Métodos basados en árboles y SVM

Jocelyn Dunstan Escudero

jdunstan@uc.cl

Departamento de Ciencia de la Computación
& Instituto de Matemática Computacional
Pontificia Universidad Católica de Chile

Santiago, Chile

15 de octubre de 2025

Objetivos

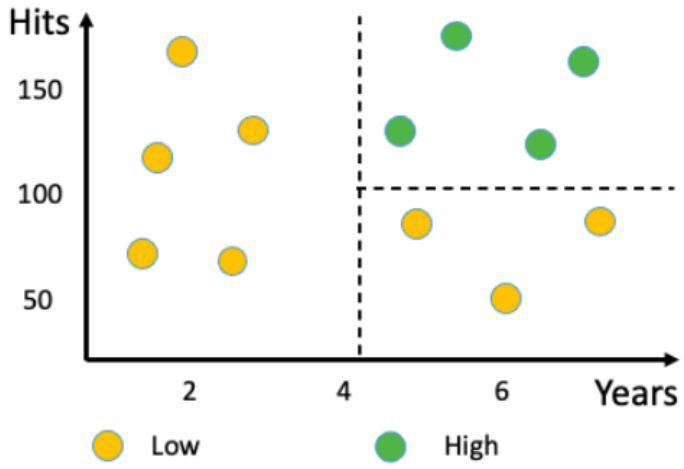
- Describir las características fundamentales de los árboles de decisión y cómo segmentan el espacio de parámetros.
- Explicar los métodos para la construcción, evaluación y poda de árboles de decisión.
- Comprender el concepto de *maximal margin classifier* y su fundamento teórico.
- Analizar las dificultades y limitaciones del *maximal margin classifier* y describir el uso de kernels para abordar dichas dificultades.

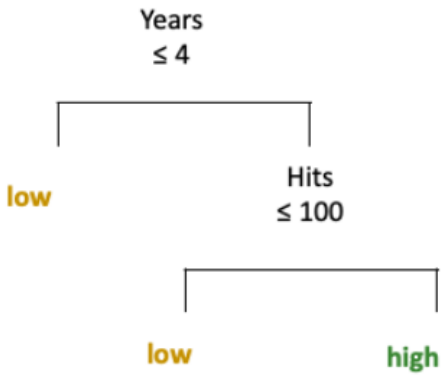
Idea general

Características

- Funcionan bien con datos pequeños
- No es necesario normalizar los datos antes, pero sí si se van a comparar con otros métodos
- Son fáciles de explicar
- Rendimiento competitivo, especialmente cuando se promedian muchos.
- Permiten la inferencia y la reducción de la dimensionalidad

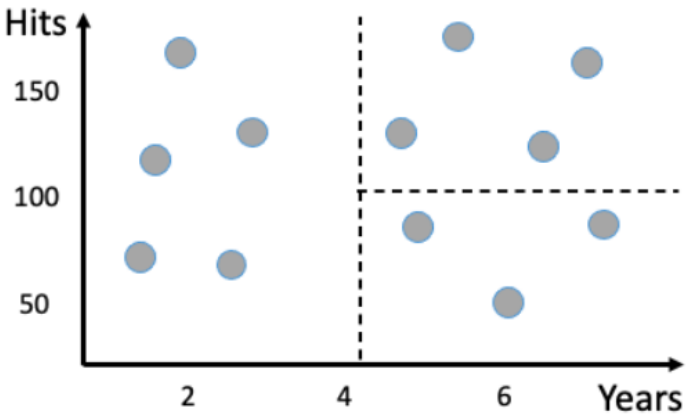
Idea general: segmentación del espacio de parámetros

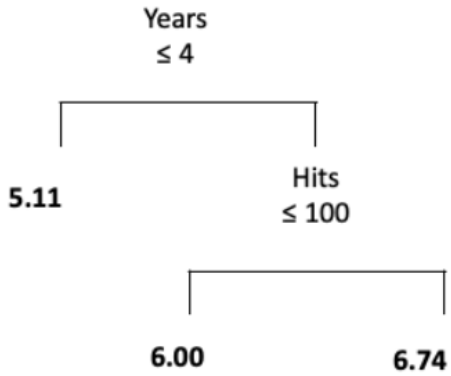




Adapted from Introduction to Statistical Learning by James, Witten, Hastie & Tibshirani

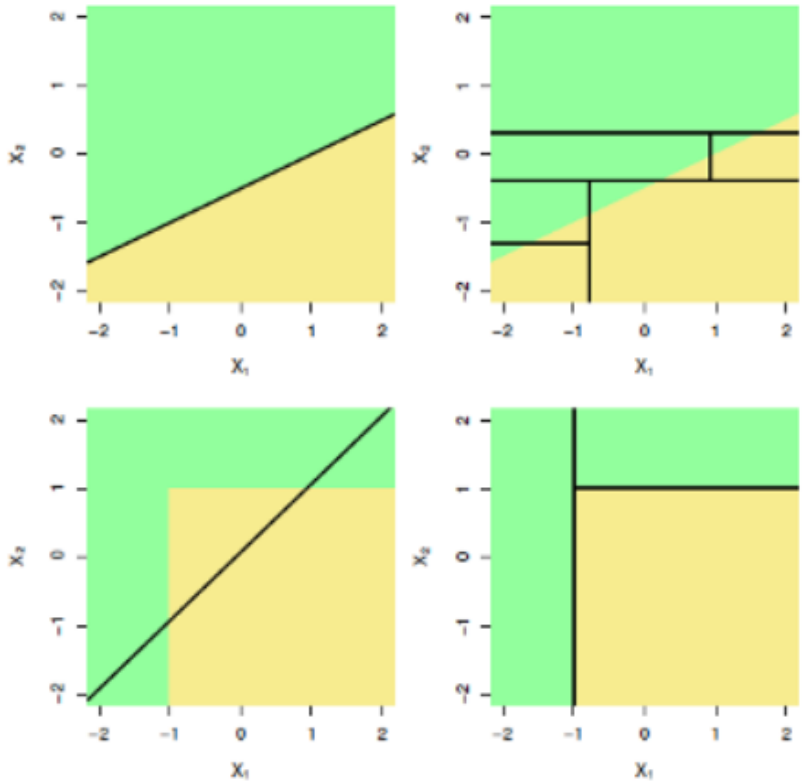
Y la misma idea es válida para los árboles de regresión





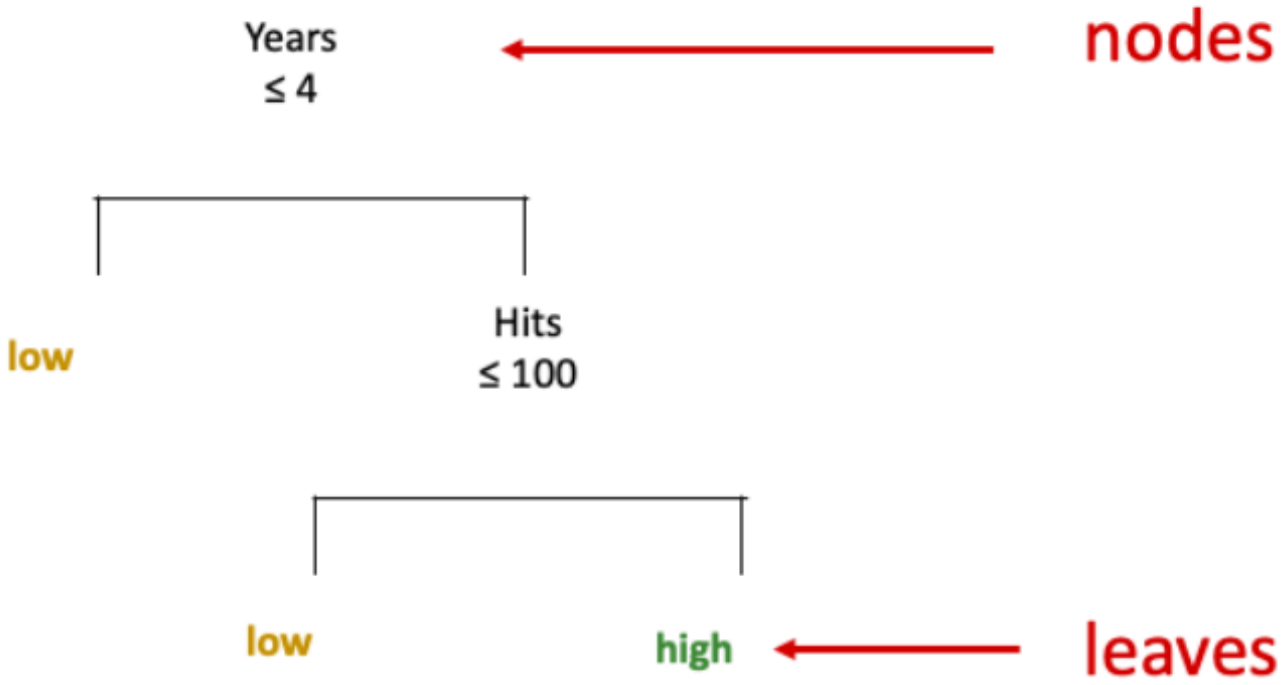
Adapted from Introduction to Statistical Learning by James, Witten, Hastie & Tibshirani

Comparación entre árboles y modelos lineales



El problema es que general resolveremos problemas en alta dimensión y no tenemos intuición de cómo se ve el la clasificación en el espacio de características

Estructura de un árbol

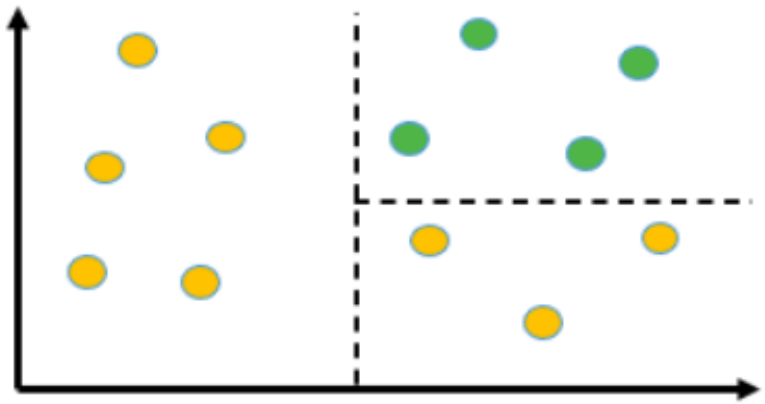


Es un árbol de cabeza!

Criterios para generar ramificación y podar

Construcción de un árbol

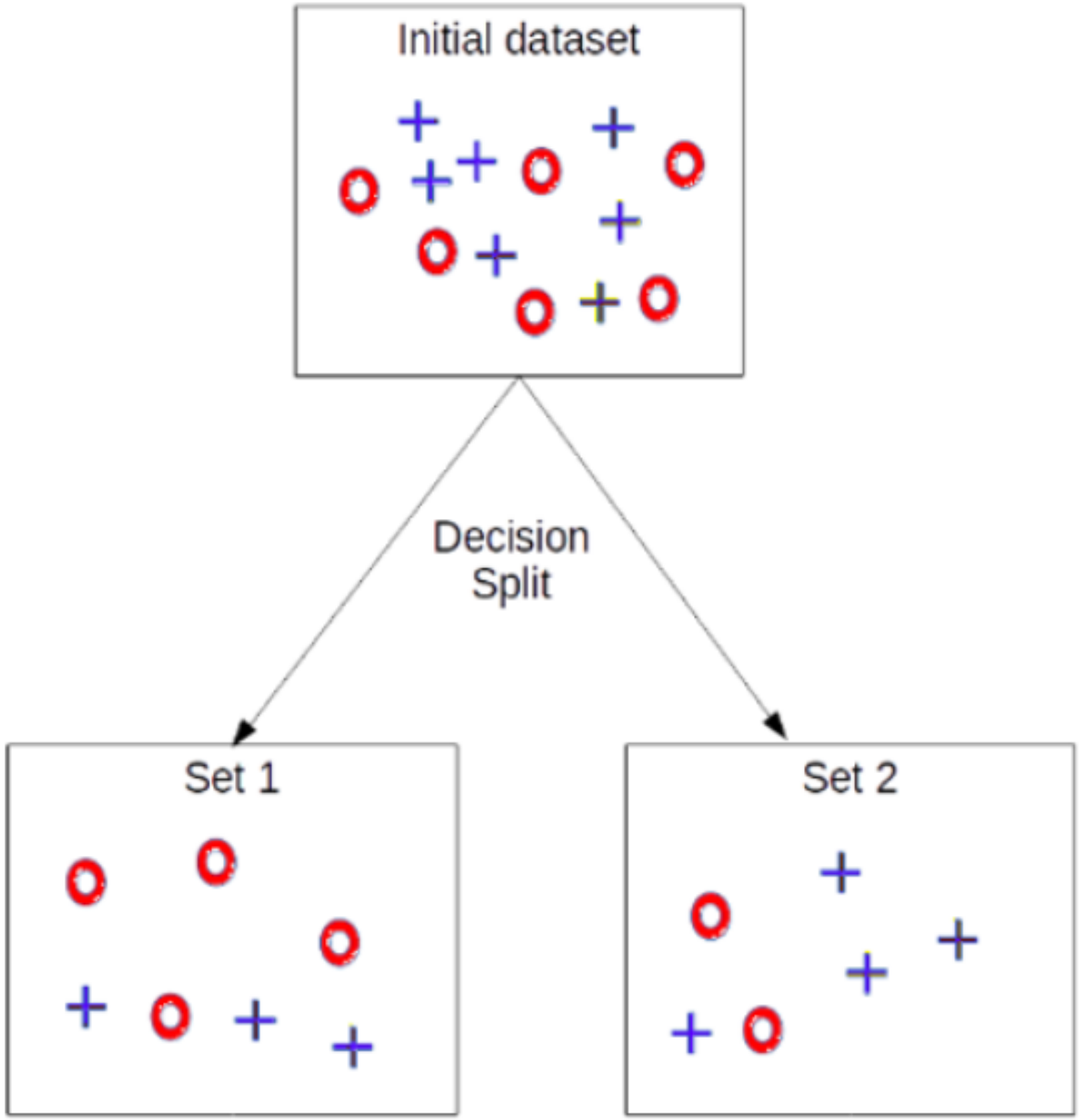
- En general, el problema de crear N cajas con diferentes tamaños a partir de los datos es inviable.
- Los árboles actúan localmente: para un predictor dado, encuentra el punto de división



Pasos para construir árboles de clasificación

Formas de detener la división:

- Fijar una profundidad máxima
- Cierta función menor a un valor (Gini o p-value)
- Número de muestras en cada nodo terminal


Initial dataset
Decision
Split
Set 1
Set 2

Gini index

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Donde p_{mk} es la proporción de observaciones de entrenamiento en la m-ésima región de la k-clase

Entropía de Shannon

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

Donde p_{mk} es la proporción de observaciones de entrenamiento en la m-ésima región de la k-clase

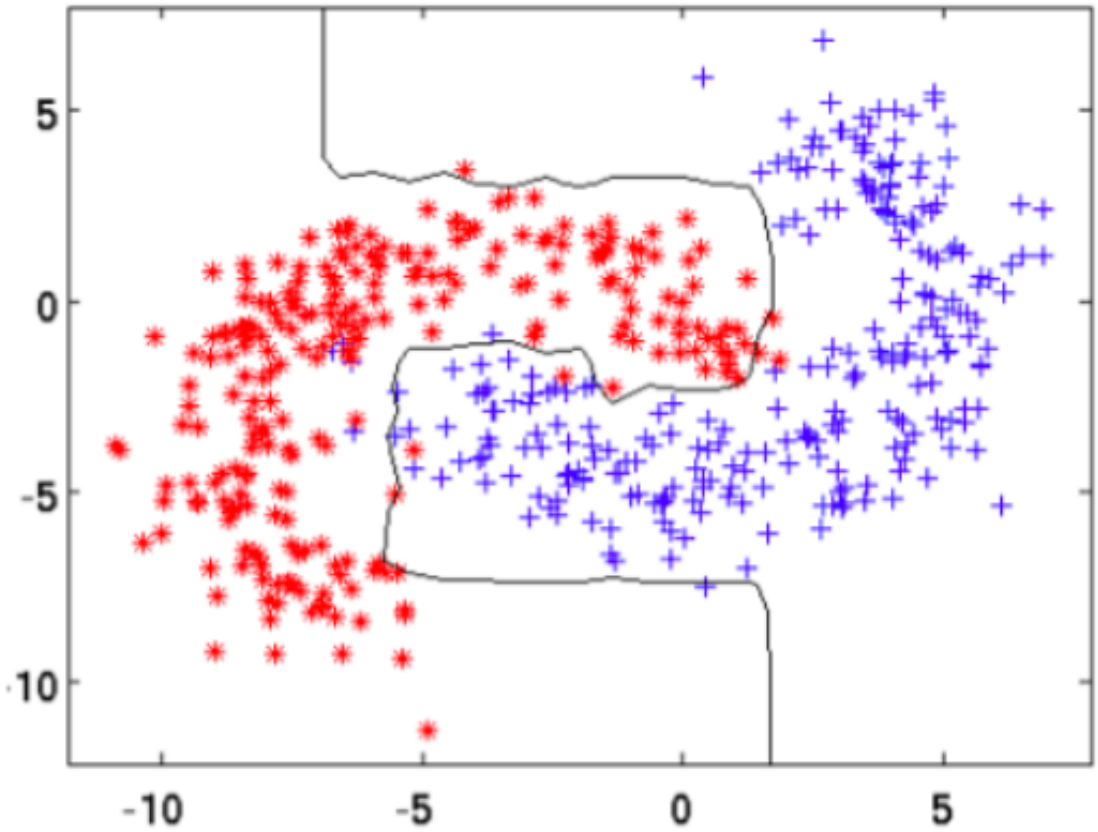
Esta métrica mide la ganancia de información

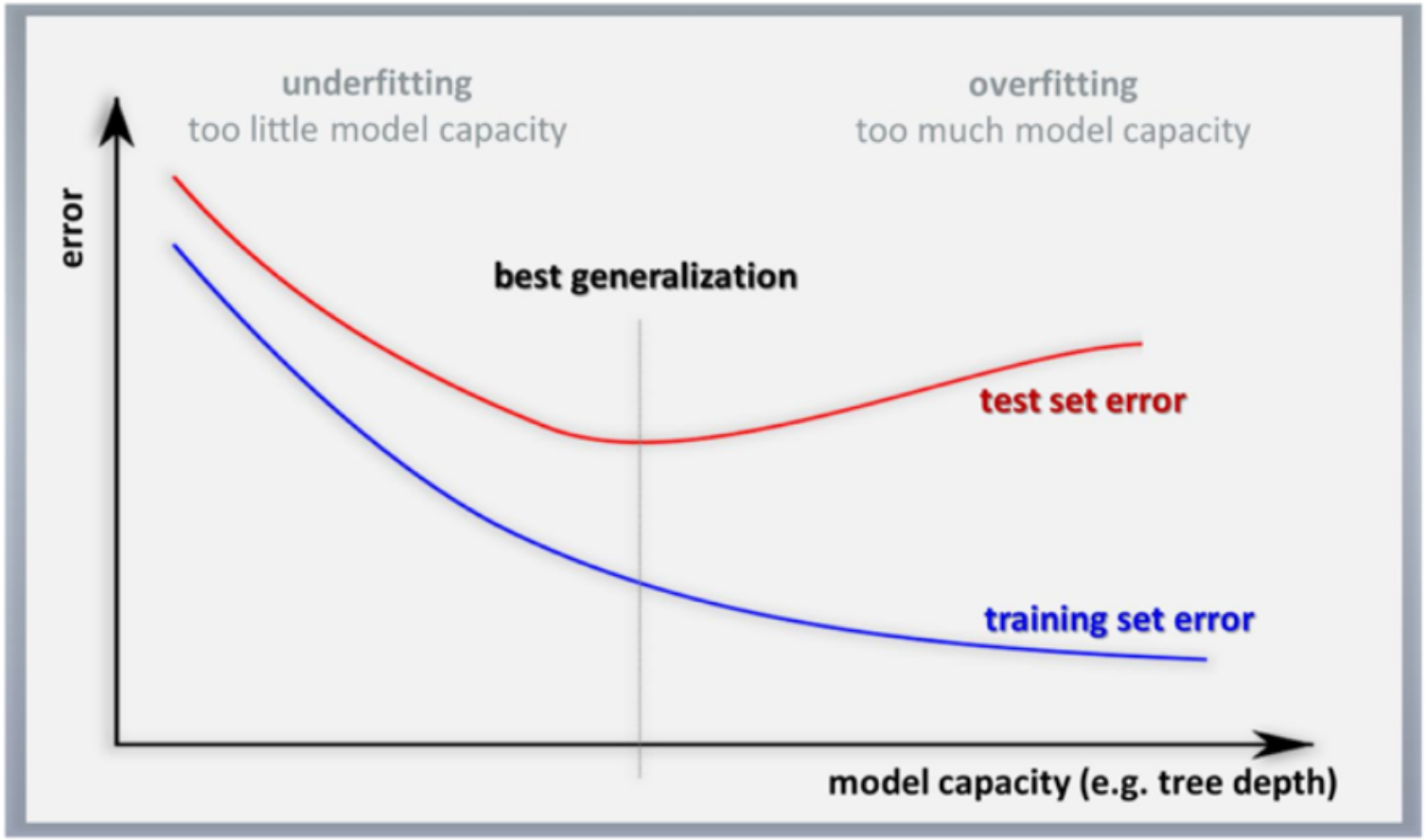
	Class 1	Class 2	Entropy$(i\|j, t_j)$
R_1	0	6	$-(\frac{6}{6}\log_2\frac{6}{6}+\frac{0}{6}\log_2\frac{0}{6})=0$
R_2	5	8	$-(\frac{5}{13}\log_2\frac{5}{13}+\frac{8}{13}\log_2\frac{8}{13})\approx 1.38$

https://harvard-iacs.github.io/2018-CS109A/

Podar un árbol

Los árboles con demasiadas ramas tienden a sobreajustar los datos.




underfitting
too little model capacity
overfitting
too much model capacity
error
best generalization
test set error
training set error
model capacity (e.g. tree depth)

Random Forest

Random Forest

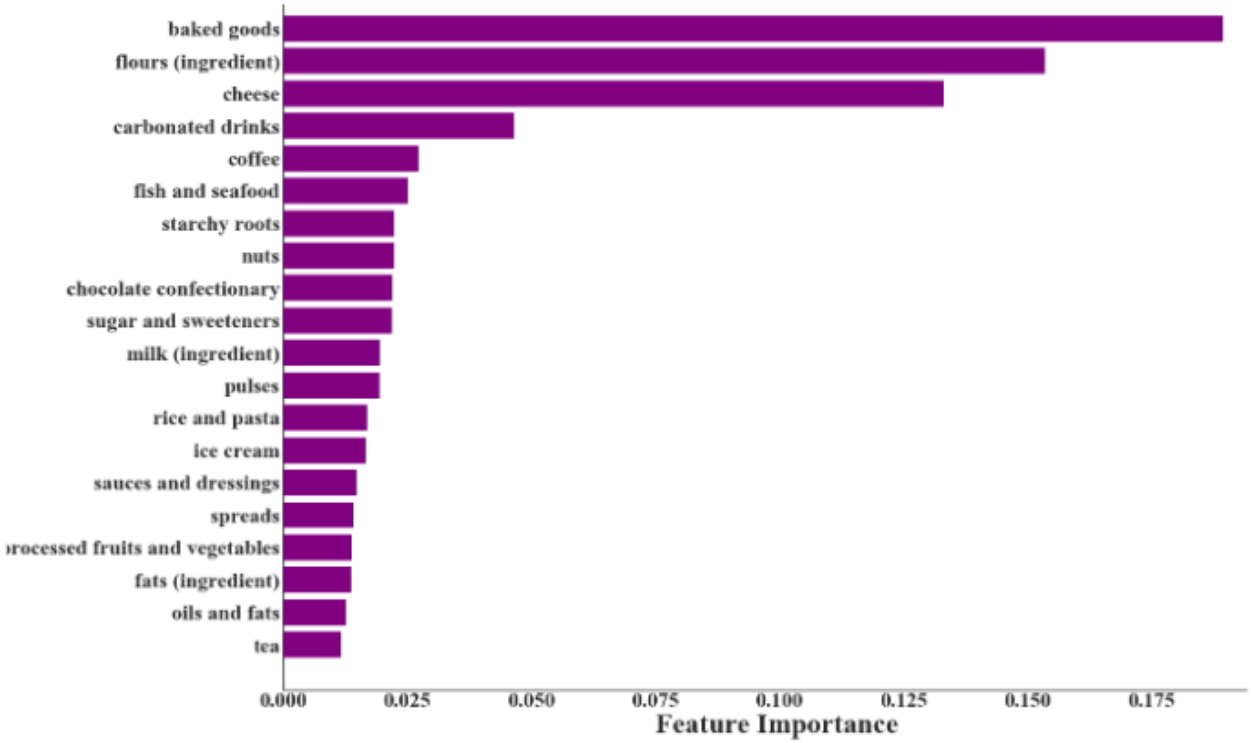
- Decorrelaciona los árboles eligiendo una selección aleatoria de m predictores cada vez ($m < n$)
- Parámetros a ajustar:
  - Número de árboles
  - m predictores
  - ¿Cuándo parar? Valor P, entropía, profundidad
- Lista de importancia de las variables

Random Forest

- Hay parámetros por defecto, pero en general deben ser ajustados para los datos de entrenamiento específicos.
- Por ejemplo, la recomendación de Breiman para m es sqrt(n) para la clasificación y n/3 para la regresión, pero es un parámetro que debe explorarse.

Variable importance list

Es una medida de la disminución de la precisión, promediada en todos los árboles, cuando se deja de lado un predictor en el modelo



Support Vector Machines

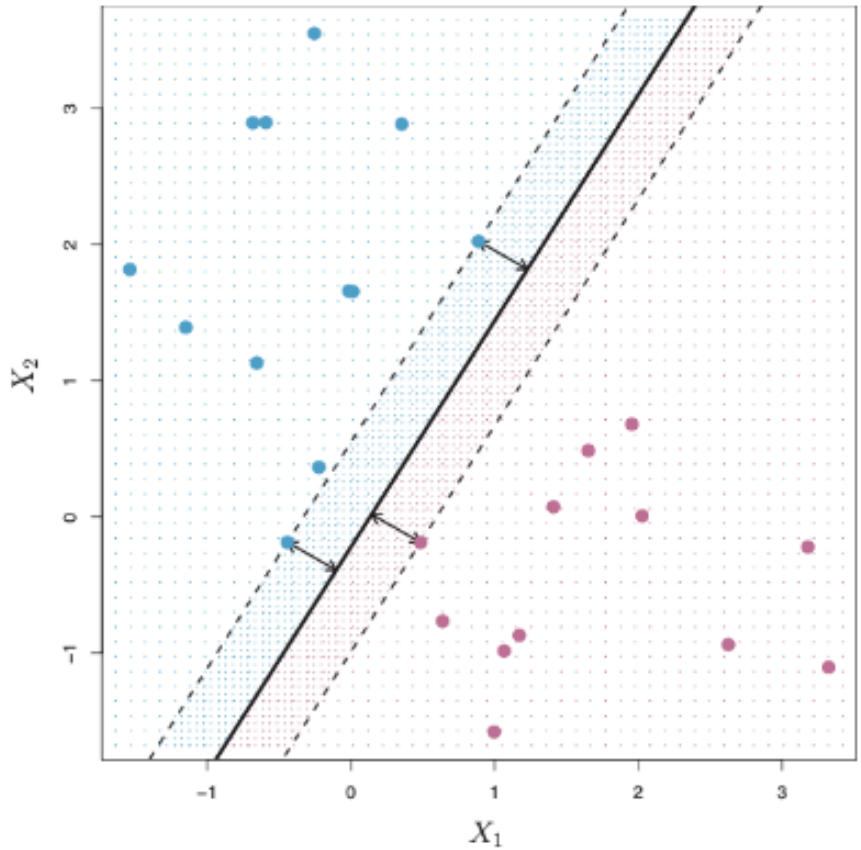
Idea general

- Por simplicidad pensemos en una clasificación binaria, y consideremos que podemos separar ambas clases por un hiperplano.
- En un espacio de p-dimensiones, un hiperplano es un sub-espacio afín de dimensión p-1 (afín porque no necesita pasar por el origen).
- Diremos que todo lo que está sobre el hiperplano es una clase, y lo de debajo es la otra.
- ¿Cómo escogemos entre los muchos hiperplanos posibles?

https://web.stanford.edu/ hastie/ISLRv2_website.pdf

The maximal margin classifier

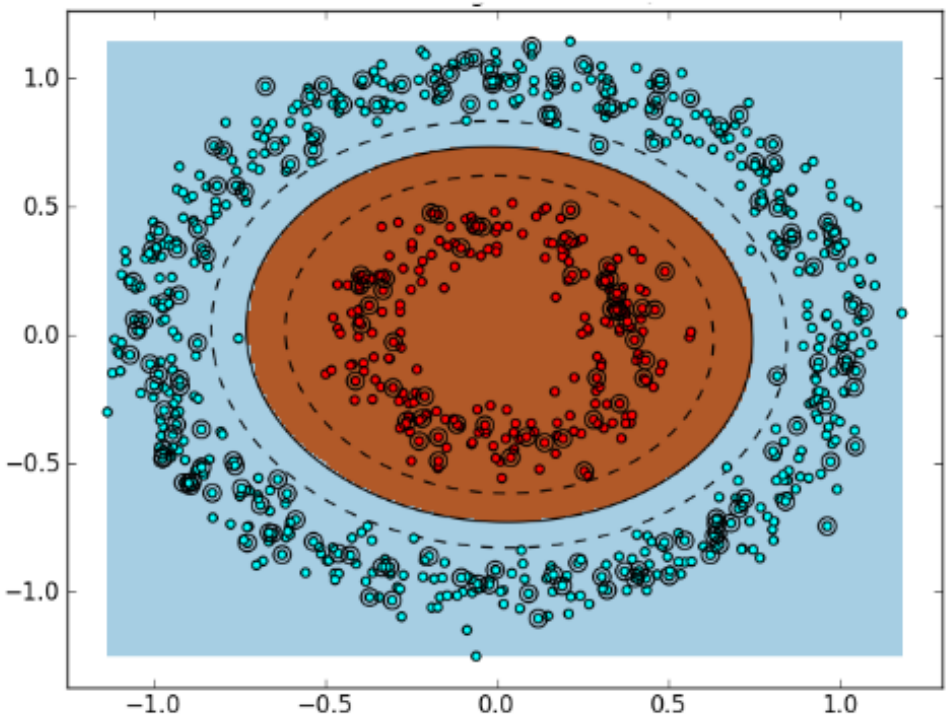
Diremos que el mejor clasificador es aquel que maximiza las distancias entre el hiperplano y los puntos más cercanos al borde.



https://web.stanford.edu/ hastie/ISLRv2_website.pdf

Dificultades

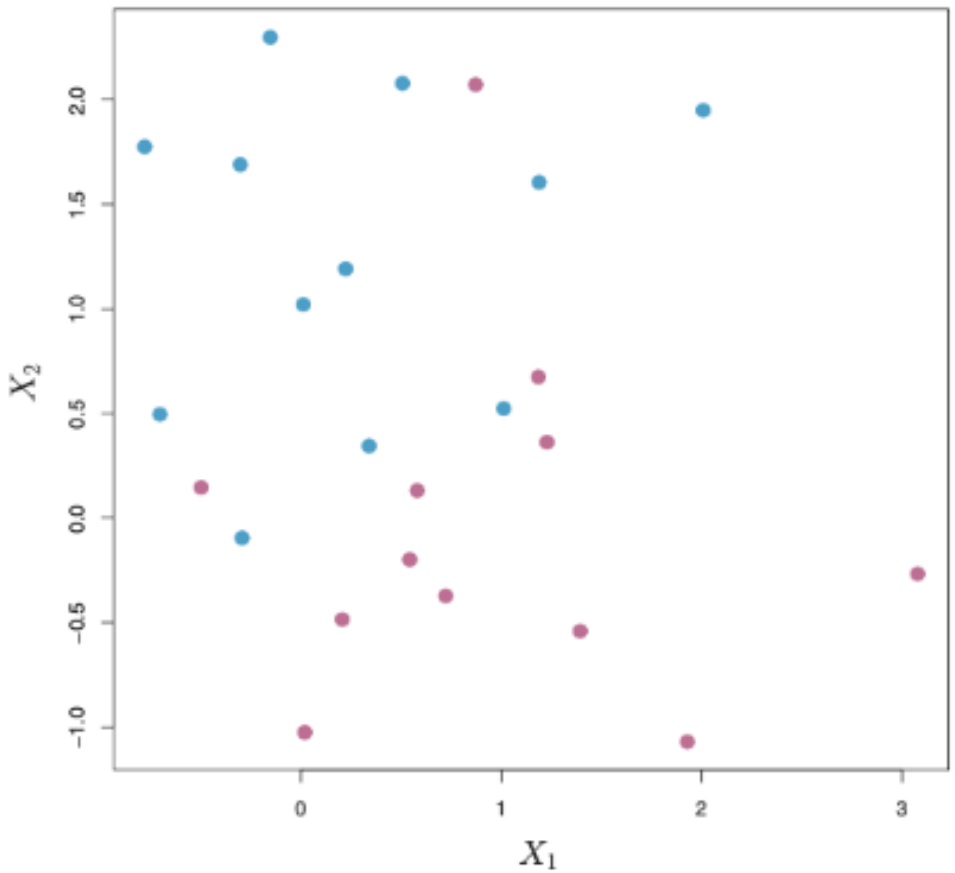
- ¿Cómo extendemos esta idea a múltiples clases o al problema de regresión?
- ¿Qué pasa con las clases no linealmente separables pero si en otra geometría?



https://web.stanford.edu/ hastie/ISLRv2_website.pdf

Dificultades

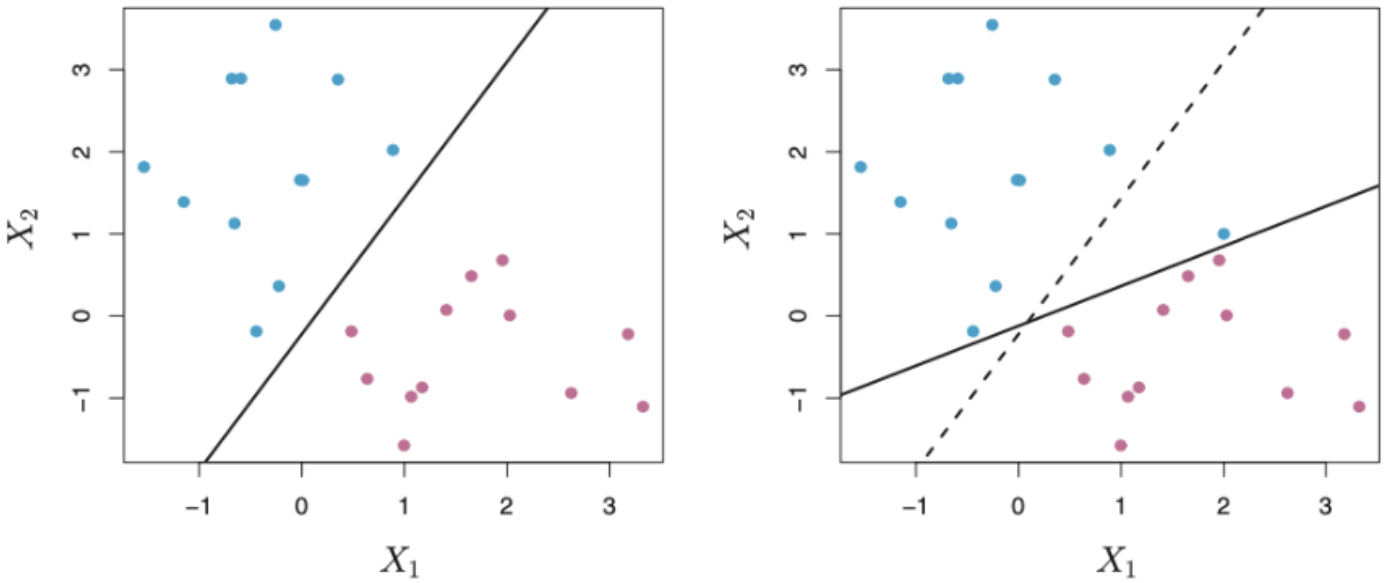
- ¿Qué pasa si no logro dejar todos los puntos de una clase a un lado de la recta?



https://web.stanford.edu/ hastie/ISLRv2_website.pdf

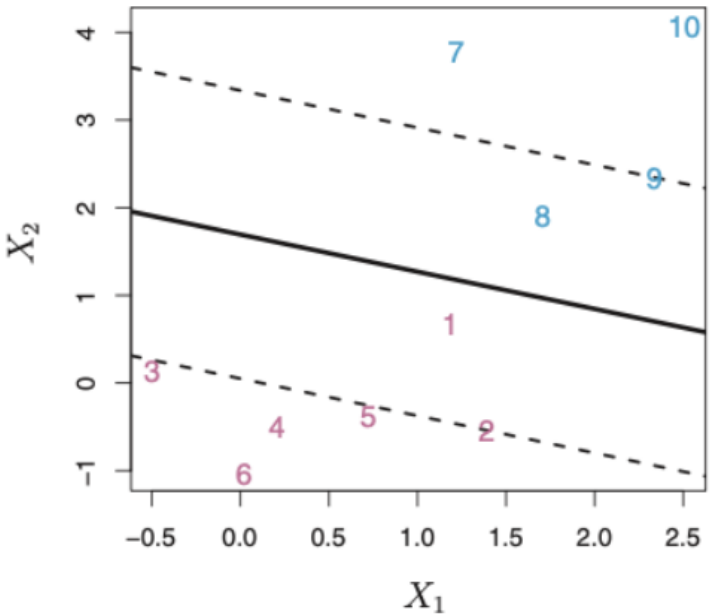
Dificultades

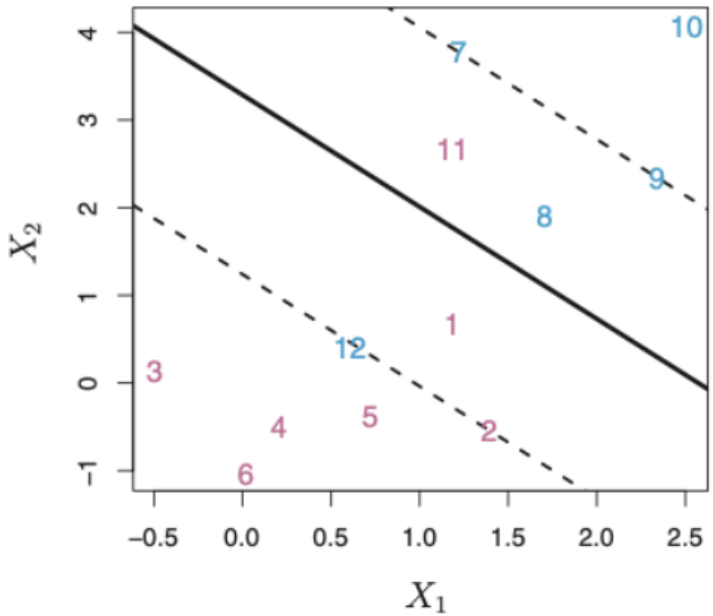
- ¿Quiero realmente que los puntos cerca del hiperplano tenga tanta influencia?



https://web.stanford.edu/ hastie/ISLRv2_website.pdf

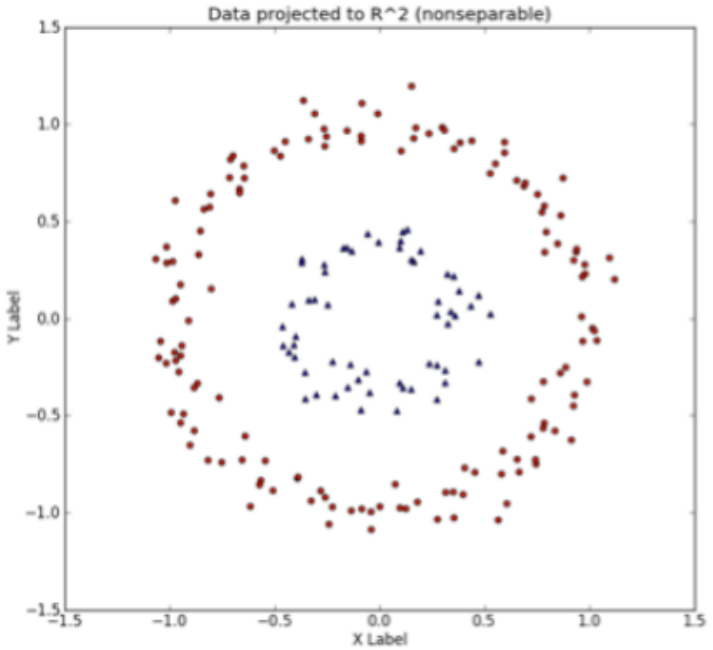
Soft margin classifier


10
7
9
8
1
3
4
5
2
6
4
3
2
1
0
−1
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
X_1
X_2


10
7
11
9
8
1
12
3
4
5
2
6
4
3
2
1
0
−1
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
X_1
X_2

Elección del kernel

Idea: Separable in higher-dimension



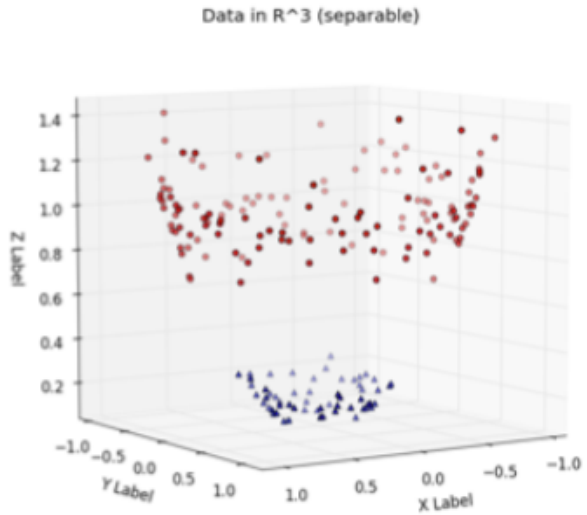


Figure 5: (Left) A dataset in $\mathbb{R}^2$, not linearly separable. (Right) The same dataset transformed by the transformation: $[x_1, x_2] = [x_1, x_2, {x_1}^2 + {x_2}^2]$.

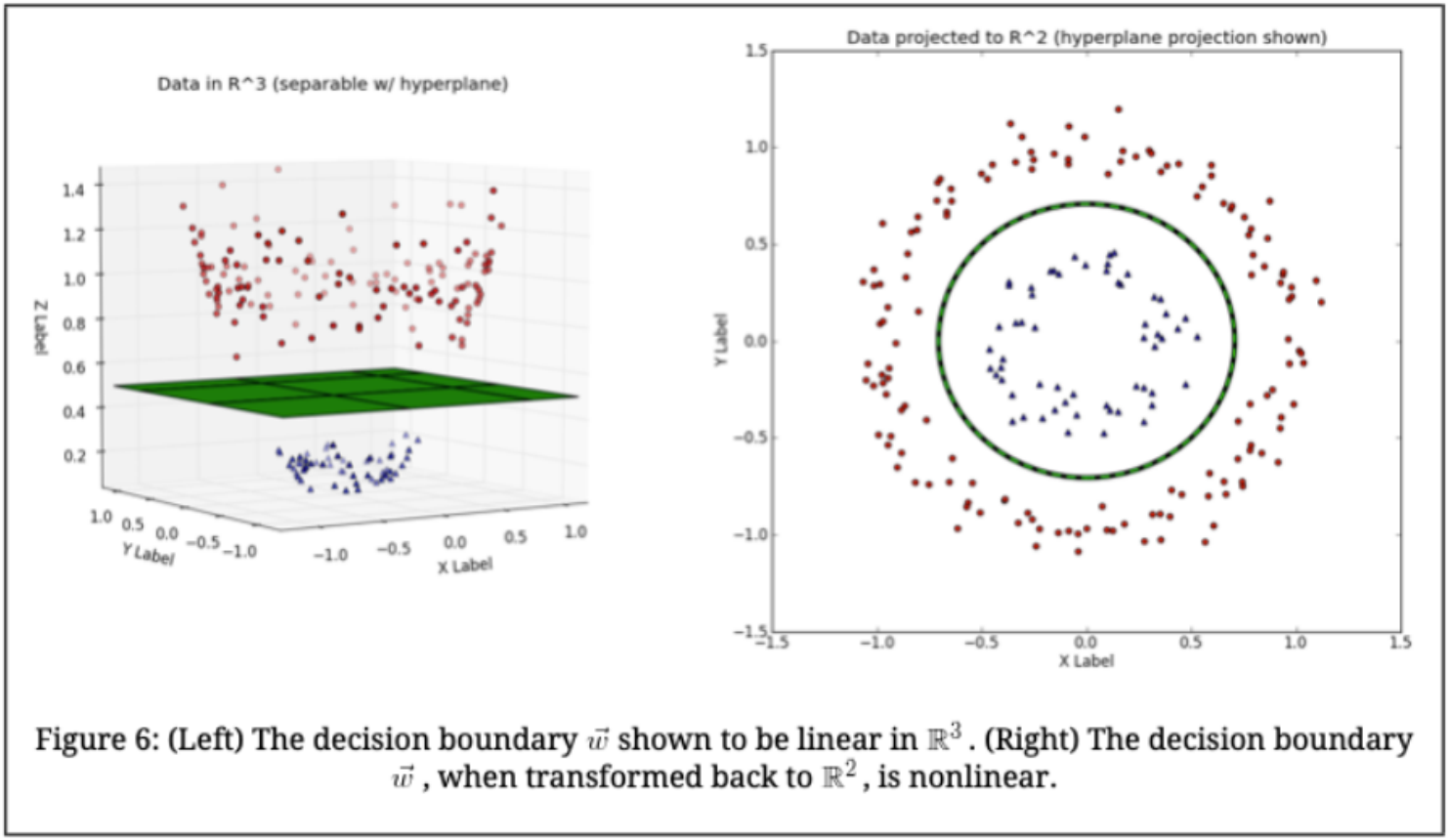


Figure 6: (Left) The decision boundary $\vec{w}$ shown to be linear in $\mathbb{R}^3$. (Right) The decision boundary $\vec{w}$, when transformed back to $\mathbb{R}^2$, is nonlinear.

http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Intuición detrás de usar un kernel particular

- Si mis datos **v**,**w** están en R^N , la función kernel K(**v**,**w**) produce un número en R. Esta función tiene propiedades matemáticas especiales, pero por ahora pensemos que un ejemplo de kernel es el producto punto.
- En el ejemplo anterior vimos que pasándonos a R^M con $M > N$ podemos encontrar un hiperplano que separa linealmente las clases. Pero esto es costoso computacionalmente.
- Lo que hacemos al tomar un kernel de un cierto tipo es aumentar las dimensiones del problema pero sin tener que llevar todos los datos a un nuevo espacio, sino que cambiando la forma en que calculamos distancias. Considere por ejemplo el siguiente kernel polinomial con el que pasamos de 2 a 5 dimensiones:

$$[x_1, x_2] = [x_1^2, x_2^2, \sqrt{2} \cdot x_1 \cdot x_2, \sqrt{2 \cdot c} \cdot x_1, \sqrt{2 \cdot c} \cdot x_2, c]$$

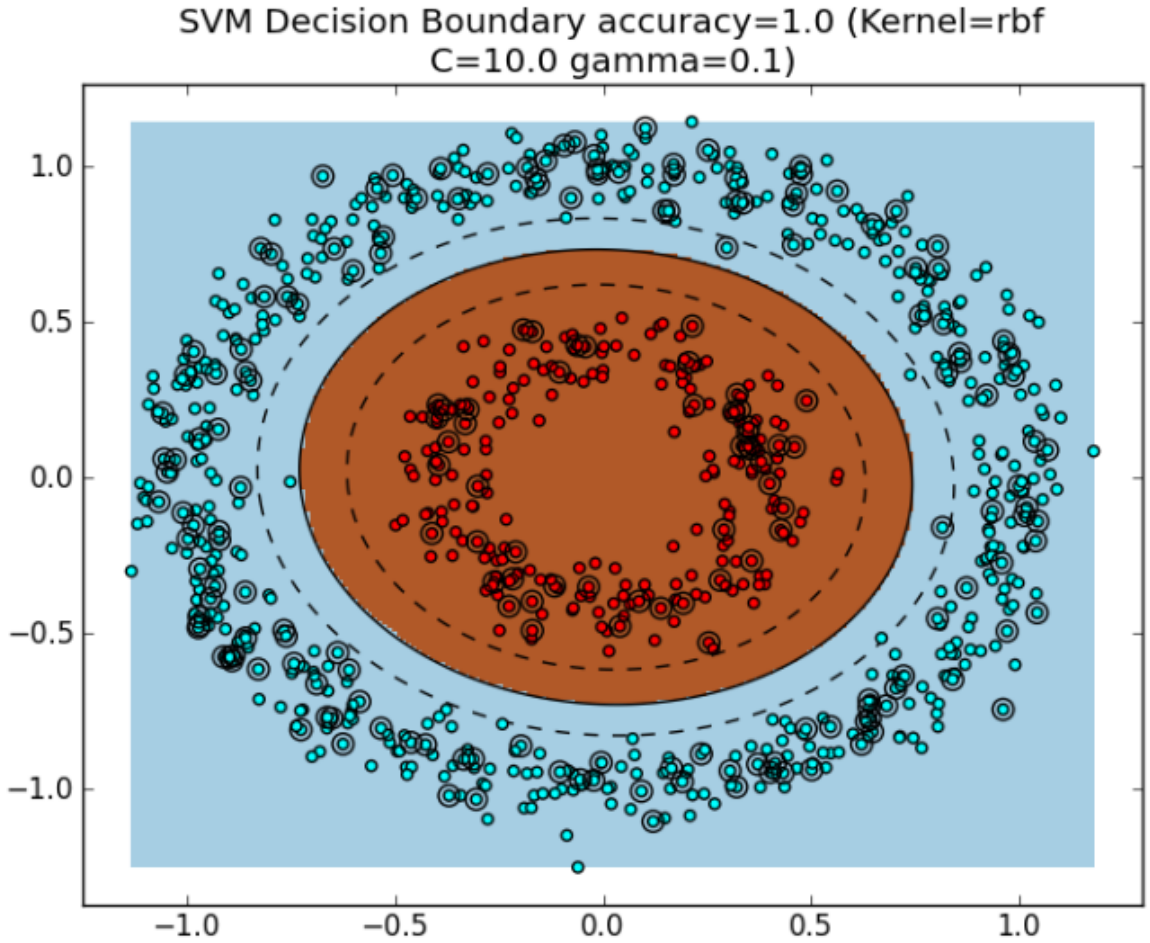
http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Kernel

- El kernel (o función del núcleo) define el producto interno en el espacio transformado.

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^p; \text{ polynomial kernel.}$$
$$K(x_i, x_j) = e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2}; \text{ Gaussian kernel; Special case of Radial Basis Function.}$$
$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}; \text{ RBF Kernel}$$
$$K(x_i, x_j) = \tanh(\eta\, x_i \cdot x_j + \nu); \text{ Sigmoid Kernel; Activation function for NN.}$$

https://towardsdatascience.com/understanding-support-vector-machine-part-2-kernel-trick-mercers-theorem-e1e6848c6c4d



http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Objetivos

- Describir las características fundamentales de los árboles de decisión y cómo segmentan el espacio de parámetros.
- Explicar los métodos para la construcción, evaluación y poda de árboles de decisión.
- Comprender el concepto de *maximal margin classifier* y su fundamento teórico.
- Analizar las dificultades y limitaciones del *maximal margin classifier* y describir el uso de kernels para abordar dichas dificultades.