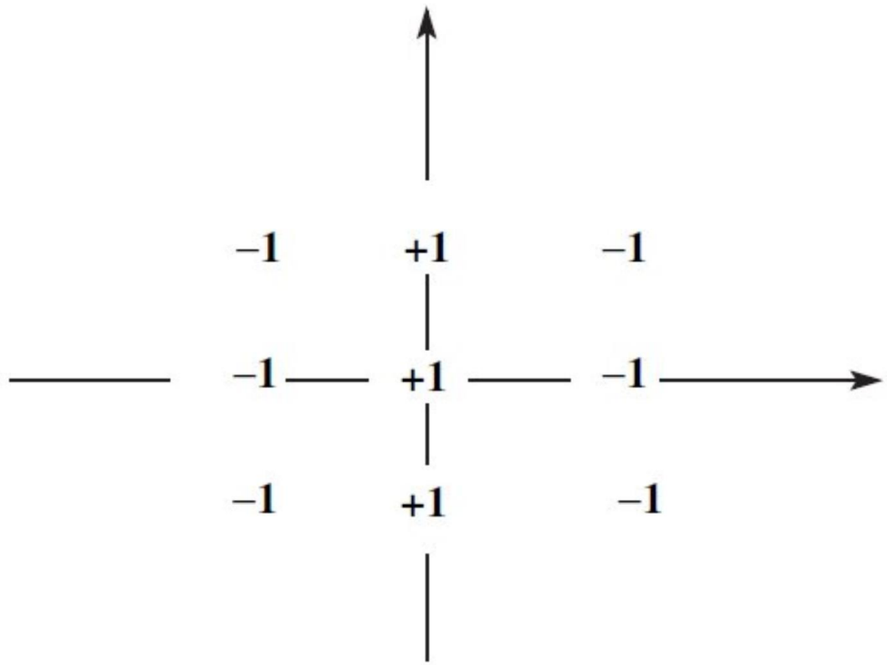
Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 - Inteligencia Artificial

Kernel-SVM

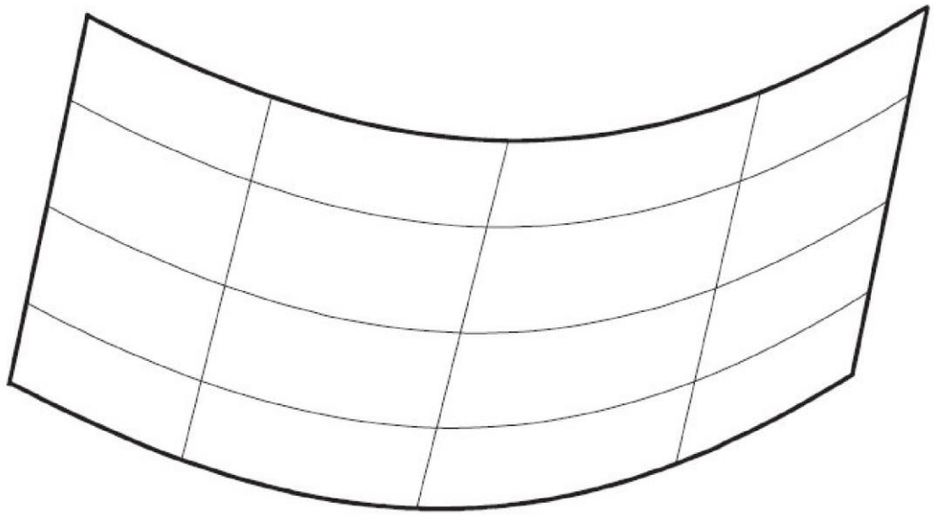
Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

Súper lindo el SVM, pero sigue siendo un clasificador lineal



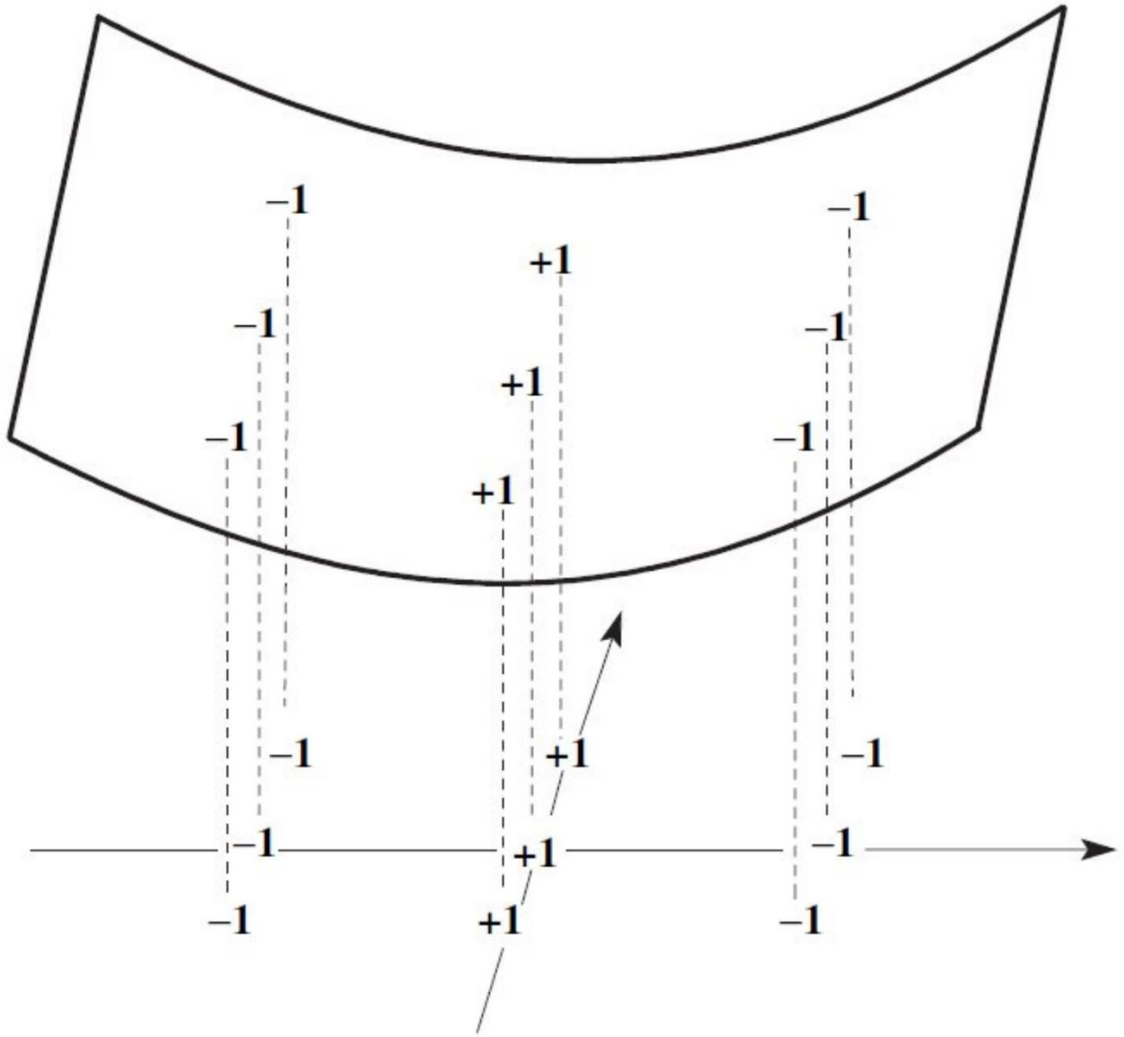
¿Es posible solucionar este problema con un SVM?

Una solución es generar un cambio de variables (transformación del espacio de características)

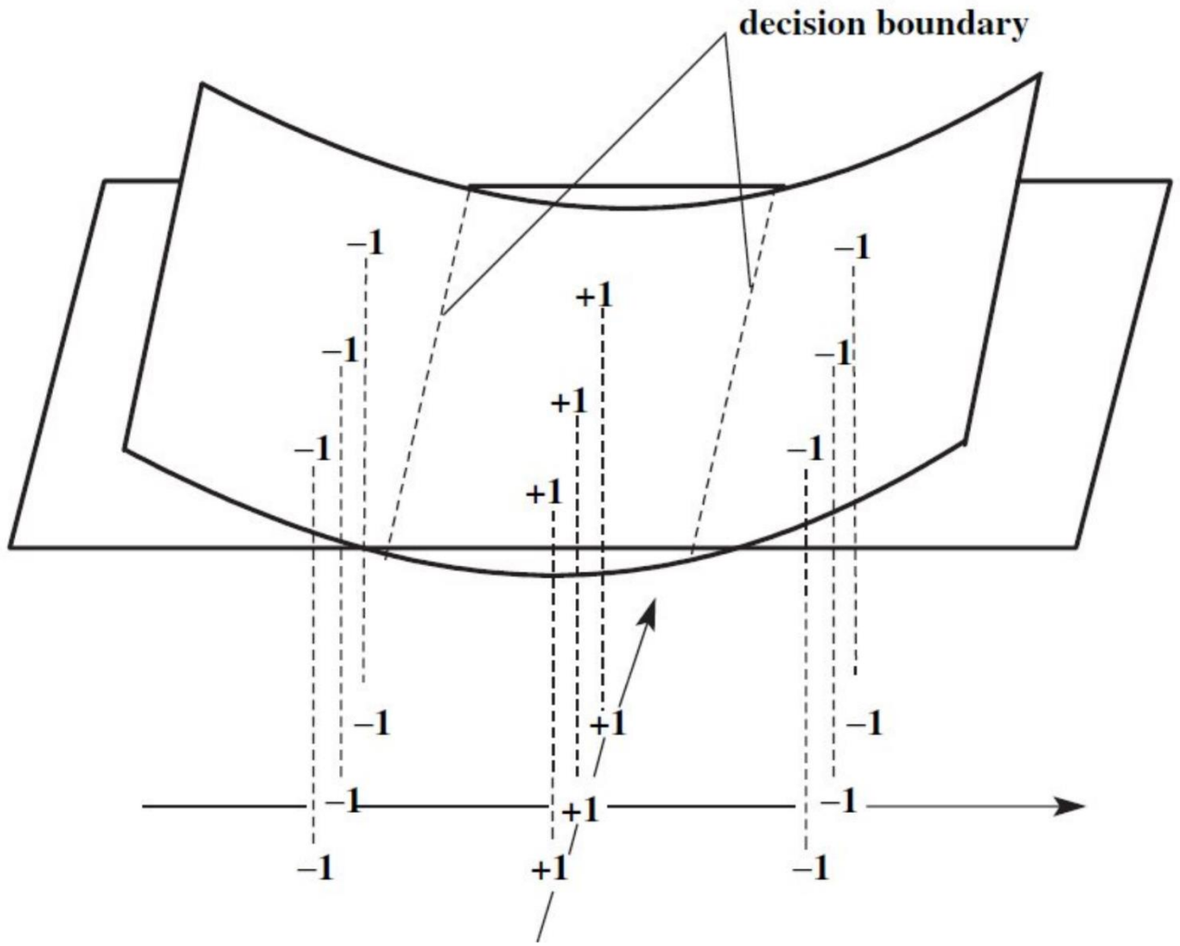


Espacio de características $f(x_1, x_2) = x_1^2$

Una solución es generar un cambio de variables (transformación de espacio de características)



No es muy distinto a regresión lineal con polinomios de mayor grado



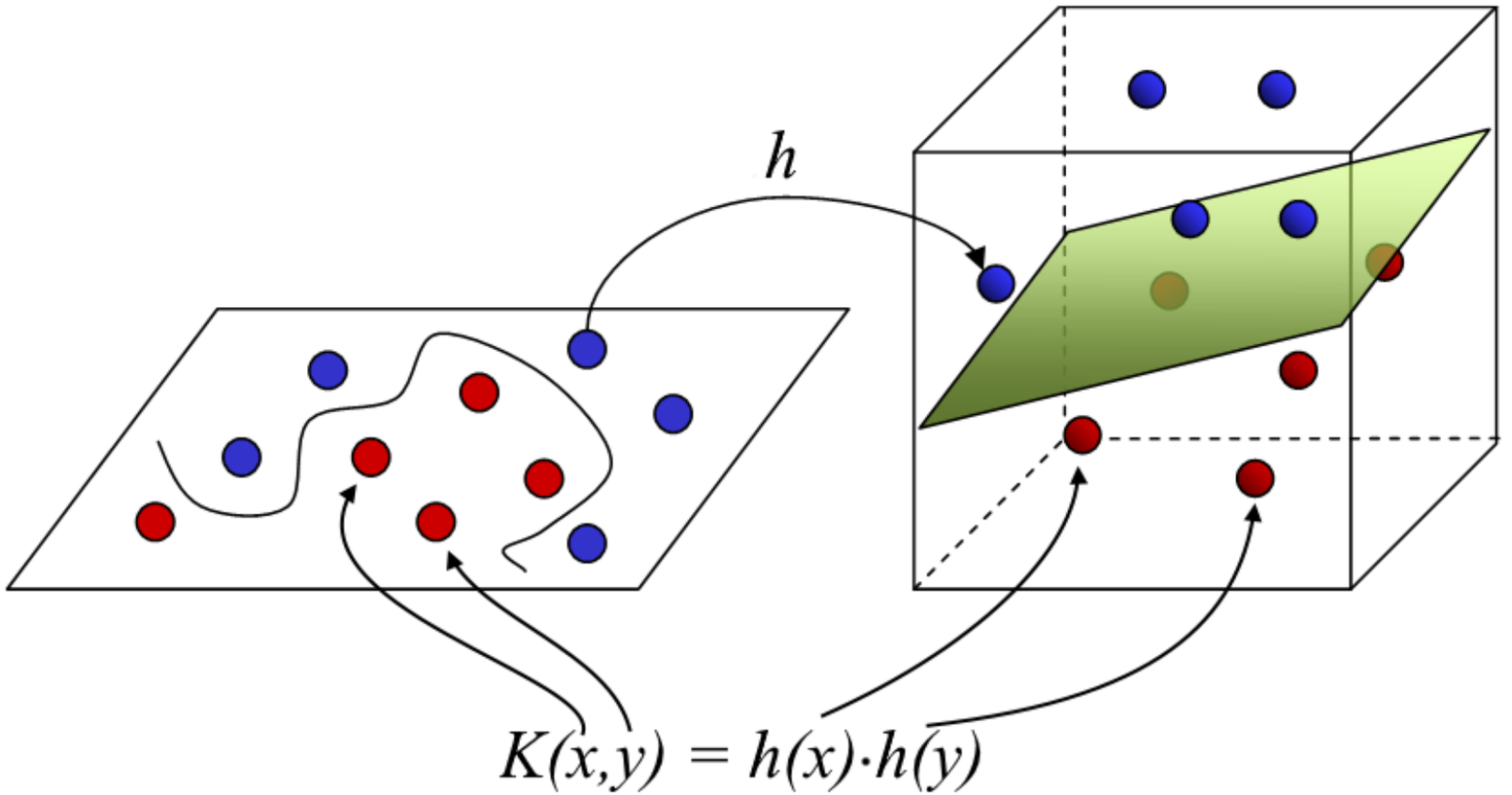
La pregunta es entonces, como podemos hacer más sistemático este mecanismo

- El ir probando grado a grado una expansión polinomial, claramente no escala en tiempo (ni de uno, ni del computador).
- Además, espacios de *features* de dimensionalidad muy grande son problemáticos de manejar (espacio + tiempo de ejecución).
- Peor aún, la distancia euclidiana entre vectores deja de tener sentido (todos los puntos están en promedio igual de cerca).
- ¿Estamos forzados a vivir en un espacio de superficies de decisión lineal?

Un pequeño interludio…

- Supongamos que tenemos una función $h(x)$, que lleva un vector x a un nuevo espacio de dimensionalidad arbitraria (potencialmente infinita).
- Imaginemos que, para cada par de vectores, $h(x_i)$ y $h(x_j)$, calculamos su producto punto y lo guardamos en la posición i, j de una matriz M.
- Acá viene la magia: si M es positiva definida ($z^\top Mz > 0, \forall z \neq \vec{0}$), entonces siempre existe una función $K(x_i, x_j)$, llamada *kernel*, que toma dos vectores, x_i y x_j, y retorna el producto punto de $h(x_i)$ y $h(x_j)$, sin necesidad de conocer la función $h(x)$.

Visualmente, la ventaja de los *kernels* toma más sentido



¿Y qué tiene que ver esto con los SVM?

Recordemos brevemente la resolución del problema de optimización asociado a un SVM

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1,\ i = 1, \ldots, N$$

Lagrangiano

$$L_P = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - 1]$$

Imponiendo las condiciones de KKT, más algo de álgebra

$$\max_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} \alpha_i\alpha_k y_i y_k x_i^T x_k$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } \alpha_i[y_i(x_i^T\beta + \beta_0) - 1] = 0\ \forall i$$

Podemos incorporar esto en los SVM, mediante algo conocido como el *kernel trick*

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \hat{x}_i^T \hat{x}_{i'}$$

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \langle \hat{x}_i, \hat{x}_{i'} \rangle$$

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle$$

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} K(x_i, x_{i'})$$

Podemos incorporar esto en los SVM, mediante algo conocido como el *kernel trick*

$$f(x) = h(x)^T\beta + \beta_0$$

$$\beta = \sum_{i=1}^{N} \alpha_i y_i h(x_i)$$

Podemos incorporar esto en los SVM, mediante algo conocido como el *kernel trick*

$$
\begin{aligned}
f(x) &= h(x)^T\beta + \beta_0 \\
&= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i)\rangle + \beta_0
\end{aligned}
\qquad\qquad
\beta = \sum_{i=1}^{N} \alpha_i y_i h(x_i)
$$

Podemos incorporar esto en los SVM, mediante algo conocido como el *kernel trick*

$$
\begin{aligned}
f(x) &= h(x)^T\beta + \beta_0 \\
&= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i)\rangle + \beta_0 \\
&= \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \beta_0
\end{aligned}
\qquad
\beta = \sum_{i=1}^{N} \alpha_i y_i h(x_i)
$$

Podemos incorporar esto en los SVM, mediante algo conocido como el *kernel trick*

$$
\begin{aligned}
f(x) &= h(x)^T\beta + \beta_0 \\
&= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0 \\
&= \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \beta_0
\end{aligned}
\qquad
\beta = \sum_{i=1}^{N} \alpha_i y_i h(x_i)
$$

- Como siempre operamos sobre el producto punto entre vectores $h(x)$, y no sobre los $h(x)$ de manera individual, también podemos sustituirlos por la aplicación del mismo *kernel* $K(x, y)$.
- Esto nos permite facilitar la búsqueda de la solución, ya que mientras más dimensiones hay, más posibilidades existen de encontrar un hiperplano óptimo (¿por qué?, ¿qué pasa con el overfitting?).

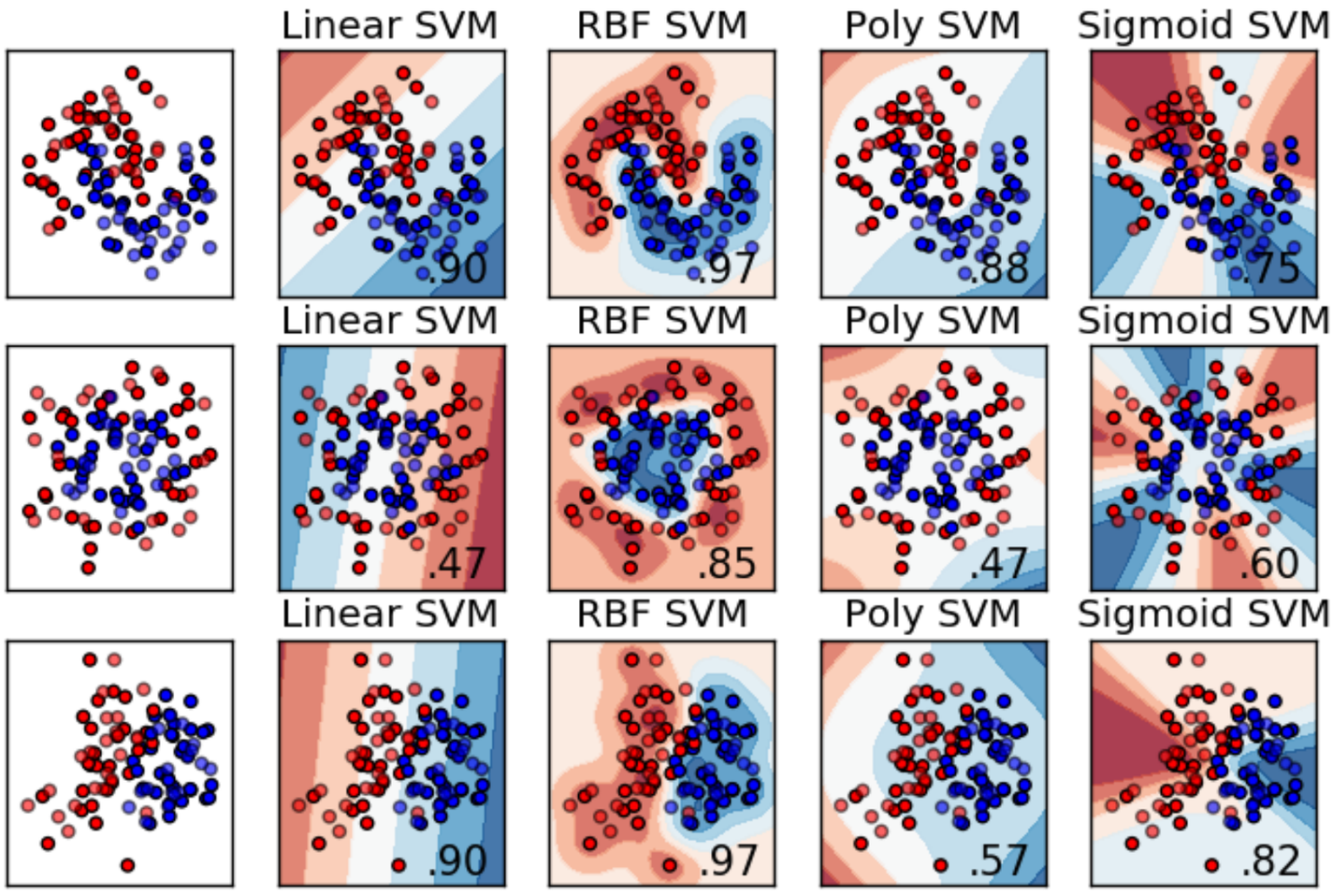
Intuitivamente, ¿qué mide un *kernel*?

Intuitivamente, un *kernel* mide la similitud entre dos vectores (pero en un espacio distinto)

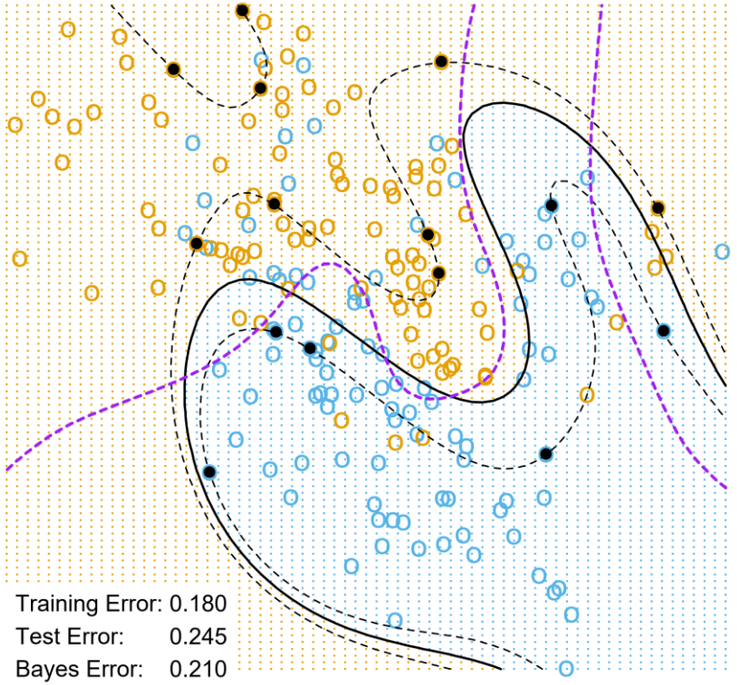
$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \beta_0$$

$$\begin{aligned} d\text{th-Degree polynomial: } & K(x, x') = (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis: } & K(x, x') = \exp(-\gamma \|x - x'\|^2), \\ \text{Neural network: } & K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2). \end{aligned}$$

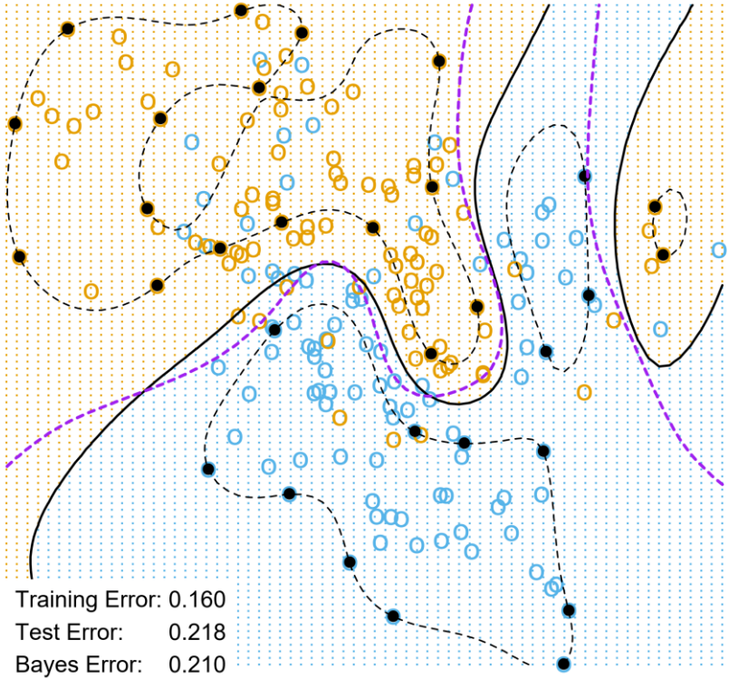
¿Cuál es el *kernel* de los SVM que vimos inicialmente?


Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.90
.97
.88
.75
Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.47
.85
.47
.60
Linear SVM
RBF SVM
Poly SVM
Sigmoid SVM
.90
.97
.57
.82

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space

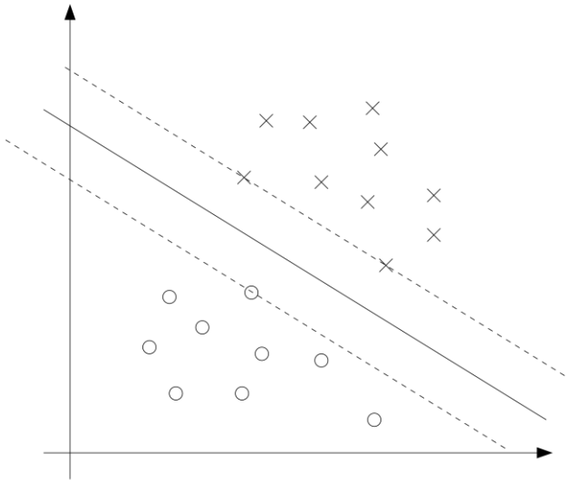


Un par de preguntas para terminar Kernel-SVMs

- ¿Cuáles son los vectores de soporte en este caso?
- ¿Cómo se ve el margen en este caso?

SVMs continúan siendo relevantes en *machine learning*

- SVMs son de los algoritmos *off-the-shelf* con mejor rendimiento.
- Simpleza, concepto de margen e incorporación de *kernels* son sus grandes fortalezas.
- Han perdido importancia durante la última década con respecto a las técnicas modernas de Deep Learning y de ensambles.



Lecturas optativas de profundización

- "*The Elements of Statistical Learning*", de Hastie, Tibshirani y Friedman, sección 4.5.2, 12.2 y 12.3: https://web.stanford.edu/~hastie/ElemStatLearn/
- "*Ensemble of Exemplar-SVMs for Object Detection and Beyond*", de T. Malisiewicz et al.: https://www.cs.cmu.edu/~tmalisie/projects/iccv11/exemplarsvm-iccv11.pdf
- "SVM Parameter Tuning": https://towardsdatascience.com/a-guide-to-svm-parameter-tuning-8bfe6b8a452c

Vamos a Colab...

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Fundamentos de Machine Learning

Kernel-SVM

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación