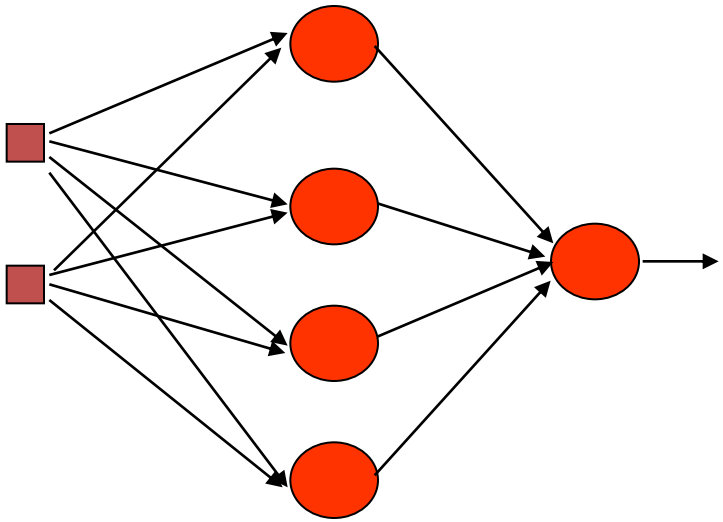
Máquinas de vectores soportes (SVM)

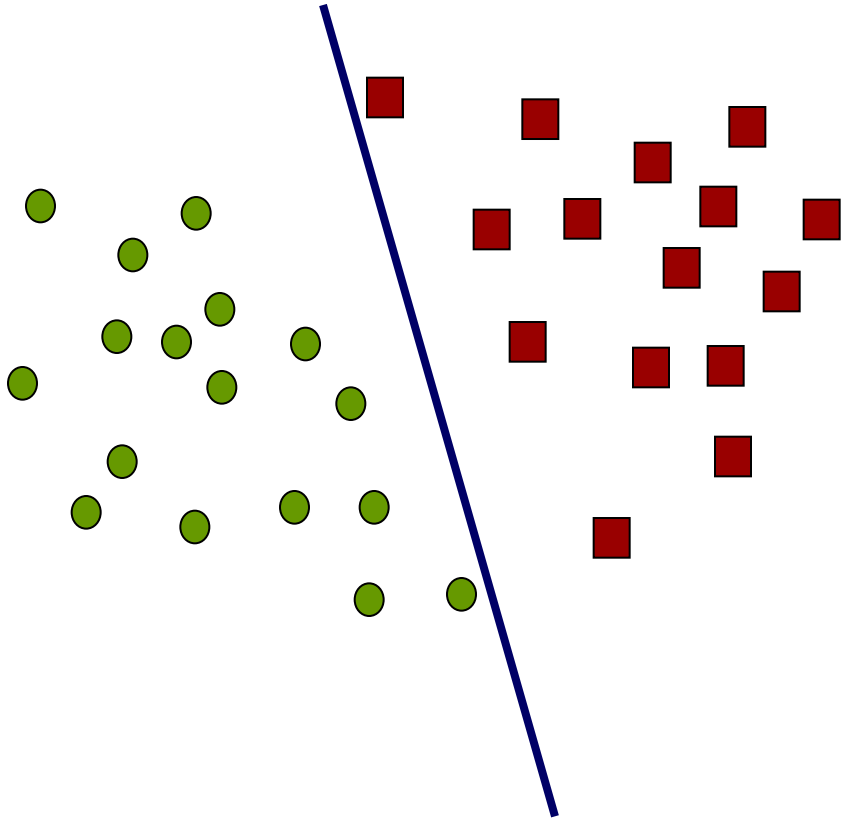


Para el problema de clasificación con datos linealmente separable ….

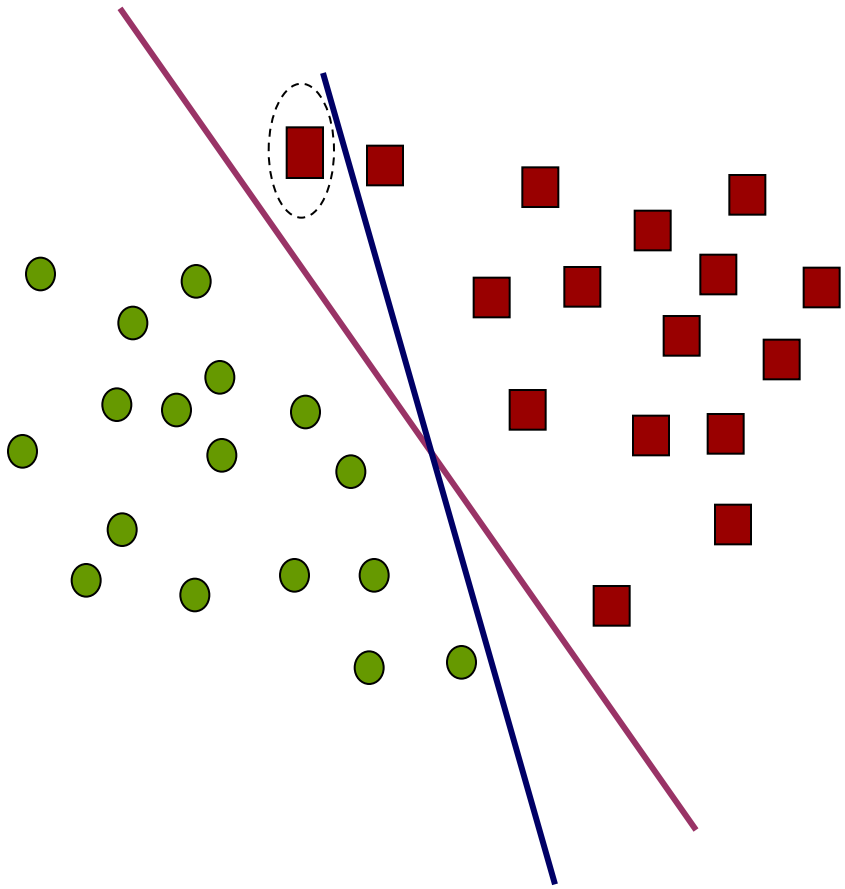
Tenemos varias herramientas a disposición: perceptrón, Adaline, LMS logístico, redes RBF, etc.

Nos quedamos por ahora con el problema linealmente separable.

Volvamos al perceptrón simple …. Un perceptrón busca una regla de decisión que discierna entre dos clases. El algoritmo se detiene cuando encuentra dicha recta.

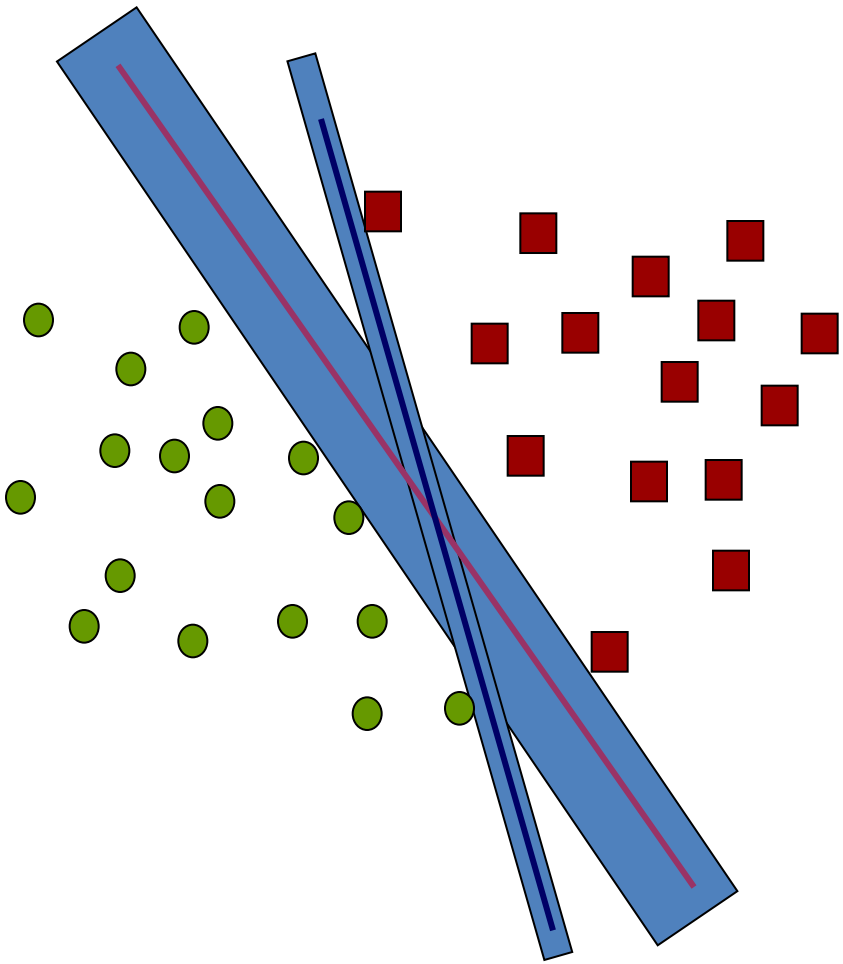


Cada vector de pesos iniciales me lleva a una recta distinta!!

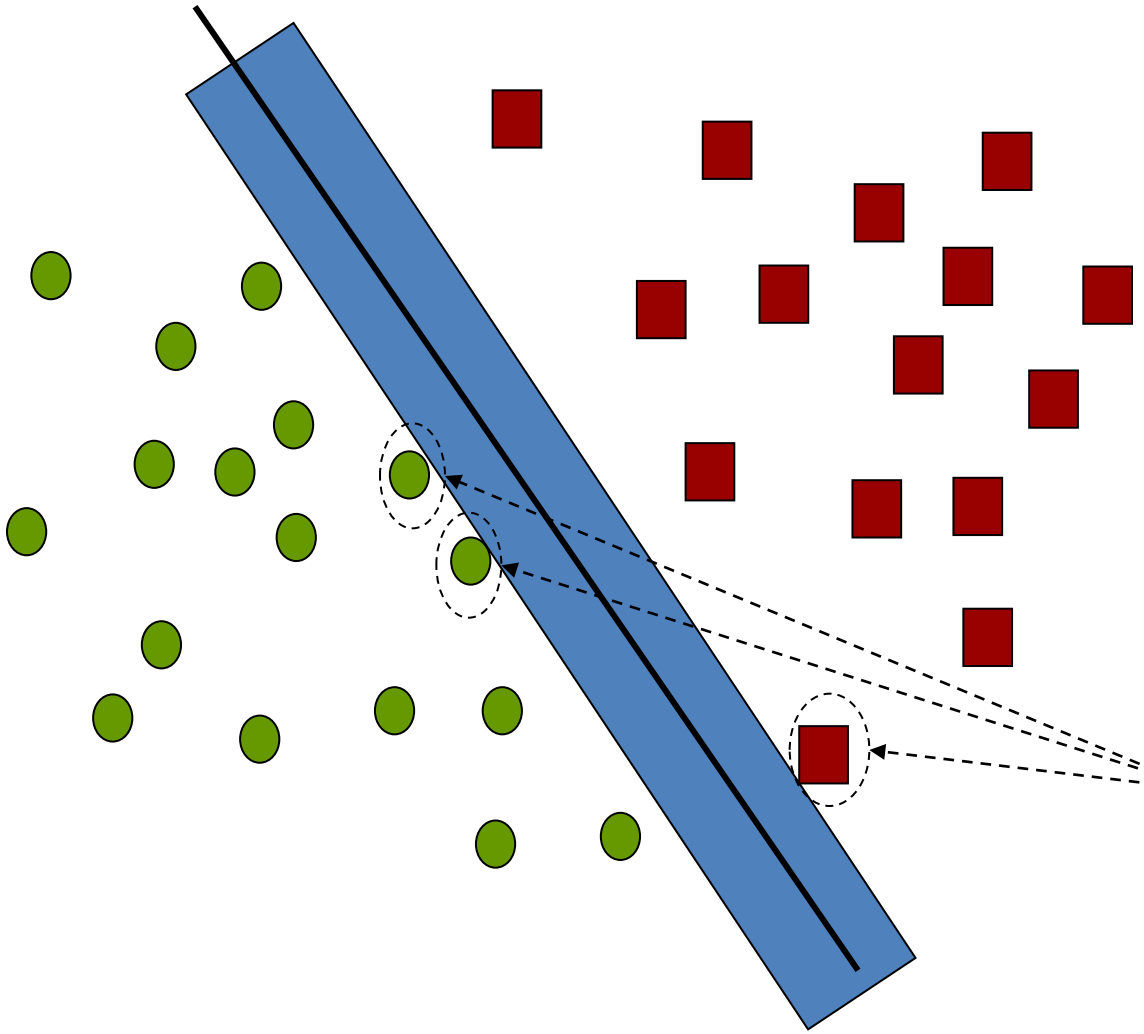


Si pudieran escoger entre estas dos, cual eligen?

Ambas clasiican correctamente, pero la azul no tiene propiedades de generalización adecuadas.



Nos interesa la regla de decisión que deje un mayor márgen de separación entre las dos clases para así mejorar las propiedades de generalización de la regla.

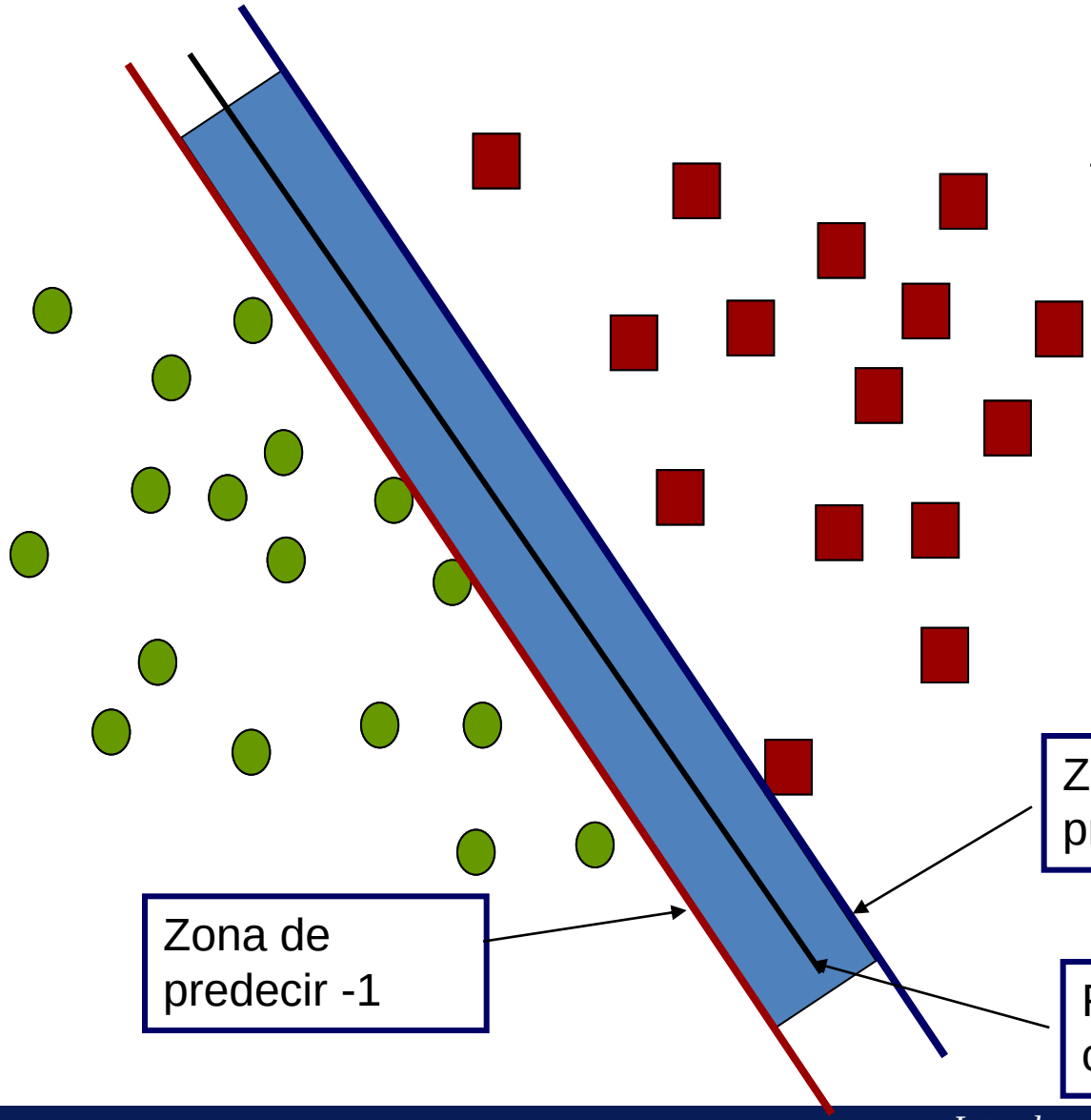


Conocido como el "Linear Support Vector Machine"

Vectores de soporte: aquellos donde se apoya el márgen máximo

Clasificamos:

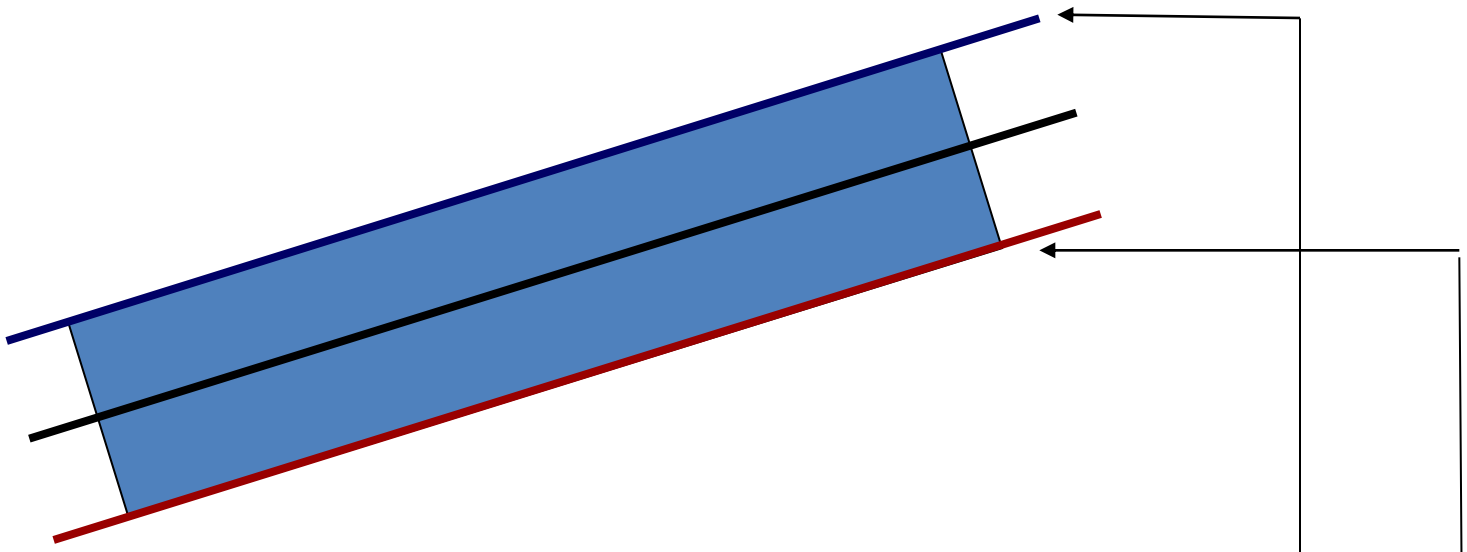
$$1 \text{ si } w^t x + b >= 1$$
$$-1 \text{ si } w^t x + b <= -1$$



Zona de predecir +1

Zona de predecir -1

Frontera del clasificador



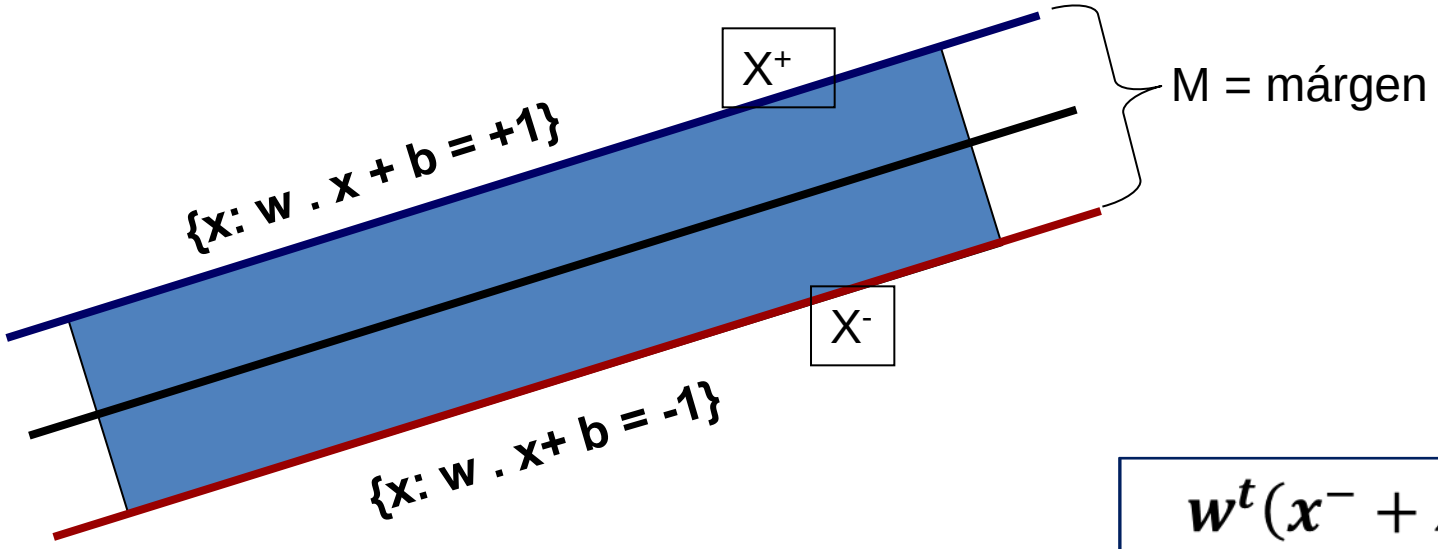
Hiperplano Positivo=$\{x: w^t x + b = 1\}$

Hiperplano Negativo = $\{x: w^t x + b = -1\}$

Clasificamos si

$$+1 \quad si \quad w^t x + b \geq 1$$
$$-1 \quad si \quad w^t x + b \leq -1$$

Queremos que: $-1 < w^t x + b < 1$



W es perpendicular al HP+, Porqué?

Dado un punto x^- en el HP-, existe un x^+ más cercano a x^- en el HP+

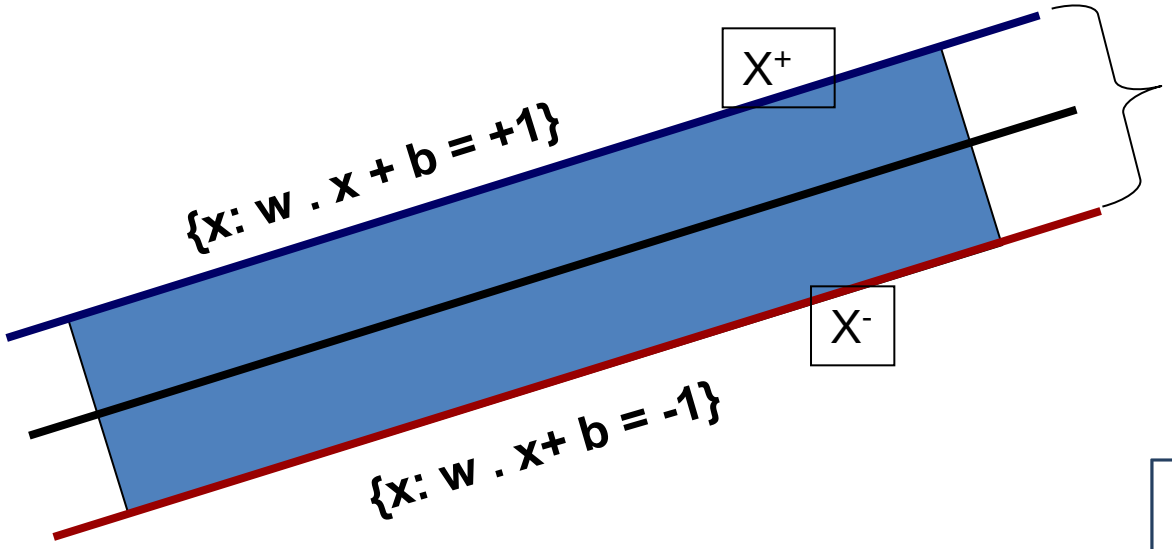
$$\mathbf{x}^+ = \mathbf{x}^- + \lambda\, w$$

$$w^t(x^- + \lambda w) + b = 1$$

$$w^t x^- + b + \lambda w^t w = 1$$

$$-1 + \lambda w^t w = 1$$

$$\lambda = \frac{2}{w^t w}$$



$$\lambda = \frac{2}{w^t w}$$

$$\mathbf{M} = |x^+ - x^-| = |\lambda w| =$$

$$= \lambda |w| = \lambda \sqrt{w^t w} =$$

$$= \frac{2\sqrt{w^t w}}{w^t w} = \frac{2}{\sqrt{w^t w}}$$

Qué tenemos hasta ahora?

Dados w y b, podemos encontrar los dos hiperplanos y el ancho del márgen que los separa.

Qué falta?

Un algoritmo que encuentre los mejores en el espacio de búsqueda

Que tal si lo formulamos matemáticamente?

Si!!

Para cada x_i, denotaremos el indicador de clase con y_i
(y_i = +1 para clase C1 y y_i = -1 para clase C2)

$$\text{Min } J(w) = \frac{1}{2}\|w\|^2$$

$$\text{sujeto a : } y_i\left(w^t x_i + w_0\right) \geq 1, \quad i = 1,2,\ldots,N$$

Minimizar la norma,maximiza el márgen

Las restricciones corresponden a la clasificación correcta de los datos.

Este es un problema de optimización cuadrática, sujeta a restricciones lineales. Las condiciones Karush-Kuhn-Tucker (KKT) son condiciones necesarias (y suficientes en nuestro caso) para optimalidad.

$$\frac{\partial L(w, w_0, \lambda)}{\partial w} = 0$$

$$\frac{\partial L(w, w_0, \lambda)}{\partial w_0} = 0$$

$$\lambda_i \geq 0, \quad i = 1,2,...., N$$

$$\lambda_i \left[y_i \left(w^t x_i + w_0 \right) - 1 \right] = 0, \quad i = 1,2,...., N$$

Con λ_l el i-ésimo multiplicador de Lagrange, y $L(w,w_0,\lambda)$ el Lagrangeano, definido como:

$$L(w, w_0, \lambda) = \frac{1}{2} w^t w - \sum_{i=1}^{N} \lambda_i \left[y_i \left(w^t x_i + w_0 \right) - 1 \right]$$

Combinando lo anterior, tenemos que:

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

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

- **Los multiplicadores de Lagrange pueden ser positivo o cero, esto implica que el vector w, de la solución óptima es una combinación lineal de $N_s < N$ datos (los que están asociados con multiplicadores positivos).**

- **Estos son los vectores de soporte y el clasificador es la máquina de vectores soportes (SVM).**

- **Un multiplicador de Lagrange distinto de cero (restricción activa) indica que los vectores soporte están en alguno de los dos planos (HP+) o (HP-). En otras palabras, estos son los datos más cercanos al clasificador y por ende elementos críticos del conjunto de entrenamiento.**

- **W_0 se puede calcular de la última de las condiciones.**

- **Nuestra función de costo es estrictamente convexa (hessiano, definido positivo) y las restricciones son lineales, asi que las condiciones KKT son también sufientes!! La solución óptima es única.**

- **El problema dual equivalente:**

$$\max \quad L(w, w_0, \lambda)$$

$$\text{s.t.} \begin{cases} w = \sum_{i=1}^{N} \lambda_i y_i x_i \\ \sum_{i=1}^{N} \lambda_i y_i = 0 \\ \boldsymbol{\lambda} = 0 \end{cases}$$

Ahora los vectores de entrenamiento están en la ecuación en una restricción de igualdad (más fácil)

Con algo mas de esfuerzo…..

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i^T x_j\right)$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{N}\lambda_i y_i=0\\ \boldsymbol{\lambda}\geq\mathbf{0}\end{cases}$$

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i^T x_j\right)$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{N}\lambda_i y_i=0\\ \boldsymbol{\lambda}\geq\mathbf{0}\end{cases}$$

$$\min_{\lambda}\left(-\mathbf{e}^t\boldsymbol{\lambda}+\frac{1}{2}\boldsymbol{\lambda}^t\mathbf{Q}\boldsymbol{\lambda}\right)$$

$$\text{s.t.}\begin{cases}\boldsymbol{\lambda}^t\mathbf{y}=0\\ \boldsymbol{\lambda}\geq\mathbf{0}\end{cases}$$

$$Q_{i,j}=y_i y_j x_i^t x_j$$

En realidad el problema dual es cuadrático con Q es positiva semidefinida, así la función objetivo es convexa. El problema tiene una sola restricción y la condición de no-getividad de los multiplicadores.

Esto mejora muchísimo la eficiencia!!

Una vez calaculados los multiplicadores de Lagrange óptimos (solución del problema arriba), calculamos el vector de pesos como:

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

Y el sesgo como:

$$\lambda_i \left[y_i \left(w^t x_i + w_0 \right) - 1 \right] = 0, \quad i = 1,2,\ldots,N$$

Solo se usan los vectores soportes!!

Y si los datos no son linealmente separables? Uh-oh!

Recuerdan HP+ y HP- ?

Hay tres tipos de datos:

1. **Aquellos que caen fuera de la banda y están correctamente clasificados.**
2. **Vectores dentro de la banda correctamente clasificados**

$$0 \leq y_i \left(w^t x + w_0 \right) < 1$$

3. **Vectores mal clasificados:**

$$y_i \left(w^t x + w_0 \right) < 0$$

Todos los casos anteriores los tratamos con el mismo tipo de restricciones, si introducimos variables de relajación (permitir imprecisiones):

$$y_i\left(w^t x + w_0\right) \geq 1 - \xi_i$$

$$y_i\left(w^t x + w_0\right) \geq 1 \Rightarrow \xi_i = 0$$

$$0 \leq y_i\left(w^t x + w_0\right) < 1 \Rightarrow 0 < \xi_i \leq 1$$

$$y_i\left(w^t x + w_0\right) < 0 \Rightarrow \xi_i > 1$$

El Problema de optimización es un poquito más complicado, pero tiene el mismo sabor…

Queremos maximizar el márgen y al mismo tiempo mantener el número de puntos con variables de relajación lo más pequeñas posibles.

$$\min \quad J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{s.t. } y_i\left[w^t x_i + w_0\right] \geq 1 - \xi_i, \quad i = 1,2,\ldots N$$

$$\xi_i \geq 0, \quad i = 1,2,\ldots N$$

En este caso, el lagrangiano es:

$$L(w, w_0, \xi, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \mu_i \xi_i - \sum_{i=1}^{N} \lambda_i \left[y_i\left(w^t x_i + w_0\right) - 1 + \xi_i\right]$$

Las Condiciones KKT son:

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

$$\sum_{i=1}^{N} \lambda_i y_i$$

$$C - \mu_i - \lambda_i = 0, \;\; i = 1,2,.....N$$

$$\lambda_i \left[y_i \left(w^t x_i + w_0 \right) - 1 + \xi_i \right] = 0, \quad i = 1,2,.....N$$

$$\mu_i \xi_i = 0, \quad i = 1,2,.....N$$

$$\mu_i \geq 0, \quad \lambda_i \geq 0, \quad i = 1,2,.....N$$

La representación dual:

$$\text{Maximizar} \quad L(w, w_0, \lambda, \xi, \mu)$$

$$\text{sujeto a} \begin{cases} w = \sum_{i=1}^{N} \lambda_i y_i x_i \\ \sum_{i=1}^{N} \lambda_i y_i = 0 \\ C - \mu_i - \lambda_i = 0, \quad i = 1,2,.....N \\ \mu_i \geq 0, \quad \lambda_i \geq 0, \quad i = 1,2,.....N \end{cases}$$

Sustituyendo las restricciones de igualdad en el lagrangiano tenemos:

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i-\frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j x_i^T x_j\right)$$

$$\text{s.t.}\quad 0\le\lambda_i\le C,\quad i-1,2,...,N$$

$$\sum_{\mathrm{i}=1}^{\mathrm{N}}\lambda_{\mathrm{i}} y_i=0$$

$$\min_{\lambda}\left(-\mathbf{e}^{\mathbf{t}}\boldsymbol{\lambda}+\frac{1}{2}\boldsymbol{\lambda}^{\mathbf{t}}\mathbf{Q}\boldsymbol{\lambda}\right)$$

$$\text{s.t.}\begin{cases}\boldsymbol{\lambda}^{\mathbf{t}}\mathbf{y}=0\\ 0\le\boldsymbol{\lambda}\ge C\end{cases}$$

Sin relajación

$$\min_{\lambda}\left(-\mathbf{e}^{t}\boldsymbol{\lambda}+\frac{1}{2}\boldsymbol{\lambda}^{t}\mathbf{Q}\boldsymbol{\lambda}\right)$$

$$\text{s.t.}\begin{cases}\boldsymbol{\lambda}^{t}\mathbf{y}=0\\ \boldsymbol{\lambda}\geq\mathbf{0}\end{cases}$$

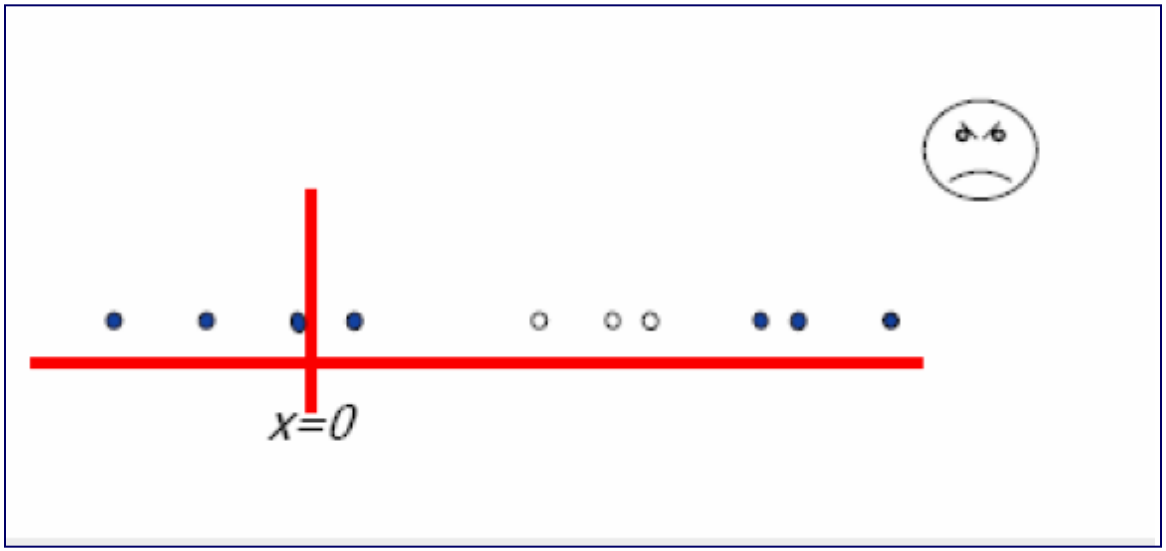
Con relajación

$$\min_{\lambda}\left(-\mathbf{e}^{t}\boldsymbol{\lambda}+\frac{1}{2}\boldsymbol{\lambda}^{t}\mathbf{Q}\boldsymbol{\lambda}\right)$$

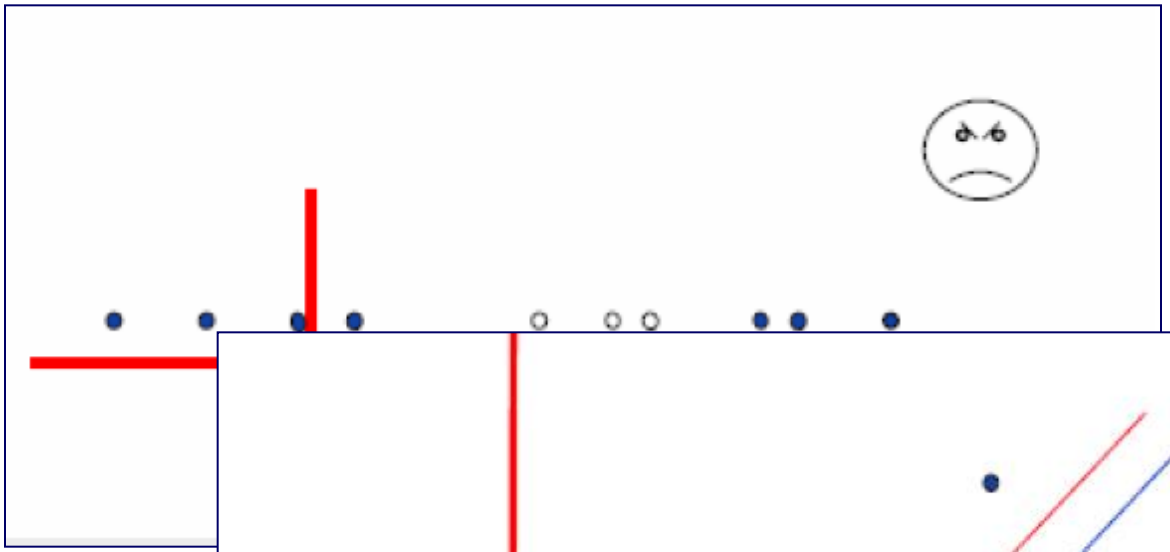
$$\text{s.t.}\begin{cases}\boldsymbol{\lambda}^{t}\mathbf{y}=0\\ 0\leq\boldsymbol{\lambda}\geq C\end{cases}$$

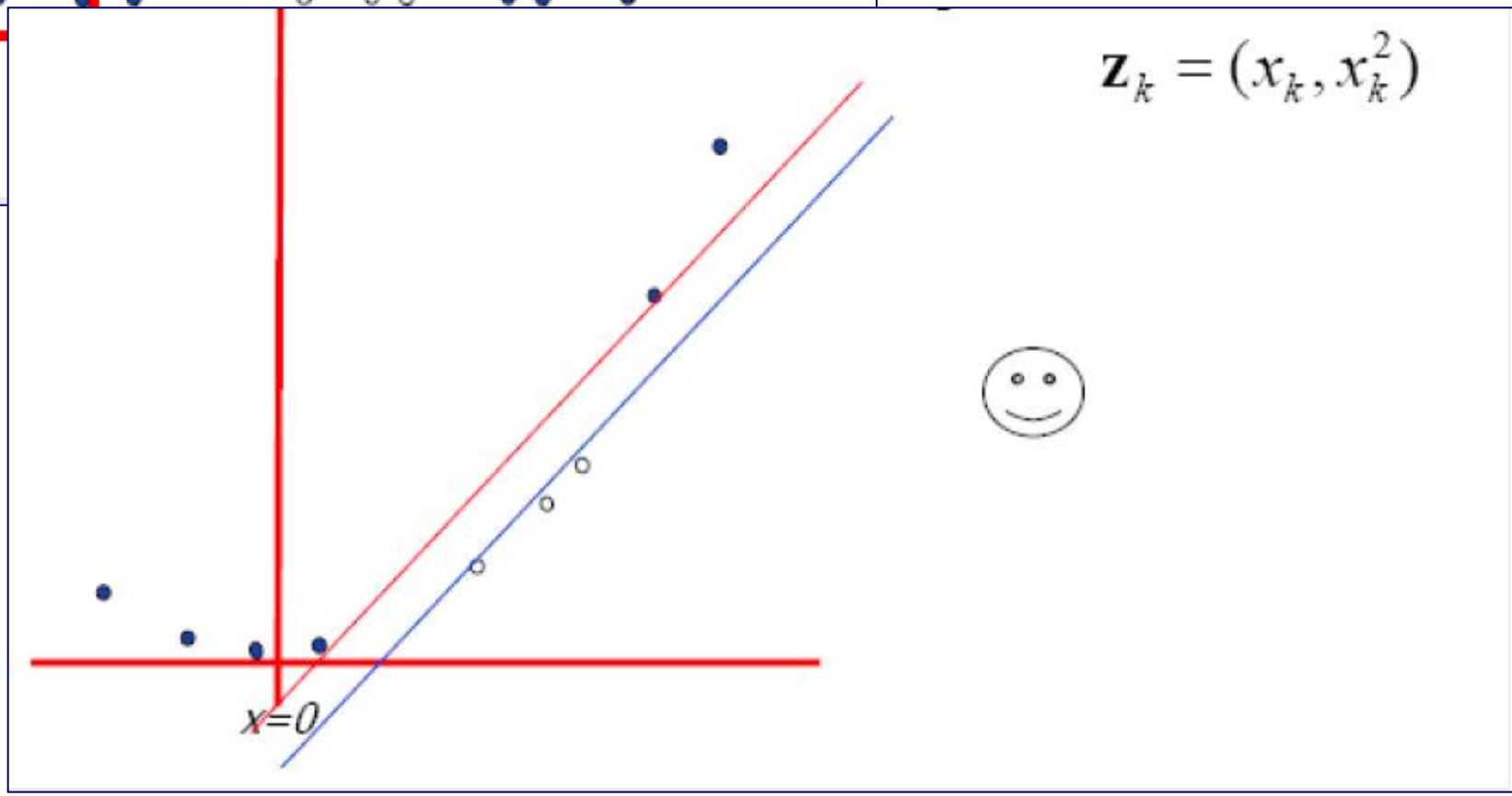
La única diferencia es que acotamos los lambdas!!

Este problema sigue siendo sencillo de resolver.

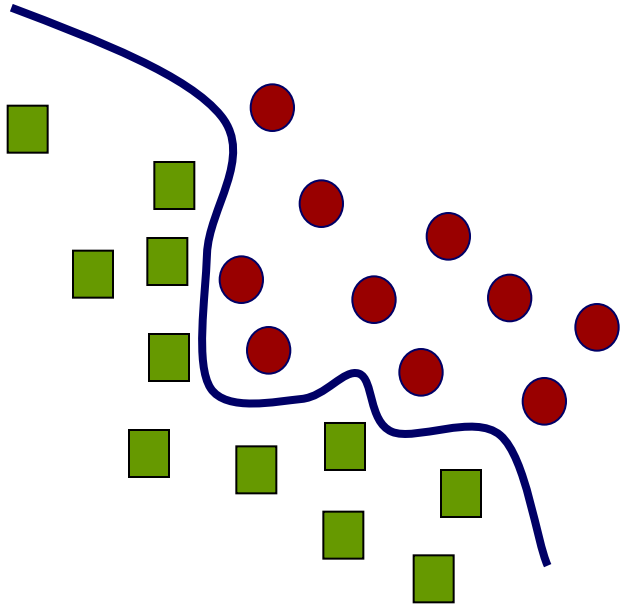


Transformemos la data y subimos la dimensión!!

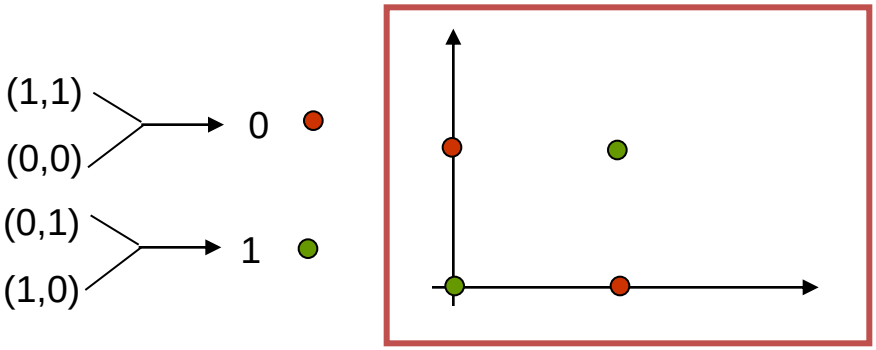



$\mathbf{z}_k = (x_k, x_k^2)$
x=0

Que pasa cuando tenemos una frontera que no es lineal?



Recuerdan el caso del O-exclusivo con una RBF?

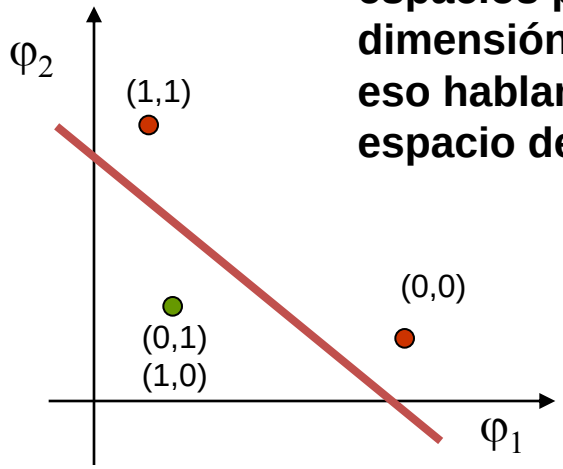


Sean $\varphi_1(x) = \exp(-||x-x_1||)$

$\varphi_2(x) = \exp(-||x-x_2||)$

Cristianini sugiere que los kernels suben la dimension y siempre hay un espacio donde ocurre que los datos sean linealmente separables pero estos espacios pueden tener dimensión infinita y por eso hablamos de un espacio de Hilbert.

	φ_1	φ_2
(1,1)	1	0.1353
(0,1)	0.3678	0.3678
(0,0)	0.1353	1
(1,0)	0.3678	0.3678



El viejo truco del Kernel!

- La idea es transformar los datos de un espacio original hasta un espacio de Hilbert H de mayor dimensionalidad, donde posiblemente sea más fácil usar el SVM

- Esta transformación debe preservar la dependencia de los datos con los productos internos.

Problema Cuadrático:

$$min\ \frac{1}{2}\|w\|^2 + C\left(\sum_{k=1}^{l}\xi_k\right)$$
$$s.a\ y_k[w^t x + b] \geq 1 - \xi_k, \forall k$$

Problema Dual:

$$max\ \sum_{k=1}^{l}\lambda_k - \frac{1}{2}\sum_{k=1}^{l}\sum_{j=1}^{l}\lambda_j\lambda_k y_k y_j k(x_k, x_j)$$
$$s.a.\ 0 \leq \lambda_k \leq C,\ k = 1, \ldots., l.$$

La complejidad no varía mucho, ahora $Q_{i,j} = y_i y_j k(x_i, x_j)$ en vez de $Q_{i,j} = y_i y_j x_i^t x_j$

Qué es un Kernel? Cuales son las condiciones para que una función sea un Kernel?

Condiciones de Mercer:

La función de Kernel K(x,y) cumple con:

$$\int K(x,y)g(x)g(y)dxdy \geq 0$$

Para cualquier g(x), tal que:

$$\int g(x)^2 dx \quad \text{es finita}$$

Ya hemos visto formas de construir kernels a partir de otros

$$
\begin{aligned}
k(x,x') &= ck_1(x,x') \\
k(x,x') &= f(x)k_1(x,x')f(x') \\
k(x,x') &= q(k_1(x,x')) \\
k(x,x') &= exp(k_1(x,x')) \\
k(x,x') &= k_1(x,x') + k_2(x,x') \\
k(x,x') &= k_3(\varphi(x),\varphi(x')) \\
k(x,x') &= k_1(x,x')k_2(x,x')
\end{aligned}
$$

Algunos comunes:

Gausianas: $k(x,y) = \exp(-\|x-y\|^2/\sigma^2)$

Muy no-lineal, buena para separar cosas bien mezcladas

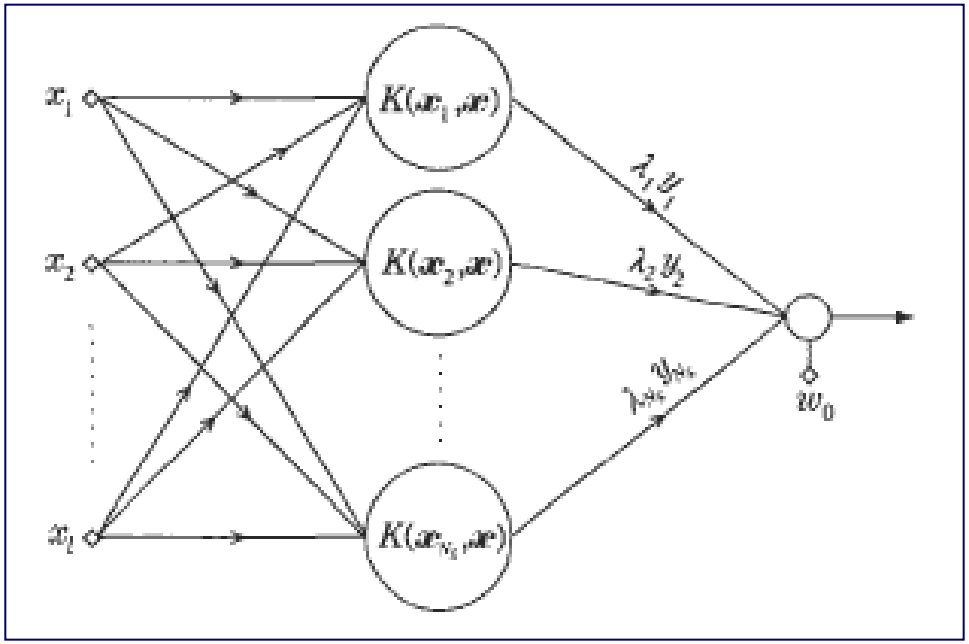
Polinomiales: $k(x,y) = (x^t y + 1)^d$

Dependiendo de d se hace más no-lineal

Sigmoidales: $k(x,y) = \tanh(k(x,y) + \Theta)$

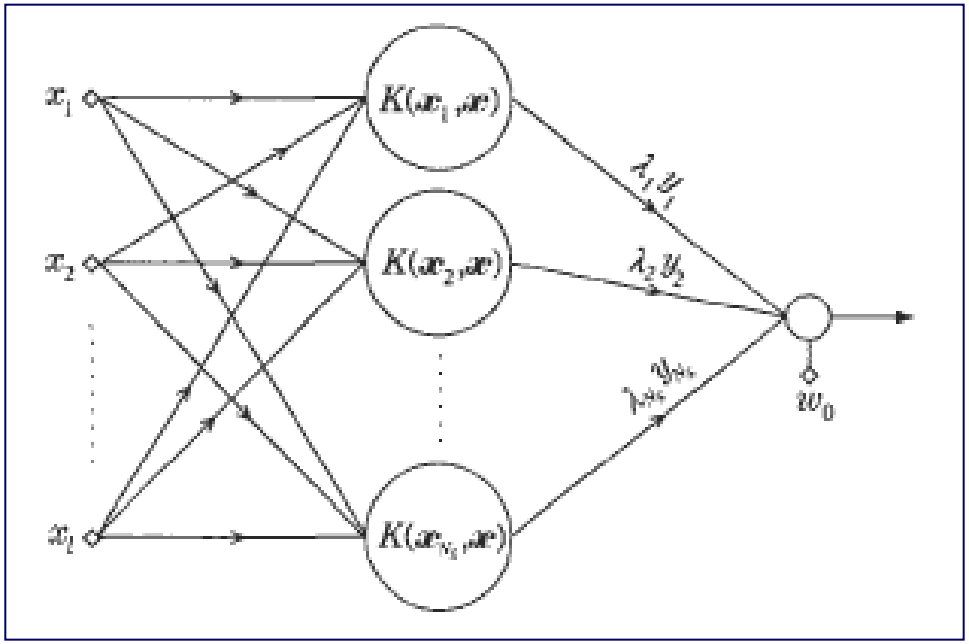
Con esta elección podemos aproximar una MLP

La arquitectura de un SVM, via Kernels



- **Si el Kernel es Gausiano, tenemos la arquitectura de una RBF, pero existe una diferencia: aqui el número de nodos y los centros se calculan como el resultado de un problema de optimización.**

La arquitectura de un SVM, via Kernels



- **Si el Kernel es sigmoide, tenemos la arquitectura de una RNA (N_s x 1) , pero aqui también el número de nodos es producto del proceso de optimización (el proceso de poda incorporado).**

Una mala noticia es que no hay una manera práctica de elegir el Kernel.

Para problemas con pocos datos el entrenamiento y validación no es un problema mayor y cualquier algoritmo puede ser usado. Pero cuando hay muchos datos la cosa cambia.

El problema dual es muy estudiado en los libros texto, la dificultad radica en que $Q \in R^{NxN}$, si $N \gg 1$ entonces tenemos una matriz grande. Si Q fuese rala *(sparce)* no sería tan problematico, pero en estos problemas la matriz Q es densa.

Mientras más datos tengamos mejor es el desempeño para la clasificación (tenemos más ejemplos de donde aprender).

Existen varios métodos de descompiosición de la matriz y del problema a resolver. Uno de los primeros métodos es propuesto por Osuna en el 96 (posible tema para proyecto….)

Existen muchas librerías disponibles para resolver estos problemas y los detalles de los métodos es tema de interés de un curso que se ofrece en el departamento.