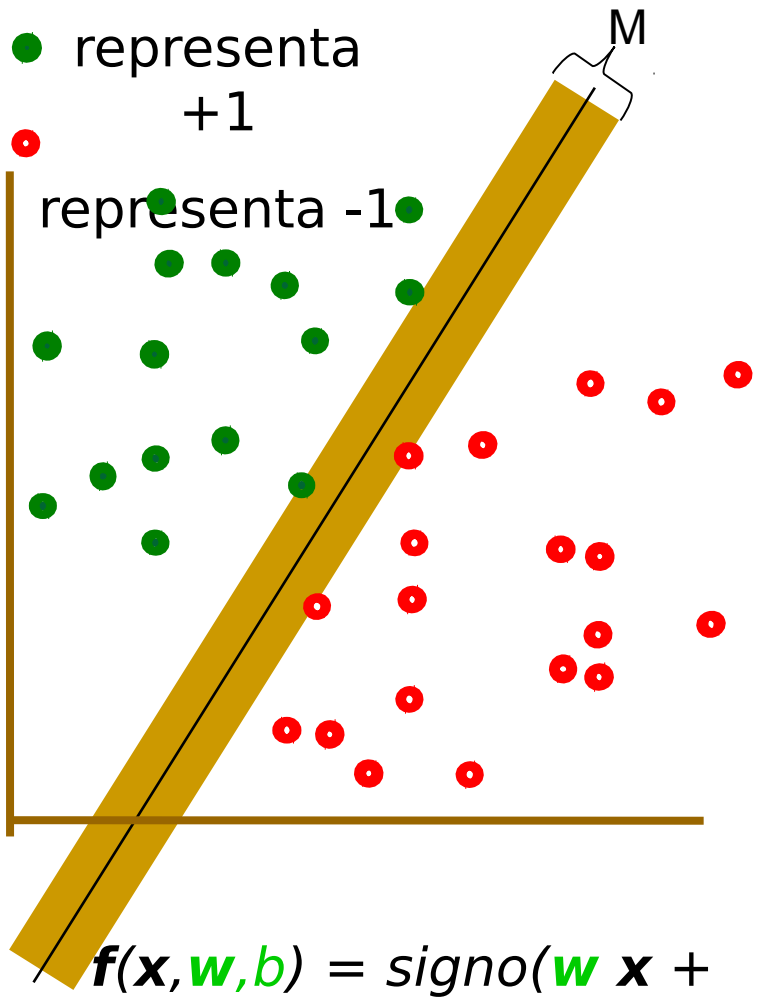
representa -1



SVM lineal:
Encontrar w y b tal que márgen se máximo

$$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}\ \boldsymbol{x} + b)$$

SVM- Clasificador Lineal
Clasificador de máximo margen



Mediante un cambio de variable se puede ver que maximizar M es equivalente a minimizar $\frac{1}{2} w^t w$

De esa forma el problema de optimización se expresa como

Minimizar $\Phi(w) = \frac{1}{2} w^t w$

Sujeto a las restricciones:

$$y_i (w x_i + b) \geq 1 \quad \forall i$$

Solución del clasificador de máximo margen

De las condiciones KKT:

$$y_i(\langle x_i, w^* \rangle + b^*) - 1 \geq 0$$

$$\lambda_i \geq 0$$

$$\lambda_i(y_i(\langle x_i, w^* \rangle + b^*) - 1) = 0$$

$$\nabla L(w^*, b^*, \lambda) = 0, \text{ donde } L(w^*, b^*, \lambda) = \frac{1}{2}||w^*||^2 - \sum_{i=1}^{n} \lambda_i(y_i(\langle x_i, w^* \rangle + b^*) - 1)$$

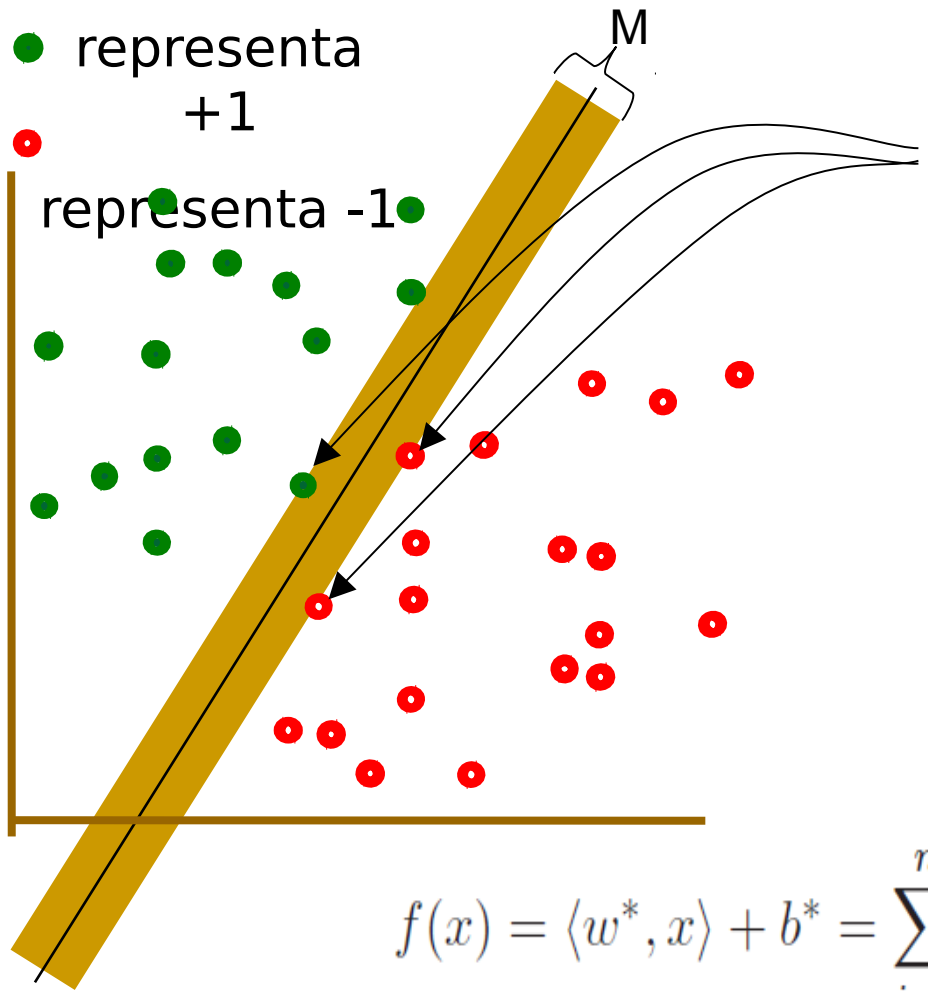
De donde se obtiene:

$$w^* = \sum_{i=1}^{n} \lambda_i y_i x_i$$

$$b^* = \frac{1 - y_{i_0}\langle x_{i_0}, w^* \rangle}{y_{i_0}}$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

Vectores soporte



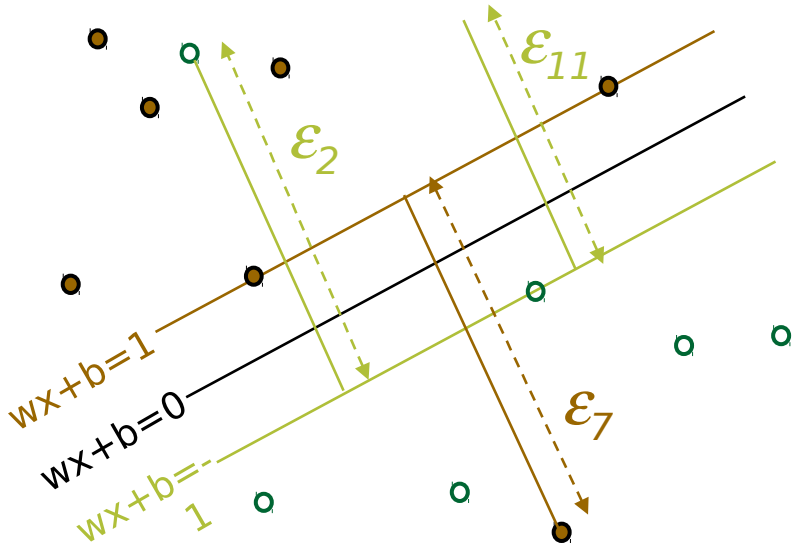
Vectores soporte

nsv: número de vectores soporte,
Aquellos que tienen multiplicador
no nulo

$$f(x) = \langle w^*, x \rangle + b^* = \sum_{i=1}^{n} \lambda_i y_i \langle x_i^*, x \rangle + b^* = \sum_{j=1}^{n_{SV}} \lambda_j y_j \langle x_j, x \rangle + b^*$$

El problema anterior puede no ser factible

Se pueden agregar variables auxiliares ξi que midan el error de los puntos que quedan mal clasificados de datos complejos o con ruido



En este caso el problema de optimización se modifica:

Minimizar $\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\varepsilon_k$

Sujeto a:

$y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

La constante C pondera el peso que se le da a los errores de clasificación.

El caso no separable

- En este caso el Lagrangiano queda:

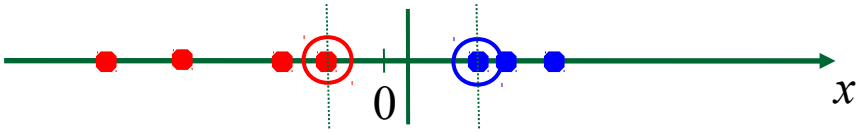
$$L(w, b, \xi, \lambda, \mu) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(\langle x_i, w\rangle + b) - 1 + \xi_i) - \sum_{i=1}^{n}\mu_i\xi_i.$$

- Planteando las condiciones KKT se obtiene una expresión similar a la anterior:

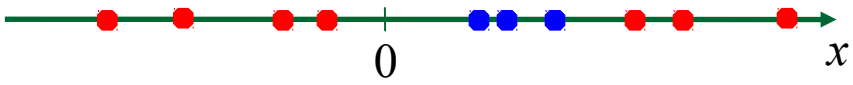
$$f(x) = \sum_{i=1}^{n_{SV}} \lambda_i y_i \langle x_i, x\rangle + b^*$$

SVM en el caso no lineal

- Para conjuntos de datos que se pueden separar linelamente y que contienen ruido funciona correctamente:



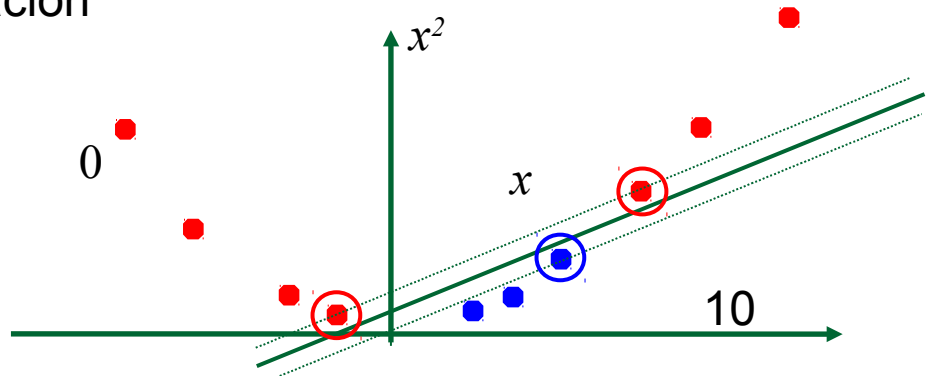
- Pero que sucede si los datos no pueden ser separados linelamente?



- La idea es mapear los datos a un espacio de dimensión mayor:

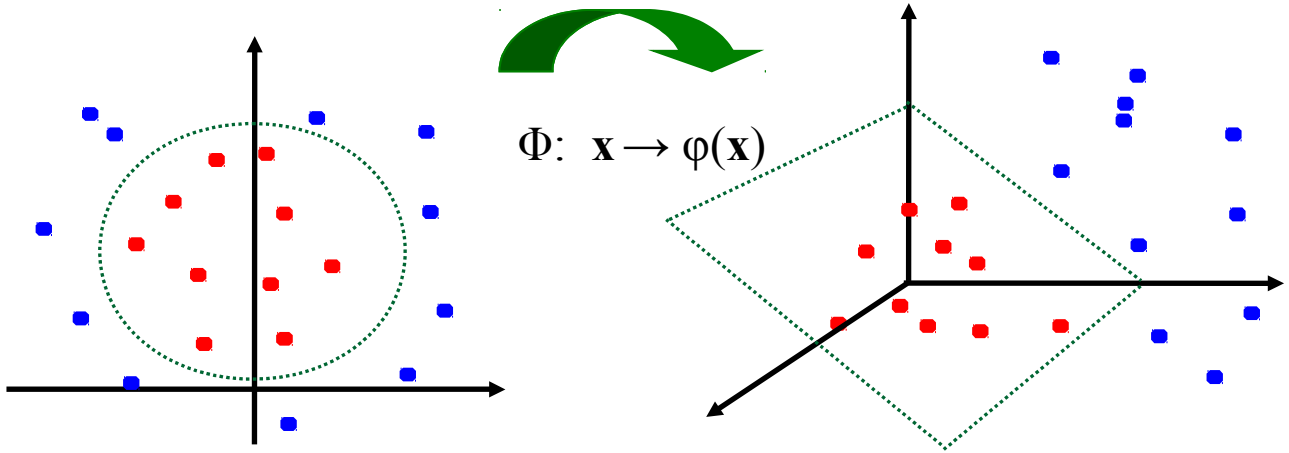
Ej.:Hacemos una transformación y pasamos de la recta a R^2 mediante la transformación

Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})=(x,x^2)$



SVM no lineal: Espacio de características

- Idea general: El espacio original puede siempre ser mapeado en un espacio de dimensión mayor donde la muestra de entrenamiento sea separable linealmente

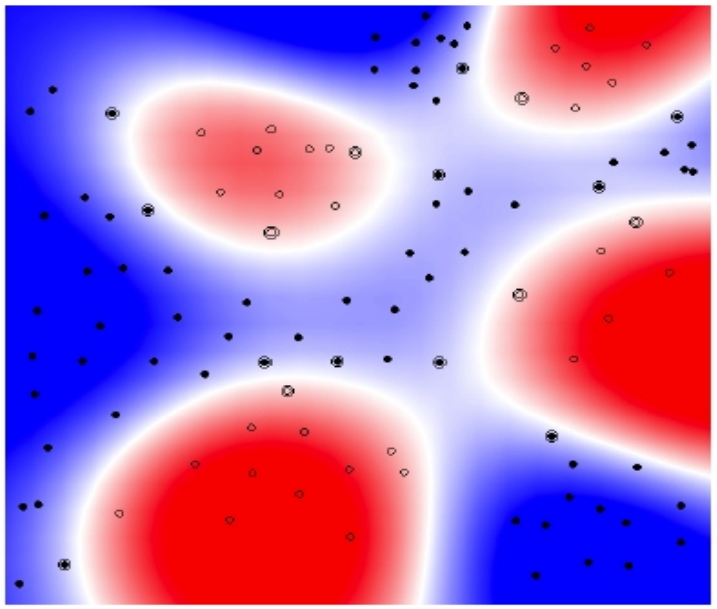


Se puede probar que no es necesario conocer explícitamente la transformación Φ sino que alcanza con conocer la función que permite calcular el producto interno en el espacio de dimensión mayor. Esta función se llama Kernel o Núcleo

Ejemplo de algunas funciones de kernel

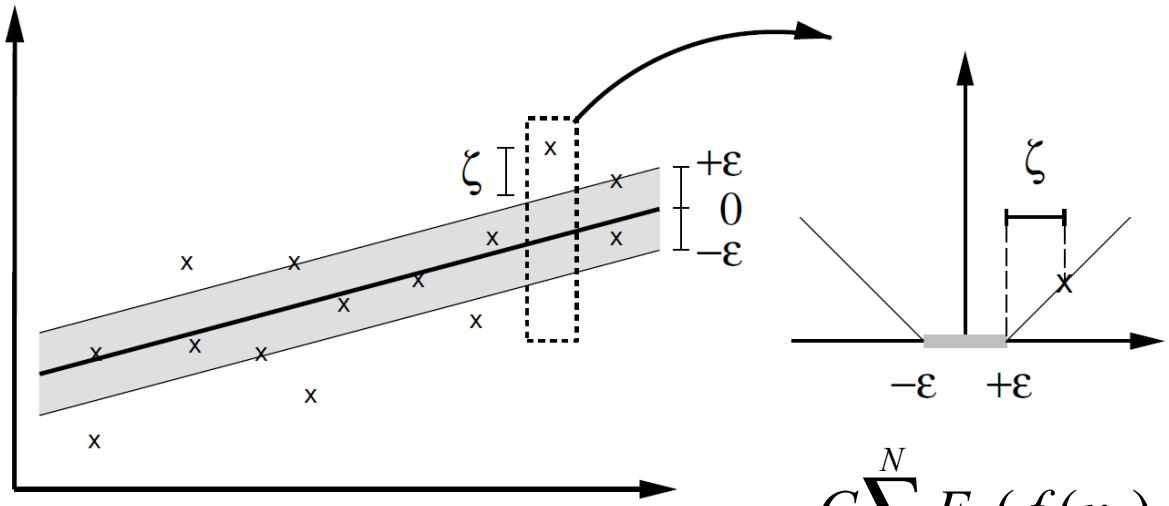
- Lineal: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^T\mathbf{x}_j$
- Polinomial de potencia p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^T\mathbf{x}_j)^p$
- Gaussiana : $$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2})$$

Kernel: $K(\vec{x}_i,\vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2/\sigma^2)$ plot by Bell SVM applet



SVM en regresión

- El problema de optimización se formula de manera similar al caso de clasificación
- Buscar el plano tal que en un tubo de radio ε caigan la mayor cantidad de puntos y penalizar a aquellos que queden fuera.



$$C\sum_{n=1}^{N} E_{\varepsilon}(f(x_n)-y_n)+\frac{1}{2}\|w\|^2$$

Variables auxiliares

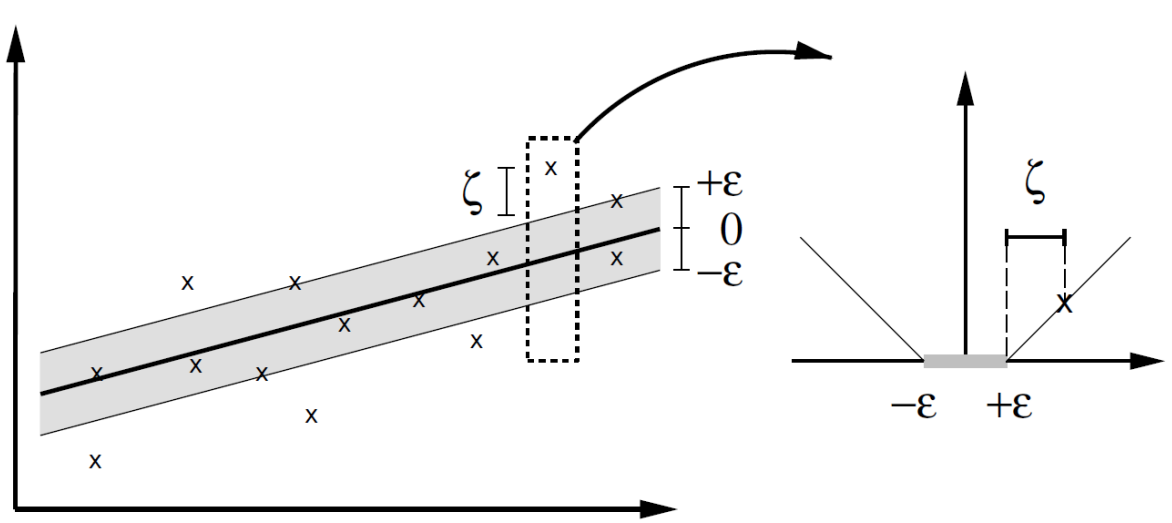
Para que un punto caiga dentro del tubo:

$$f(x_n) - \epsilon \leq y_n \leq f(x_n) + \epsilon$$

Introducimos variables auxiliares para permitirle a los puntos a residir fuera del tubo:

$$y_n \leq f(x_n) + \epsilon + \xi_n$$

$$y_n \geq f(x_n) - \epsilon - \xi_n^-$$



f*(*x*,*w*,*b*) = *w* *x* + *b

Minimizamos la siguiente función de error:

Minimizar:

$$C\sum_{n=1}^{N}(\xi_n + \xi_n^-) + \frac{1}{2}\|w\|^2$$

Sujeto a:

$$\xi_n \geq 0 \quad \text{y} \quad y_n \leq f(x_n) + \in + \xi_n$$

$$\xi_n^- \geq 0 \qquad\qquad y_n \geq f(x_n) - \in - \xi_n^-$$

Construnimos el Lagrangiano

$$L = C\sum_{n=1}^{N}(\xi_n + \xi_n^-) + \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N}(\mu_n\xi_n + \mu_n^-\xi_n^-) - \sum_{n=1}^{N}a_n(\in +\xi_n + y_n - f(x_n)) - \sum_{n=1}^{N}a_n^-(\in +\xi_n^- - y_n + f(x_n))$$

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^{N}(a_n - a_n^-)x_n$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N}(a_n - a_n^-) = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n + \mu_n = C$$

$$\frac{\partial L}{\partial \xi_n^-} = 0 \Rightarrow a_n^- + \mu_n^- = C$$

Karush-Kuhn-Tucker (KKT)

Los vectores soporte son los puntos que se encuentran sobre la frontera del tubo o fuera de él.

$$a_n(\in +\xi_n + y_n - f(x_n)) = 0$$

$$a_n^-(\in +\xi_n^- - y_n + f(x_n)) = 0$$

$$(C - a_n)\xi_n = 0$$

$$(C - a_n^-)\xi_n^- = 0$$

La forma dual del Lagrangiano

Maximizar:

$$W(a,a^-) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - a_n^-)(a_m - a_m^-)(x_n, x_m) - \in \sum_{n=1}^{N}(a_n + a_n^-) + \sum_{n=1}^{N}(a_n - a_n^-)y_n$$

$$0 \le a_n \le C$$

$$0 \le a_n^- \le C$$

La predicciónpuede ser realizada utilizando:

$$y(x) = \sum_{n=1}^{N}(a_n - a_n^-)(x, x_n) + b$$

Referencias

- Está presentación está basada en los siguientes trabajos:

- A Tutorial on Support Vector Regression Alex J. Smola and Bernhard Sch¨olkopf, Statistics and Computing, Volume 14, Number 3, August 2004 , pp. 199-222(24), Springer
- Prof. Andrew Moore's SVM tutorial at http://www.cs.cmu.edu/~awm/tutorials
- INTRODUCTION TO *Machine Learning* ETHEM ALPAYDIN © The MIT Press, 2004
- Vladimir N. Vapnik, The nature of statistical learning theory, Springer-Verlag New York, Inc., New York, NY, 1995
- Support Vector Machine & Its Applications, Mingyue Tan, The University of British Columbia, November 2004