Support Vector Machines

Edgar Acuna
Departamento de Ciencias Matematicas
Universidad de Puerto Rico en Mayaguez

SVMs en su forma lineal fue introducido por Vapnik en 1963. (ahora en Facebook). SVM no lineales fueron introducidos por Boser, Guyon and Vapnik en1992 y se hicieron populares a los finales de los 1990s.

SVM es uno de los mejores clasificadores para datos de Bioinformatica y datos textuales.

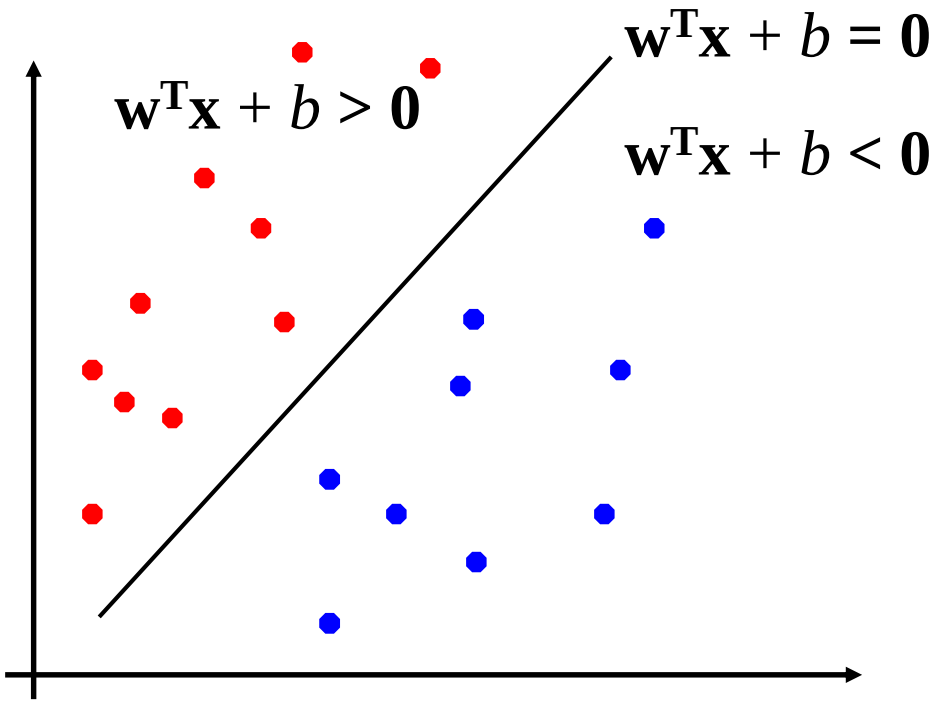
SVMs pueden ser aplicados a diversos tipos de datos complejos disenando adecuadamente funciones kernels.

Aplicacion de SVM ha sido extendida a otras tareas tales como regresion [Vapnik *et al.* ’97], y analisis de componentes principales [Schölkopf *et al.* ’99], etc.

El afinamiento de SVMs tales como seleccionar el kernel y parametros del modelo es hecho heuristicamente.

Separadores Lineales

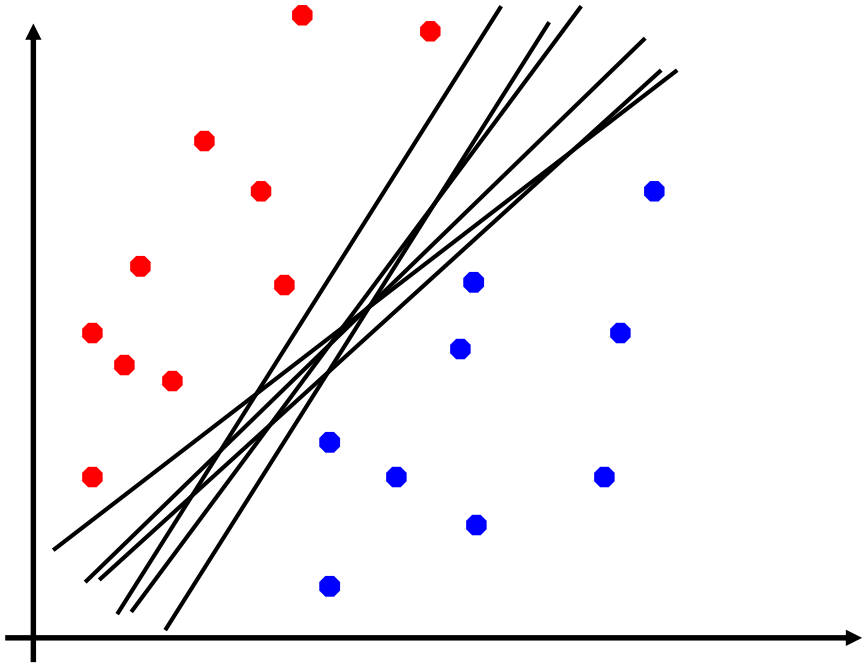
- Clasificacion binaria puede ser vista como la tarea de separar clases en el espacio de los atributos.



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

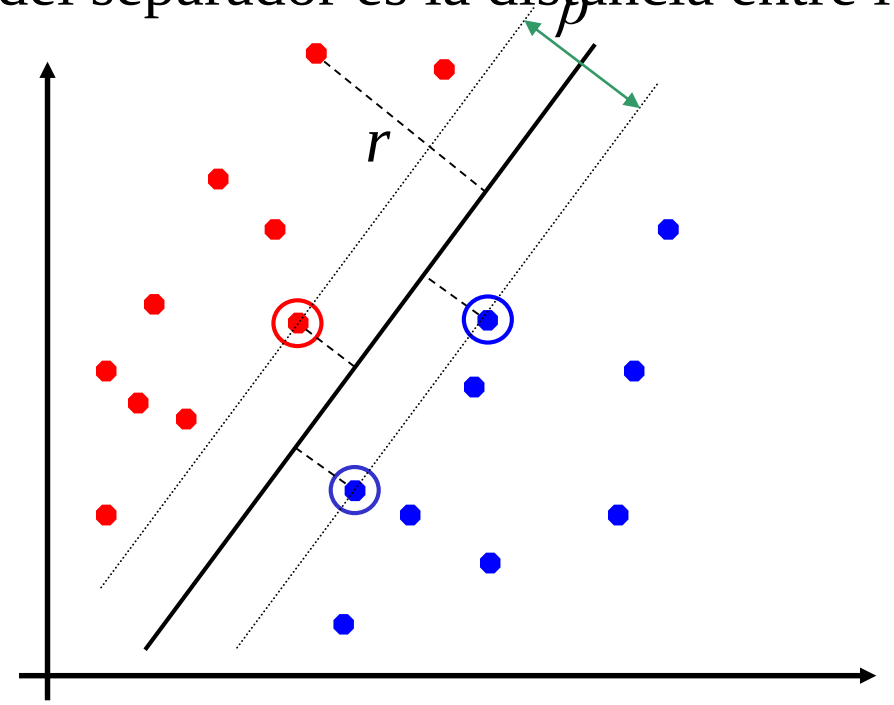
Separadores Lineales

Cual de estos separadores lineales es optimo?



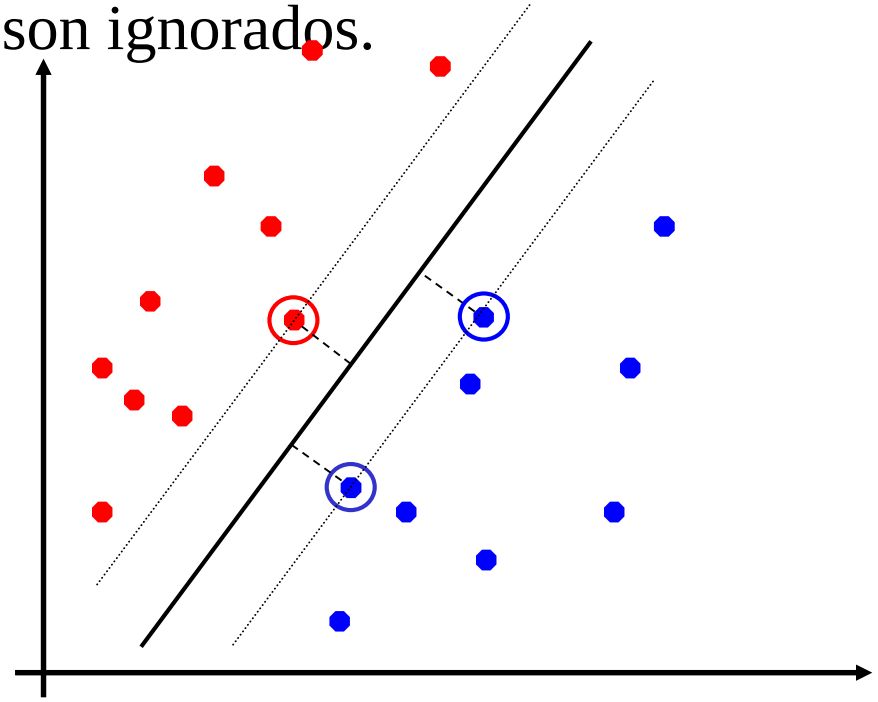
Classificacion por el Margen

- La distancia del ejemplo $\mathbf{x}_i$ al separador es $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Los puntos mas cercanos al hiperplano son llamados vectores de suporte.
- ***El Margen*** ρ del separador es la distancia entre los vectores de soporte.



Clasificacion maximizando el Margen

- Intuitivamente es bueno maximizar el margen. Esto tambien es justificado por la teoria PAC (Probablemente aproximadamente correcto).
- Implica que solamente los vectores de soporte importan, los otros puntos son ignorados.



Formalizacion de SVM Lineal

- Sea el conjunto de entrenamiento $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbf{R}^d$, $y_i \in \{-1, 1\}$ que puede ser separado por un hiperplane con margen ρ. Entonces, para cada instancia $(\mathbf{x}_i, y_i)$:

$$\mathbf{w}^T\mathbf{x}_i + b \leq -\rho/2 \quad \text{if } y_i = -1$$
$$\mathbf{w}^T\mathbf{x}_i + b \geq \rho/2 \quad \text{if } y_i = 1 \quad \Leftrightarrow \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho/2$$

- Para cada vector de soporte $\mathbf{x}_s$ la desigualdad anterior es una igualdad. Despues de re-escalar $\mathbf{w}$ y b por $\rho/2$ en la igualdad, se obtiene que la distancia entre cada $\mathbf{x}_s$ y el hiperplano es:

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

- Luego el margen puede ser expresado como:

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

Formalizacion del SVM lineal (cont.)

- Luego, se puede formular el problema de optimizacion cuadratica:

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

$\rho = \frac{2}{\|\mathbf{w}\|}$ es maximizado

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

El cual puede ser reformulado como:

Hallar $\mathbf{w}$ y b tal que:

$\mathbf{\Phi}(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w}^T\mathbf{w}$ es minimizado

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

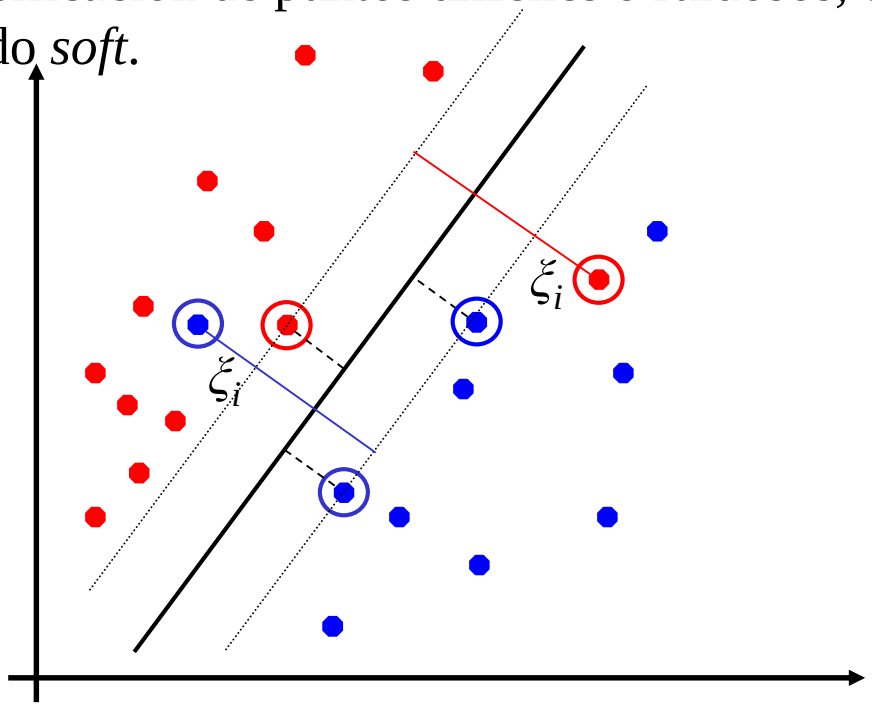
Hallar **w** and b tal que
$\mathbf{\Phi(w)} = \mathbf{w}^T\mathbf{w}$ es minimizado
Para todo $(\mathbf{x}_i, y_i)$, $i=1..n$: $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- Hay que optimizar una function cuadratica sujeta a restricciones lineales.
- Los problemas de optimizacion cuadratica son problemas de programacion matematica para el cual existen varios algoritmos (no triviales).
- La solucion envuelve construir un problema dual donde el multiplicador de Lagrange α_i *esta asociado* con toda restriccion de desigualdad en el problema:

Hallar $\alpha_1 \ldots \alpha_n$ tal que:
$\mathbf{Q(\alpha)} = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$ es maximizado y
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ para todo α_i

Classificacion usando margen suave

- Que pasa si el conjunto de entrenamiento no es linealmente separable?
- *En este caso se anaden variables de holgura (Slack variables* ξ_i) para permitir la clasificacion de puntos dificiles o ruidosos, esto da a lugar a un margen llamado *soft*.



Clasificacion por margen suave-Formulacion Matematica

- La anterior formulacion:

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

- Puede ser modificada para incluir variables de holgura asi:

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

- El parametro C es usado para controlar el overfitting, balancea la importancia relativa de maxmizar el margen y ajustar la data de entrenamiento.

Repaso del SVM lineal

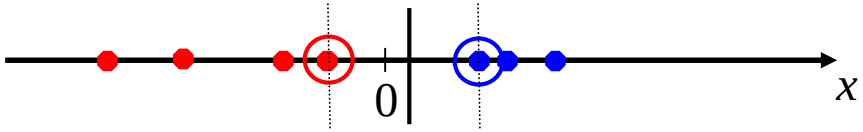
- El clasificador es un *hiperplano separante.*
- Los puntos de entrenamiento mas importantyes son los vectores de soporte; ellos definen el hiperplano.
- Los algoritmos de optimizacion cuadratica pueden identificar que puntos de entrenamiento $\mathbf{x}_i$ son vectores de soporte con multiplicadores de Lagrange distintos de zeros α_i**.** distintos de zeros.
- Tanto en la formulacion del problema dual como en la solucion, los puntos de entrenamiento solo aparecen dentro del producto interior:

Hallar $\alpha_1 \ldots \alpha_N$ 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** maximizado y
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \le \alpha_i \le C$ para todo α_i

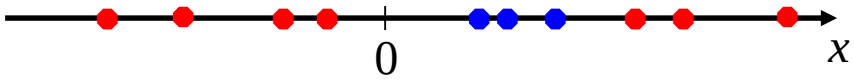
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x}_i^{\mathrm{T}}\mathbf{x} + b$$

SVM No lineal

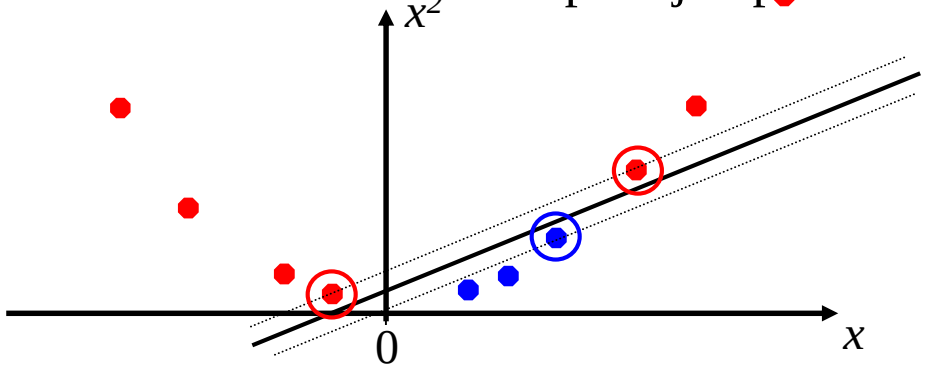
- Los datos que son linealmente separables no dan problemas:



- Pero que se hace si no lo son?

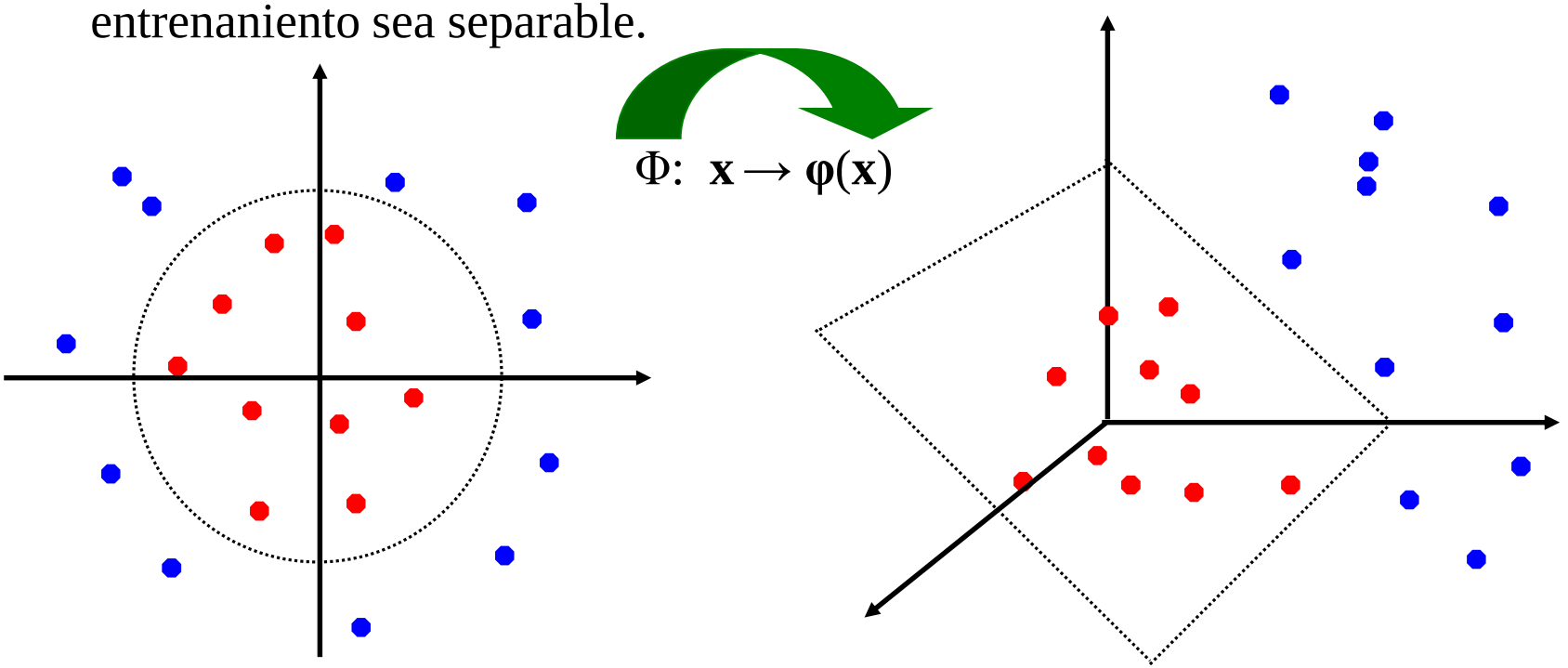


- La solucion es mapearlo en un conjunto de dimension mayor para que sea separable. Como cuadrar los datos por ejemplo



SVM NO lineal: Espacio de predictoras

- Idea General: El espacio original de variables predictoras siempre puede ser mapeado a un espacio de mayor dimension donde el conjunto de entrenaniento sea separable.



Uso de Kernels

- El clasificador lineal se basa en el producto interno entre vectores $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$
- Si todo datapoint es mapeado en un espacio de mayor dimension via alguna transformacion Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, entonces el product interno sera:

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

- Una funcion *kernel* es una funcion que es eqiuvalente a un product interno en algun espacio de predictoras.
- Ejemplo:

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

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

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

Que funciones son Kernels?

- Para algunas fucniones $K(\mathbf{x}_i,\mathbf{x}_j)$ cotejar que $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^{\mathbf{T}}\varphi(\mathbf{x}_j)$ puede ser complicado.
- Teorema de Mercer:

Every semi-positive definite symmetric function is a kernel

- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

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 de funciones Kernel

- Lineal: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ is $\mathbf{x}$ itself

- Polinomial de potencia p: $K(\mathbf{x}_i,\mathbf{x}_j)= (1+ \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j)^p$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ has $\binom{d+p}{p}$ dimensions

- Gaussiana (funciones bases radiales): $K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$
  - Mapping Φ: $\mathbf{x} \rightarrow \boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\mathbf{x})$ is *infinite-dimensional*: every point is mapped to *a function* (a Gaussian); combination of functions for support vectors is the separator.

- Higher-dimensional space still has *intrinsic* dimensionality *d* (the mapping is not *onto*), but linear separators in it correspond to *non-linear* separators in original space.

Formulacion Matematica del SVM NO lineal

- Formulacion del problema 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

- La solucion es:

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

- Nuevamente hay que usar tecnicas de ptimization para hallar las α_i's.

SVM aplicado a varias clases

Suponiendo que hay k clases
La estrategia: one versus el resto, contruye k clasificadores, considerando los elementos que no estan en la clase k como la clase negative.
En la estrategia, one versus one: se construyen $k(k-1)/2$ clasificadores de dos clases solamente. Luego se aplica voting
Hay una ultima estartegia: DAGSVM, parecida al oneversus one pero usando grafos dirigidos.

SVM en R

La libreria e1071 de R tiene una funcion svm que hace support vector machine

```
data(iris)
attach(iris)
 model <- svm(Species ~ ., data = iris)
# o alternativamente
x <- subset(iris, select = -Species)
y <- Species
model <- svm(x, y)
print(model)
summary(model)
```

SVM en R

```
# Prueba usando la muestra de entrenamiento
pred <- predict(model, x)
# o tambien
 pred <- fitted(model)
# Cotejando precision
table(pred, y)
# Calculando valores de decision y probabilidades:
pred <- predict(model, x, decision.values = TRUE)
attr(pred, "decision.values")[1:4,]
# visualizando (clases por color, SV por cruces):
plot(cmdscale(dist(iris[,-5])), col = as.integer(iris[,5]), pch = c("o","+")[1:150
%in% model$index + 1])
```